NetScaler 11.0

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler 11.0 Release Notes

Jun 12, 2018

The release notes describe the changes or enhancements, fixed issues, and known issues in a build. The list of known issues is cumulative, that is, it includes issues that are newly found in that build and also issues from previous builds.

The release notes document includes all or some of the following sections:

What's New: The enhancements and other changes released in the build.

Fixed Issues: The issues that are fixed in the build.

Known Issues: The issues that exist in the build.

Points to Note: The important aspects to keep in mind while using the build.

Limitations: The limitations that exist in the build.

> ## Note
>
> - The [# XXXXXX] labels under the issue descriptions are internal tracking IDs used by the NetScaler team.
> - These release notes do not document security related fixes. For a list of security related fixes and advisories, see the Citrix security bulletin.

.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

FAQs

Sep 08, 2016

This section provides the frequently asked questions on the following NetScaler features:

- AppFlow
- AutoScale
- Call Home
- Clustering
- Connection Management
- Configuration Utility
- Content Switching
- Debugging
- High Availability
- Hardware
- Integrated Caching
- Load Balancing
- Migration
- SDX
- SSL
- Installing, Upgrading, and Downgrading

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppFlow

May 13, 2016

Which build of NetScaler supports AppFlow?

AppFlow is supported on NetScaler appliances running version 9.3 and above with nCore build.

What is the format used by AppFlow to transmit data?

AppFlow transmits information in the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information.

What do AppFlow records contain?

AppFlow records contain standard NetFlow or IPFIX information, such as time stamps for the beginning and end of a flow, packet count, and byte count. AppFlow records also contain application-level information (such as HTTP URLs, HTTP request methods and response-status codes, server response time, and latency). IPFIX flow records are based on templates that must be sent before sending flow records.

After an upgrade to NetScaler Version 9.3 Build 48.6 Cl, why does an attempt to open a virtual server from the GUI result in the error message "The AppFlow feature is only available on Citrix Netscaler Ncore"

AppFlow is supported only on nCore appliances. When you open the virtual server configuration tab, clear the **AppFlow** checkbox.

What does the transaction ID in an AppFlow records contain?

A transaction ID is an unsigned 32-bit number identifying an application-level transaction. For HTTP, a transaction corresponds to a request and response pair. All flow records that correspond to this request and response pair have the same transaction ID. A typical transaction has four uniflow records. If the NetScaler generates the response by itself (served from the integrated cache or by a security policy), there might be only two flow records for the transaction.

What is an AppFlow action ?

An Appflow action is a set of collectors to which the flow records are sent if the associated AppFlow policy matches.

What commands can I run on the NetScaler appliance to verify that the AppFlow action is a hit?

The show appflow action. For example:

```
> show appflow action
1)    Name: aFL-act-collector-1
      Collectors: collector-1
      Hits: 0
      Action Reference Count: 2
2)    Name: apfl-act-collector-2-and-3
      Collectors: collector-2, collecter-3
      Hits: 0
      Action Reference Count: 1
3)    Name: apfl-act-collector-1-and-3
      Collectors: collector-1, collecter-3
      Hits: 0
      Action Reference Count: 1
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

What is an AppFlow collector?

A collector receives flow records generated by the NetScaler appliance. To be able to send flow records, you must specify at least one collector. You can specify up to four. You can remove unused collectors.

What NetScaler verison is required for using AppFlow?

Use NetScaler version 9.3.49.5 or higher, and remember that AppFlow is available in only the nCore builds.

What transport protocol does AppFlow use?

AppFlow uses UDP as the transport protocol.

What ports needs to be opened if I have a firewall in the network?

Port 4739. It is the default UDP port the AppFlow collector uses for listening on IPFIX messages. If the user changes the default port, that port should be opened on the firewall.

How can I change the default port AppFlow uses?

When you add an AppFlow collector by using the add appflowCollector command, you can specify the port to be used.

```
> add appflowCollector coll1 -IPAddress
10.102.29.251 -port 8000
 Done
```

What does setting clientTrafficOnly do?

NetScaler generates AppFlow records only for client-side traffic.

How many collectors can be configured at a time?

You can configure up to four AppFlow collectors at a time on the NetScaler appliance. Please note that the maximum number of collectors that can be configured on a NetScaler appliance is four.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AutoScale

May 13, 2016

What are the prerequisites for setting up AutoScale?

For prerequisites for setting up AutoScale, see "Prerequisites".

Can the CloudPlatform AutoScale feature be used without a NetScaler appliance?

No. The NetScaler appliance is currently required for the AutoScale feature to work. If the CloudPlatform administrator configures AutoScale in a network that does not include a NetScaler appliance, CloudPlatform throws an error.

What happens if the AutoScale feature is used with a NetScaler release that does not support AutoScale?

If the AutoScale feature is used with a NetScaler release that does not support AutoScale, the CloudPlatform user interface throws an error. CloudPlatform also writes a message to the log file, indicating that the configured NetScaler does not support AutoScale.

What versions of CloudPlatform and NetScaler should I use to implement AutoScale?

For information about NetScaler releases that support AutoScale, see Supported Environment.

In a load balancing rule, can manually provisioned virtual machine instances coexist with instances provisioned by the AutoScale feature?

No. The CloudPlatform virtual machine group in a load balancing rule can contain only manually provisioned instances or only instances provisioned by the AutoScale feature. They cannot coexist.

Is there a limit on the number of virtual machine instances to which we can scale up by using AutoScale?

Yes. The CloudPlatform administrator specifies the maximum number of members to which the configuration can scale up. When the limit is reached, virtual machines are not provisioned even if the scale-up condition is satisfied. The upper limit prevents uncontrolled spawning of VMs due to misconfiguration of the AutoScale feature or unexpected load conditions.

Are AutoScale events observable?

The events generated for deploying or destroying virtual machines are observable. These events are logged in the NetScaler logs (ns.log) and in the CloudPlatform logs (management-server.log). However, you cannot observe the metric values collected by NetScaler monitors.

What metrics can be used in AutoScale policies?

In an AutoScale policy, you can use any metric that is exposed through SNMP, or any NetScaler statistics associated with the load balancing virtual server used in the AutoScale configuration. For example, you can use metrics associated with CPU, memory, or disk usage, and NetScaler metrics such as throughput or response time.

What should a CloudPlatform administrator do before performing maintenance tasks on a CloudPlatform network in which AutoScale is configured?

The CloudPlatform administrator should disable the AutoScale configuration from the CloudPlatform user interface. Disabling the AutoScale configuration temporarily disables any scale-up or scale-down events. However, disabling AutoScale for an application, in CloudPlatform, does not affect the ability of the NetScaler appliance to serve traffic to existing virtual machines.

With AutoScale configured, are any configured VM limits enforced on the user account?

The NetScaler appliance works in the context of an AutoScale user account. Therefore, any limits that the CloudPlatform

© 1999-2017 Citrix Systems, Inc. All rights reserved.

administrator has imposed on the number of VMs that can be created by the account are automatically enforced when the NetScaler appliance attempts to create more VMs than are permitted.

Is AutoScale supported in a high availability (HA) NetScaler pair?

No. Currently, HA mode is not supported for AutoScale.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Call Home

Jun 14, 2016

Note: The current NetScaler 1000V release does not support this feature.

What is the Call Home feature on a NetScaler appliance?

The Call Home feature registers your NetScaler appliance with the Citrix Technical Support server (CIS) and monitors the appliance for common error conditions. If your appliance is successfully registered with CIS server, Call Home automatically uploads system logs to that server in the event that one of the conditions occurs. The appliance keeps a full log of all upload events. If you are unable to correct the problem after reviewing the appliance's log, you can contact the Citrix Technical Support team and open a service request. The team can analyze the uploaded system data and recommend possible solutions.

Which release of NetScaler Software supports Call Home?

NetScaler Release 10.0 and later.

Does Call Home support monitoring of any error conditions in a NetScaler virtual appliance?

Yes, Call Home supports monitoring of NetScaler virtual appliances from NetScaler release 11.0.

Which NetScaler hardware models support Call Home?

NetScaler MPX appliances, NetScaler VPX appliances (NetScaler VPX models 1000 and higher), and NetScaler VPX instances running on NetScaler SDX appliances support Call Home.

Do you need a separate license for Call Home?

No. The Call Home feature does not require a separate license. It is available with all NetScaler platform licenses.

Does Call Home support monitoring of cluster events or error conditions?

No. Call Home does not currently support monitoring of cluster specific error conditions. However, you can use the Call Home feature on individual nodes of a cluster.

What error conditions does Call Home monitor in a NetScaler appliance?

Call Home supports monitoring of the following events in a NetScaler appliance:

- Compact flash drive errors
- Hard disk drive errors
- Power supply unit failure
- SSL card failure
- Warm restart

What mechanism does Call Home use to upload the Call Home tar file to CIS?

Call Home uses the HTTPS protocol to upload the Call Home tar file.

Does Call Home support automatic Technical Support service request creation?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

No. Call Home does not currently support automatic Technical Support service request creation. You must contact the Citrix Technical Support team to open a service request.

What is the frequency of Call Home tar file uploads to CIS?

Call Home creates the Call Home tar file and uploads it to the Citrix Technical Support server (CIS) upon first occurrence of a particular error condition since the appliance was last started. That is, a reoccurrence of same error condition does not trigger another upload unless the appliance was rebooted after the previous occurrence.

Must I configure SNMP for Call Home to monitor error conditions?

No, you need not configure SNMP for Call Home to monitor error conditions, because SNMP and Call Home uploads are independent of each other. If you want to be alerted each time an error condition occurs, you can configure the corresponding SNMP alarm for that error condition.

Can system logs generated by Call Home be uploaded through a proxy server?

Yes. If your NetScaler appliance does not have direct Internet connectivity, you can configure a proxy server, through which system logs generated by Call Home are uploaded to the Citrix Technical Support server (CIS).

Does Call Home mask sensitive data before uploading the system logs to the Citrix Technical Support server?

Yes, all sensitive information from the system logs (such as passwords, SSL key names, and so on) are masked before they are uploaded to the Citrix Technical Support server.

Is the Call Home feature enabled by default on the appliance?

No, the Call Home feature is disabled, by default. You must first enable the feature to register the appliance for critical error conditions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Clustering

May 13, 2016

Click here for FAQs on clustering.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Connection Management

May 13, 2016

What is an admin connection?

An admin connection establishes a connection to the NetScaler IP (NSIP) address and allows administrators to configure and monitor the NetScaler appliance.

What are the types of admin connections?

There are two types of admin connections:

- SSH connection – Admin users use an SSH client to logon through the NetScaler IP (NSIP) address.
- NITRO API connection – Admin users use NITRO API's to automate the logon process to NetScaler appliance.

Note: Admin users can also log on through the NetScaler GUI to log on, by using a browser to connect to the NSIP address. The GUI internally opens a NITRO API connection. Therefore, a GUI session is equivalent to a NITRO API connection, and FAQs related to NITRO API apply to GUI.

How many concurrent admin connections are allowed on a NetScaler appliance?

The appliance allows up to 20 concurrent admin connections.

Which login credentials are required for an admin logon?

Admin logon requires a user name and a password.
Note: An authentication key can be used instead of a password.

Which external authentication methods does a NetScaler appliance support?

The appliance supports the following external authentication methods:

- RADIUS
- LDAP
- TACACS

For more details, see [External Authentication]

What is a client?

A client is a device (laptop or desktop), used by admin user to open an admin connection.

What is a session token?

A session token is a unique identifier that the NetScaler appliance issues to a client that sends a NITRO API logon request.

- API clients can reuse the session token, if it has not expired, for subsequent API requests on new TCP connections
- GUI clients internally open NITRO API connections and keep the session token active for the duration of the GUI session.

What is an active session on a NetScaler appliance?

A CLI session is considered active if the session has not expired and has an open SSH connection with a NetScaler appliance.

A NITRO API session is considered active if the session token timeout has not expired on the NetScaler appliance.

How does NetScaler enforce the concurrent connection limit?

Every time the NetScaler appliance receives an admin connection request (SSH or NITRO API), it checks the number of

© 1999-2017 Citrix Systems, Inc. All rights reserved.

admin connections it has open. If the number is lower than 20, a new connection is opened.

Which counter reflects the number of admin connections on a NetScaler appliance?

The connection counter (nsconfigd_cur_clients) reflects the number of active connections. This counter is incremented when a client opens new connection to the appliance, and is decremented when a connection is closed.

Which counter reflects the number of active tokens on the NetScaler appliance?

The *configd_cur_tokens* counter reflects the number of active tokens on NetScaler appliance.

How does NetScaler appliance handle errors on a connection?

The NetScaler appliance immediately closes the client (CLI, API, and GUI) connection if it encounters errors on a connection.

Does a CLI or GUI session on a connection to the management address count against the admin connection limit?

Yes, all CLI and GUI connections are TCP based connections, and every TCP connection to the management address counts against the admin connection limit.

Does a NITRO session count against the admin connection limit?

A NITRO session counts against the admin connection limit if there is an open TCP connection using the session token issued by the NetScaler appliance.

What is the default timeout period for API, GUI, and CLI sessions on NetScaler appliance?

The following table lists the default timeout period for API, GUI and CLI sessions on the NetScaler appliance:

NetScaler Releases	CLI default timeout period (min)	API default timeout period (min)	GUI default timeout period (min)
NetScaler 9.3	None	30 Minutes	30 Minutes
NetScaler 10.1	None	30 Minutes	30 Minutes
NetScaler 10.5 Onwards	15 Minutes	30 Minutes	15 Minutes

How can you set the CLI sessions time out on a NetScaler appliance?

The CLI session timeout can be configured by executing the following command at the CLI prompt:

```
set cli mode -timeout <xx seconds>
```

How do you override the default timeout period when using the NITRO API?

You can override the default timeout period for a NITRO API by setting the timeout duration in the “timeout” field of the login object. If the session timeout is set to zero, the session token has an infinite timeout.
Note: An infinite timeout is not advisable, because sessions that do not time out continue to count against the admin connection count.

What happens if a user account is deleted from the NetScaler appliance after an admin session is created?

For internal system users, NetScaler appliance closes the existing CLI or NITRO API session.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For external system users, session remains active until it expires.

Can NITRO API clients use a single session token to open multiple admin connections on the NetScaler appliance?

Yes. Each such connection counts against the admin connection limit.

If management access is enabled for a MIP or SNIP address, do admin connections to that address count against the limit for the number of admin connections?

Yes, admin connections to management address (MIP or SNIP) count against the admin connection limit on NetScaler ADC.

Can a NetScaler admin log on to the NetScaler appliance after the maximum connections limit is reached?

Yes. One additional admin connection is allowed after the maximum connection limit is reached.

Can NITRO API endpoints open multiple admin connections on NetScaler the appliance?

Yes, NITRO API endpoints can open multiple admin connections and exhaust the concurrent admin connection limit on a NetScaler appliance. In such situations, an additional SSH/CLI connection is allowed and the admin can force closure of old API sessions, or reduce the session timeout duration for the existing API sessions.

Can same client open multiple API sessions on a NetScaler appliance?

Yes, a client can open multiple API session by repeatedly logging on. For example, the client might log back on after a reboot.
Note: Repeated client logons count against the admin connection limit on NetScaler appliance.

Can API clients use the entire API session token limit?

Yes, API clients can use the entire API session token limit, provided by repeatedly logging on without using a previously issued token.

Note: If a client's session timeout is zero, the token is valid forever. Repeated logons using new session tokens can count against the limit for API session tokens.

Do CLI sessions count against the API session token limit?

No, CLI sessions are not counted against the API session token limit.

Can admin users use telnet to open a CLI session?

No. Only an SSH client can open a CLI session.

What is connection limit and API session limit applicable for various NetScaler releases?

The following table lists the maximum concurrent admin connection and active API session limits applicable for various NetScaler releases:

NetScaler Releases	9.3	10.1 (Before 130.x)	10.1 (Before 130.10)	10.1 (From 130.10)
Maximum number of concurrent admin connections	20	20	20	20
Maximum number of active API sessions*	1000	20	1000	1000

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note*:

- API sessions are considered active if they have not timed out. For example, if 500 API sessions were created but 100 have expired, 400 API sessions are active.
- An API session need not open a TCP connection to the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuration Utility

Oct 09, 2016

Q: When I use Firefox to compare two NetScaler configurations, the browser seems to freeze.

A: Firefox will eventually display the differences in the configurations, but the process takes a considerable amount of time if there are more than 1000 differences. Use Chrome for a faster response.

Q: I am using a MAC Safari browser to upgrade a NetScaler ADC. On the upgrade wizard, when I click the Browse button to choose the build file from the appliance, the dialog box does not show any files or folders. Also, when I navigate back to the root folder, the dialog box displays the top level folder, but I cannot browse it. What should I do?

A: On the Safari browser, click the Settings icon and navigate to Preferences > Security > Manage Website Settings > Java, and then change value of the When visiting other websites setting to Run in unsafe mode.

Q: What should I do before accessing the NetScaler configuration utility?

A: Before accessing a new version of the NetScaler software:

- Clear browser cache including cookies.
- Access GUI in browser incognito mode.
- Access GUI in some other browser.
- Uncheck 'Use software acceleration' option in setting and restart the browser.
- Access chrome: extensions, uncheck the 'Enable' box and restart Chrome browser.

Q: I am using HTTP to access the configuration utility. Which port should I open?

A: Open TCP port 3010 when using HTTP to access the configuration utility.

Q: I am using HTTPS to access the configuration utility. Which port should I open?

A: Open TCP port 3008 when using HTTPS to access the configuration utility.

Q: With which browsers is the configuration utility compatible for different operating systems?

A: The following table lists the compatible browsers:

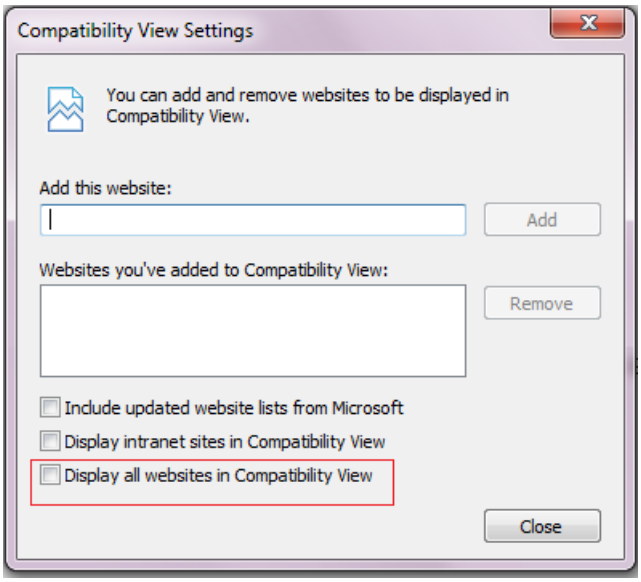
Operating System	Browser	Versions
Windows 7	Internet Explorer	8 and 9
	Mozilla Firefox	3.6.25 and above
	Chrome	15 and above
Windows 64 bit	Internet Explorer	8 and 9

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operating System	Browser	Versions
	Chrome	15 and above
MAC OS	Mozilla Firefox	12 and above
	Safari	5.1.3

Q: When I access the NetScaler configuration utility by using Internet Explorer version 8 or 9, the browser displays only a grey bar at the top of the screen. What should I do?

A: The browser might be set in compatibility mode. To disable compatibility mode, go to **Tools** > **Compatibility View Settings** and clear the **Display all websites in Compatibility View** check box.



Q: Even after I disable compatibility mode in Internet Explorer version 8 or 9, the configuration utility does not appear. What should I do?

A: Make sure that the browser mode and document mode in the browser are set to the same version. To view the configuration, press F12. Set the values to either Internet Explorer 8 or Internet Explorer 9.

Q: After logging into the NetScaler appliance, the page appears blank. What should I do?

A: Make sure you have disabled the Protected mode in your browser settings. If this is enabled, the Java Script is causing the NetScaler user interface screen to appear blank.

To disable this option:

1. In your IE browser settings, go to **Internet Options**.
2. Go to **Security** tab settings, click **Restricted sites** zone to disable **Enable Protected Mode** check box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. Click **Apply** and **OK**.

Q: When I access the NetScaler configuration utility by using Internet Explorer version 9, the utility displays the following error message: "You are not logged in. Please login." What should I do?

A: Make sure that the cookies are not blocked in your Interner Explorer settings. Go to **Tools** > **Internet Options**. Click the **Privacy** tab, and then under **Settings**, make sure that the slider is set to **Medium** or any lower value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Content Switching

Nov 03, 2016

I have installed a non-NetScaler load balancing appliance on the network. However, I would like to use the content switching feature of the NetScaler appliance to direct the client requests to the load balancing appliance. Is it possible to use the Content switching feature of the NetScaler appliance with a non-NetScaler load balancing appliance?

Yes. You can use the Content switching feature of the NetScaler appliance with the load balancing feature of the NetScaler appliance or a non-NetScaler load balancing appliance. However, when using the non-NetScaler load balancing appliance, make sure that you create a load balancing virtual server on the NetScaler appliance and bind it to the non-NetScaler load balancing appliance as a service.

How is a Content switching virtual server different from a load balancing virtual server?

A Content switching virtual server is capable only of sending the client requests to other virtual servers. It does not communicate with the servers.

A Load balancing virtual server balances the client load among servers and communicates with the servers. It monitors server availability and can be used to apply different load balancing algorithms to distribute the traffic load.

Content switching is a method used to direct client requests for specific types of content to targeted servers by way of load balancing virtual servers. You can direct the client requests to the servers best suited to handle them. This result in reduced overheads to process the client requests on the servers.

I want to implement the Content switching feature of the NetScaler appliance to direct the client requests. What types of client request can I direct by using the Content switching feature?

You can direct only HTTP, HTTPS, FTP, TCP, Secure TCP, and RTSP client requests by using the Content switching feature. To direct HTTPS client requests, you must configure the SSL offload feature on the appliance.

I want to create Content switching rules on the NetScaler appliance. What are the various elements of the client request on which I can create a content switching rule?

You can create the content switching rules based on the following elements and their values in the client request:

- URL
- URL tokens
- HTTP version
- HTTP Headers
- Source IP address of the client
- Client version
- Destination TCP port

I understand that the content switching feature of the NetScaler appliance helps enhance the performance of the network. Is this correct?

Yes. You can direct the client requests you the servers best suited to handle them. The result is reduced overhead for processing the client requests on the servers.

Which feature of the NetScaler appliance should I configure on the NetScaler appliance to enhance the site manageability and response time to the client requests?

You can configure the content switching feature of the NetScaler appliance to enhance the site manageability and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response time to the client request. This feature enables you to create content groups within the same domain name and IP address. This approach is flexible, unlike the common approach of explicitly partitioning the content into different domain names and IP addresses, which are visible to the user.

Multiple partitions dividing a Web site into various domain names and IP addresses force the browser to create a separate connection for each domain it finds when rendering and fetching the content of a web page. These additional WAN connections degrade the response time for the web page.

I have hosted a web site on a web server farm. What advantages does the NetScaler content switching feature offer for this type of setup?

The content switching feature provides the following advantages on a NetScaler appliance in a site that is based in a web server farm:

- Manage the site content by creating a content group within the same domain and IP address.
- Enhance the response time to client requests by using the content group within the same domain and IP address.
- Avoid the need for full content replication across domains.
- Enable application-specific content partitioning. For example, you can direct client requests to a server that handles only dynamic content or only static content, as appropriate for the request.
- Support multi-homing of multiple domains on the same server and use the same IP address.
- Reuse connections to the servers.

I want to implement the content switching feature on the NetScaler appliance. I want to direct the client requests to the various servers after evaluating the various parameters of each request. What approach should I follow to implement this setup when configuring the content switching feature?

You can use policy expressions to create policies for the content switching feature. An expression is a condition evaluated by comparing the qualifiers of the client request to an operand by using an operator. You can use the following parameters of the client request to create an expression:

- **Method**- HTTP request method.
- **URL**- URL in the HTTP header.
- **URL TOKENS**- Special tokens in the URL.
- **VERSION**- HTTP request version.
- **URL QUERY**- Contains the URL Query LEN, URL LEN, and HTTP header.
- **SOURCEIP**- IP address of the client.

Following is a complete list of the operators that you can use to create an expression:

- == (equals)
- != (not equals)
- EXISTS
- NOT EXISTS
- CONTAINS
- NOT CONTAINS
- GT (greater than)
- LT (less than)

You can also create various rules, which are logical aggregations of a set of expressions. You can combine multiple expressions to create rules. To combine expressions, you can use the && (AND) and || (OR) operators. You can also use parenthesis to create nested and complex rules.

I want to configure a rule based policy along with a URL based policy for the same content switching virtual

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

server. Is it possible to create both types of policies for the same content switching virtual server?

Yes. You can create both type of policies for the same content switching virtual server. However, be sure to assign priorities to set an appropriate precedence for the policies.

I want to create content switching policies that evaluate the domain name, along with a prefix and suffix of a URL, and direct the client requests accordingly. Which type of content switching policy should I create?

You can create a Domain and Exact URL policy. When this type of policy is evaluated, the NetScaler appliance selects a content group if the complete domain name and the URL in the client request match the ones configured. The client request must match the configured domain name and exactly match the prefix and suffix of the URL if they are configured.

I want to create content switching policies that evaluate the domain name, along with a partial prefix and suffix of URL, and direct the client requests accordingly. Which type of content switching policy should I create?

You can create a Domain and Wildcard URL policy for the content switching virtual server. When this type of policy is evaluated, the NetScaler appliance selects a content group if the request matches the complete domain name and partially matches the URL prefix.

What is a Wildcard URL policy?

You can use wildcards to evaluate partial URLs in client requests to the URL you have configured on the NetScaler appliance. You can use wildcards in the following types of URL-based policies:

- Prefix only. For example, the /sports/* expression matches all URLs available under the /sports URL. Similarly, the /sports* expression matches all URLs whose prefix is /sports .
- Suffix only. For example the /*.jsp expression matches all URLs with a file extension of jsp.
- Prefix and Suffix. For example, the /sports/*.jsp expression matches all URLs under the /sports/ URL that also have the jsp file extension. Similarly, the /sports*.jsp expression matches all URLs with a prefix of /sports* and a file extension of jsp.

What is a Domain and Rule policy?

When you create a Domain and Rule policy, the client request must match the complete domain and the rule configured on the NetScaler appliance.

What is the default precedence set for evaluating policies?

By default, the rule based policies are evaluated first.

If some of the content is the same for all client requests, what type of precedence should I use for evaluating policies?

If some of the content is same for all the users and different content should be served on the basis of client attributes, you can use URL-based precedence for policy evaluation.

What policy expression syntaxes are supported in content switching?

Content switching supports two types of policy expressions:

- **Classic Syntax-** Classic syntax in content switching starts with the keyword REQ and is more advanced than the default syntax. Classic policies cannot be bound to an action. Therefore, the target load balancing virtual server can be added only after binding the content switching virtual server.
- **Default Syntax:** Default syntax generally starts with key word HTTP and is easier to configure. A target load balancing virtual server action can be bound to a Default Syntax policy, and the policy can be used on multiple content switching virtual servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Can I bind a single content switching policy to multiple virtual servers?

Yes. You can bind a single content switching policy to multiple virtual servers by using policies with defined actions. Content switching policies that use an action can be bound to multiple content switching virtual servers because the target load balancing virtual server is no longer specified in the content switching policy. The ability to bind a single policy to multiple content switching virtual servers helps to further reduce the size of the content switching configuration.

For more information, see the following Knowledge Center articles and eDocs topics:

- See CTX122918 - "How to Bind the Same Content Switching Policy to Two Content Switching vServers on a NetScaler Appliance."
- See CTX122736 - "How to Bind the Same Advanced Policy to Multiple Content Switching Virtual Servers using Policy Labels.
- "Configuring a Content Switching Action."

Can I create an action based policy using classic expressions?

No. As of now NetScaler does not support policies using classic syntax expressions with actions. The target load balancing virtual server should be added when binding the policy instead of defining it in an action.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Debugging

May 13, 2016

How can I determine the interface (CLI, GUI, or API) through which an operation was performed?

The NetScaler keeps track of the interfaces through which operations are performed. You can view this information in syslogs (in the NetScaler GUI, navigate to Configuration > System > Auditing > Audit Messages > Syslog messages) or in the ns.log (located at the /var/log/ directory) file.

For example, operations that are performed through the API are flagged as "API CMD_EXECUTED."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

High Availability

Aug 29, 2017

What are the various ports used to exchange the HA-related information between the nodes in an HA configuration?

In an HA configuration, both nodes use the following ports to exchange HArelated information:

- UDP Port 3003, to exchange heartbeat packets
- Port 3010, for synchronization and command propagation

What configurations are not synced or propagated in an HA configuration in either INC or non-INC mode?

Configurations implemented with the following commands are neither propagated nor synced to the secondary node:

- All node specific HA configuration commands. For example, add ha node, set ha node, and bind ha node.
- All Interface related configuration commands. For example, set interface and unset interface.
- All channel related configuration commands. For example, add channel, set channel, and bind channel.

Note: For more information about HA Configuration in INC mode, see Configuring High Availability Nodes in Different Subnets.

What configurations are not synced or propagated in an HA configuration in INC mode?

The following configurations are neither synced nor propagated. Each node has its own.

- MIPs
- SNIPs
- VLANs
- Routes (except LLB routes)
- Route monitors
- RNAT rules (except any RNAT rule with VIP as the NAT IP)
- Dynamic routing configurations.

What are the conditions that trigger synchronization?

Synchronization is triggered by any of the following conditions:

- The incarnation number of the primary node, received by the secondary, does not match that of the secondary node. Note: Both nodes in an HA configuration maintain a counter called *incarnation number*, which counts the number of configurations in the node's configuration file. Each node sends its incarnation number to each other node in the heartbeat messages. The incarnation number is not incremented for the following commands:
  - All HA configuration related commands. For example, add ha node, set ha node, and bind ha node.
  - All Interface related commands. For example, set interface and unset interface.
  - All channel-related commands. For example, add channel, set channel, and bind channel.
- The secondary node comes up after a restart.
- The primary node becomes secondary after a failover.

Does a configuration added to the secondary node get synchronized on the primary?

No, a configuration added to the secondary node is not synchronized to the primary.

What could be the reason for both nodes claiming to be the primary in an HA configuration?

The most likely reason is that the primary and secondary nodes are both healthy but the secondary does not receive the heartbeat packets from the primary. The problem could be with the network between the nodes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Does an HA configuration run into any issues if you deploy the two nodes with different system clock settings?

Different system-clock settings on the two nodes can cause the following issues:

- The time stamps in the log file entries do not match. This situation makes it difficult to analyze the log entries for any issues.
- After a failover, you might have problems with any type of cookie based persistence for load balancing. A significant difference between the times can cause a cookie to expire sooner than expected, resulting in termination of the persistence session.
- Similar considerations apply to any time related decisions on the nodes.

What are the conditions for failure of the *force HA sync* command?

Forced synchronization fails in any of the following circumstances:

- You force synchronization when synchronization is already in progress.
- The secondary node is disabled.
- HA synchronization is disabled on the current secondary node.
- HA propagation is disabled on the current primary node and you force synchronization from the primary.

What are the conditions for failure of the *sync HA files* command?

Synchronizing configuration files fail if the secondary node is disabled.

In an HA configuration, if the secondary node takes over as the primary, does it switch back to secondary status if the original primary comes back online?

No. After the secondary node takes over as the primary, it remains as primary even if the original primary node comes back online again. To interchange the primary and secondary status of the nodes, run the *force failover* command.

What are the conditions for failure of the *force failover* command?

A forced failover fails in any of the following circumstances:

- The secondary node is disabled.
- The secondary node is configured to remain secondary.
- The primary node is configured to remain primary.
- The state of the peer node is unknown.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Hardware

May 13, 2016

Are transceivers shipped with the MPX 8005/8015/8200/8400/8600/8800 appliance?

No. Transceivers are available for purchase separately. Contact your Citrix sales representative to order transceivers for your appliance.

Are transceivers hot-swappable?

The 1G SFP transceiver is hot-swappable with release 9.3 build 47.5 or later on the following NetScaler appliances, which use the Intel e1k interface:

- MPX 7500/9500
- MPX 8005/8015/8200/8400/8600/8800
- MPX 9700/10500/12500/15500
- MPX 11500/13500/14500/16500/18500/20500
- MPX 11515/11520/11530/11540/11542
- MPX 22040/22060/22080/22100/22120
- MPX 24100/24150

The 10G SFP+ transceiver is hot-swappable with release 9.3 build 57.5 or later on the following NetScaler appliances, which use the ixgbe (ix) interface:

- MPX 8005/8015/8200/8400/8600/8800
- MPX 9700/10500/12500/15500
- MPX 11500/13500/14500/16500/18500/20500
- MPX 17500/19500/21500
- MPX 17550/19550/20550/21550
- MPX 11515/11520/11530/11540/11542
- MPX 22040/22060/22080/22100/22120
- MPX 24100/24150
- MPX 25100T/25160T

Why does the 10G SFP+ transceiver autonegotiate to 1G speed?

Autonegotiation is enabled by default on the 10G SFP+ ports into which you insert your 10G SFP+ transceiver. When a link is established between the port and the network, the speed is autonegotiated. For example, if you connect the port to a 1G network, the speed is autonegotiated to 1G.

Can I insert a 1G transceiver into a 10G slot?

The 10G slot supports copper 1G transceivers, which can operate at up to 1 Gbps in a 10 Gbps slot.

Note that you cannot insert a 10G transceiver into a 1G slot.

The following table shows the compatibility matrix of transceivers and ports available on the NetScaler appliance.

Ports	Transceivers		
	10G	1G Fiber	1G Copper

© 1999-2017 Citrix Systems, Inc. All rights reserved.

10G	Supported	Not Supported	Supported
1G Fiber	Not Supported	Supported	Not Supported
1G Copper	Not Supported	Not Supported	Supported

What is QSFP+?

QSFP+ stands for Quad Small Form-factor Pluggable, which is a small, hot-pluggable transceiver for connecting data devices. This transceiver is used for 40G interfaces.

QSFP+ to Four SFP+ Copper Breakout Cables—These cables connect to four SFP+ 10GE ports of a NetScaler appliance on one end and to a QSFP+ 40G port of a Cisco switch on the other end.

Support for 40G connectivity—NetScaler models that have at least four 10G SFP+ ports connect to Cisco 40G interfaces by aggregating four of the 10G SFP+ ports to form a 40G link aggregation channel. QSFP to Four port SFP+ Copper Breakout Cable **QSFP-4SFP10G-CU3M (reports as L45593-D178-C30)** is used.

Which NetScaler appliances support the QSFP-4SFP10G-CU3M (reports as L45593-D178-C30) Breakout Cable?

NetScaler appliances that have at least four 10G SFP+ ports support this cable. The following appliances have at least four 10G SFP+ ports:

- MPX 11500/13500/14500/16500/18500/20500
- MPX 17550/19550/20550/21550
- MPX 11515/11520/11530/11540/11542
- MPX 22040/22060/22080/22100/22120
- MPX 24100/24150
- MPX 25100T/25160T

QSFP-4SFP10G-CU3M breakout cable is supported by NetScaler release 9.3 build 65.8 or later, and release 10.1 build 122.17 or later.

Is the power supply on the NetScaler MPX 5500 and MPX 5550/5650 appliances field replaceable?

No. The power supply on the NetScaler MPX 5500 and MPX 5550/5650 appliances is fixed.

Does the MPX 8005/8015/8200/8400/8600/8800 appliance ship with two power supplies?

No. The MPX 8005/8015/8200/8400/8600/8800 appliance supports dual power supplies but ships with one power supply. Contact your Citrix sales representative to order a second power supply.

How many power supplies are shipped with each platform?

The following table lists the number of power supplies shipped with each platform:

Platform	Number of Power Supplies shipped
MPX 5500	1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Platform	Number of Power Supplies shipped
MPX 7500/9500	1 (You can order a second power supply.)
MPX 9700/10500/12500/15500	2
MPX 15000/17000	1 (You can order a second power supply.)
MPX 11500/13500/14500/16500/18500/20500	2
MPX 17500/19500/21500	1 (You can order a second power supply.)
MPX 17550/19550/20550/21550	2

Are power supplies hot-swappable?

Yes. If the appliance has two power supplies, you can replace one power supply without shutting down the appliance, provided the other power supply is working.

Do you have different rail kits for 1U and 2U appliances?

No. All MPX and SDX appliances use the same rail kit. The kit contains two pairs of slide rails, of different lengths, for a 1U and a 2U appliance.

Which rail kit should I buy?

The appliance ships with the standard 4-post rail kit that fits racks from 28-38 inches.

The compact 4-post rail kit for racks from 23-33 inches, or the 2-post rail kit for 2-post racks, has to be purchased separately. Contact your Citrix sales representative to order the appropriate kit.

What are the maximum and the minimum lengths of the outer rack rails?

The length of a standard outer rack rail is from 28 to 38 inches. The length of a shorter outer rack rail is from 23 to 33 inches.

What is the space required between the front post and rear post of the rack?

Standard racks require 28–38 inches between the front and rear posts. Shorter racks require from 23 to 33 inches.

How far can an appliance extend from the front post of the rack?

The chassis can extend up to 1.25 inches from the front post for all NetScaler MPX and SDX appliances.

How much space is required for maintaining the front and rear area of an appliance?

Minimum clearance areas of 36 inches for the front area and 24 inches for the rear area are required for maintenance of all NetScaler MPX and SDX appliances.

Which LOM features are supported on the NetScaler MPX Appliance?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The MPX 8005/8015/8200/8400/8600/8800, MPX 11500/13500/14500/16500/18500/20500, and MPX 17550/19550/20550/21550 have an Intelligent Platform Management Interface (IPMI), also known as the Lights out Management (LOM) port, on the front panel of the appliance. The following three LOM features are supported on those platforms:

- Configuring the LOM port
- Power cycling the appliance
- Performing a core dump

Can the LOM interface be configured to accept only encrypted Virtual Network Computer (VNC) sessions on TCP port 5900?

Yes, customers who enable Transport Layer Security (TLS) on their LOM interface will have their VNC connections delivered over TLS as well.

For more information on LOM security guidelines, see Secure Deployment Guide for NetScaler MPX, VPX, and SDX Appliances.

Can the version of SSH used on the LOM interface be upgraded? Is there a patch available?

Individual components of the LOM cannot be upgraded independently. You must upgrade the entire LOM firmware as a package. The latest available LOM package can be found on the Citrix downloads website under LOM Firmware Upgrade.

Is it possible to add a third-party or self-signed SSL certificate to the LOM interface?

Yes, you can enable SSL on the latest binaries for third-party and self-signed SSL certificates, except on the 88XX models. On those models, the current LOM release does not support third-party certificates.

What is the recommended terminal emulator?

PuTTY.

Which platforms support Pay-As-You-Grow licenses?

The following platforms support Pay-As-You-Grow licenses:

- MPX 5550 to MPX 5650
- MPX 7500 to MPX 9500
- MPX 8005 to MPX 8015 to MPX 8200 to MPX 8400 to MPX 8600 to MPX 8800
- MPX 11500 to MPX 13500 to MPX 14500 to MPX 16500 to MPX 18500 to MPX 20500
- MPX 17500 to MPX 19500 to MPX 21500
- MPX 17550 to MPX 19550 to MPX 20550 to MPX 21550
- MPX 22040 to MPX 22060 to MPX 22080 to MPX 22100 to MPX 22120

Do you support direct attach cable (DAC)?

Yes, Citrix NetScaler appliances support a passive DAC in the following releases and builds:

- Release 9.3, build 63.4 and later
- Release 9.3.e, build 60.3007.e and later
- Release 10, build 74.2 and later
- Release 10.1, build 112.15 and later

Which port should I insert the DAC into?

DAC is inserted into the 10G port on the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Does the 1G port support DAC?

No. The DAC might fit into a 1G port but is not supported.

How can I order a DAC?

Contact your Citrix sales representative to order a DAC.

Can I mix DAC and fiber transceivers on the same appliance?

Yes. You can mix DAC and fiber transceivers on the same appliance. Each 10G port supports both options.

Can I mix SFP+ fiber and DAC in ports that are part of the same link aggregation channel (LAC)?

No. There must be symmetry between all elements in the same LAC.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Integrated Caching

May 13, 2016

How is a DEFAULT content group different from other content groups?

The behavior of the DEFAULT content group is exactly the same as any other group. The only attribute that makes the DEFAULT content group special is that if an object is being cached and no content group has been created, the object is cached in the DEFAULT group.

What is the 'cache-Control' option of the content group level?

You can send any cache-control header the browser. There is a content group level option, -cacheControl, which enables you to specify the cache-control header that you want to be inserted in the response to the browser.

What is the 'Minhit' option in content group level?

Minhit is an integer value specifying the minimum number of hits to a cache policy before the object is cached. This value is configurable at the content group level. Following is the syntax to configure this value from the command line interface.
add/set cache contentGroup <Content_Group_Name> [-minHits <Integer>]

What is the use of the expireAtLastByte option?

The expireAtLastByte option enables the integrated cache to expire the object as soon as it has been downloaded. Only requests that are outstanding requests at that time are served from cache. any new requests are sent to the server. This setting is useful when the object is frequently modified, as in the case of stock quotes. This expiry mechanism works along with the Flash Cache feature. To configure expireAtLastByte option, run the following command from the command line interface:

add cache contentGroup <Group_Name> –expireAtLastByte YES

What is a caching policy?

Policies determine which transactions are cacheable and which are not. Additionally, policies add or override the standard HTTP caching behavior. Policies determine an action, such as CACHE or NOCACHE, depending on the specific characteristics of the request or response. If a response matches policy rules, the object in the response is added to the content group configured in the policy. If you have not configured a content group, the object is added to the DEFAULT content group.

What is a policy hit?

A hit occurs when a request or response matches a cache policy.

What is a miss?

A miss occurs when a request or response does not match any cache policy. A miss can also occur if the request or response matches a cache policy but some override of RFC behavior prevents the object from being stored in the cache.

I have configured Integrated Caching feature of the NetScaler appliance. When adding the following policy, an error message appears. Is there any error in the command?

add cache policy image_caching -rule exp1 | ns_ext_not_jpeg –action cache
> ERROR: No such command
In the preceding command, the expression should be within the quotation marks. Without quotation marks, the operator is considered to be the pipe operator.

What are the commands that I can run on the NetScaler appliance to check the memory allocated to cache?

To display the memory allocated for cache in the NetScaler appliance, run any of the following commands from the command line interface:

- show cache parameter
  In the output, check the value of the Memory usage limit parameter. This is the maximum memory allocated for cache.
- show cache <Content_Group_Name>
  In the output, check the values of the Memory usage and Memory usage limit parameters indicating the memory used and allocated for the individual content group.

My NetScaler appliance has 2 GB of memory. Is there any recommended memory limit for cache?

For any model of the NetScaler appliance, you can allocate half of the memory to the cache. However, Citrix recommends allocating a little less than half of the memory, because of internal memory dependency. You can run the following command to allocate 1 GB of memory to cache:

set cache parameter -memLimit 1024

Is it possible to allocate memory for individual content groups?

Yes. Even though you allocate memory for the integrated cache globally by running the set cache parameter –memlimit<Integer>, you can allocate memory to individual content groups by running the set cache <Content_Group_Name> –memLimit <Integer> command. The maximum memory you can allocate to content groups (combined) cannot exceed the memory you have allocated to the integrated cache.

What is the dependency of memory between integrated cache and TCP buffer?

If the NetScaler appliance has 2 GB memory, then the appliance reserves approximately 800 to 900 MB of memory and remaining is allocated to FreeBSD operating system. Therefore, you can allocate up to 512 MB of memory to integrated cache and the rest is allocated to TCP buffer.

Does it affect the caching process if I do not allocate global memory to the integrated cache?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If you do not allocate memory to integrated cache, all requests are sent to the server. To make sure that you have allocated memory to the integrated cache, run the show cache parameter command. Actually no objects will be cached if global memory is 0, so this needs to be set first.

What are the options for displaying cache statistics?

You can use either of the following options to display the statistics for cache:

- stat cache
  To display the summary of the cache statistics.
- stat cache –detail
  To display the full details of the cache statistics.

What are the options for displaying the cached content?

To display the cached content, you can run the show cache object command.

What is the command that I can run to display the characteristics of an object stored in cache?

If the object stored in the cache is, for example, GET //10.102.12.16:80/index.html, you can display the details about the object by running the following command from the command line interface of the appliance:

```
show cache object -url '/index.html' -host 10.102.3.96 -port 80
```

Is it mandatory to specify the group name as a parameter to display the parameterized objects in cache?

Yes. It is mandatory to specify the group name as a parameter to display the parameterized objects in cache. For example, consider that you have added the following policies with the same rule:

```
add cache policy p2 -rule ns_url_path_cgibin -action CACHE –storeInGroup g1
add cache policy p1 -rule ns_url_path_cgibin -action CACHE -storeInGroup g2
```

In this case, for the multiple requests, if policy p1 is evaluated, its hit counter is incremented and the policy stores the object in the g1 group, which has hit parameters. Therefore, you have to run the following command to display the objects from the cache:

```
show cache object -url "/cgi-bin/setCookie.pl" -host 10.102.18.152 groupName g1
```

Similarly, for another set of multiple requests, if policy p2 is evaluated, its hit counter is incremented and the policy stores the object in the g2 group, which does not have hit parameters. Therefore, you have to run the following command to display the objects from the cache:

```
show cache object -url "/cgi-bin/setCookie2.pl" -host 10.102.18.152
```

I notice that there are some blank entries in the output of the nscachemgr command. What are those entries?

Consider the following sample output of the nscachemgr command. The blank entries in this output are highlighted in bold face for your reference:

```
root@ns# /netscaler/nscachemgr -a
//10.102.3.89:80/image8.gif
//10.102.3.97:80/staticdynamic.html
//10.102.3.97:80/
//10.102.3.89:80/image1.gif
//10.102.3.89:80/file5.html
//10.102.3.96:80/
//10.102.3.97:80/bg_logo_segue.gif
//10.102.3.89:80/file500.html
//10.102.3.92:80/
//10.102.3.96:80/cgi-bin/rfc/ccProxyReval.pl
Total URLs in IC = 10
```

The blank entries in the output are due to the default caching properties for GET / HTTP/1.1.

How can I flush a selective object from the cache?

You can identify an object uniquely by its complete URL. To flush such object, you can perform any of the following tasks:

- Flush cache
- Flush content group
- Flush the specific object

To flush the specific object, you have to specify the query parameters. You specify the invalParam parameter to flush the object. This parameter applies only to a query.

Does any change in the cache configuration trigger flushing of cache?

Yes. When you make any changes to the cache configuration, all the SET cache commands inherently flush the appropriate content groups.

I have updated the objects on the server. Do I need to flush the cached objects?

Yes. When you update objects on the server, you must flush the cached objects, or at least the relevant objects and content groups. The integrated cache is not affected by an update to the server. It continues to serve the cached objects until they expire.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

What is Flash Cache feature of the NetScaler appliance?

The phenomenon of Flash crowds occurs when a large number of clients access the same content. The result is a sudden surge in traffic toward the server. The Flash Cache feature enables the NetScaler appliance to improve performance in such situation by sending only one request to the server. All other requests are queued on the appliance and the single response is served to all of the requests. You can use either of the following commands to enable the Fast Cache feature:

- add cache contentGroup <Group_Name> -flashCache YES
- set cache contentGroup <Group_Name> -flashCache YES

What is the limit for Flash Cache clients?

The number of Flash Cache clients depends on the availability of resources on the NetScaler appliance.

Does the NetScaler appliance proactively receive objects upon expiry?

The NetScaler appliance never proactively receives objects on expiry. This is true even for the negative objects. The first access after expiry triggers a request to the server.

Does the integrated cache add clients to the queue for serving even before it starts receiving the response?

Yes. The integrated cache adds clients to the queue for serving even before it starts receiving the response.

What is the default value for the Verify cached object using parameter of the cache configuration?

HOSTNAME_AND_IP is the default value.

Does the NetScaler appliance create log entries in the log files?

Yes. The NetScaler appliance creates log entries in the log files.

Are compressed objects stored in the cache?

Yes. Compressed objects are stored in the cache.

What happens to objects that are currently stored in cache and are being accessed though SSL VPN?

Objects stored in the cache and accessed regularly are served as cache hits when accessed through SSL VPN.

What happens to objects stored in the cache when accessed through SSL VPN and later accessed through a regular connection?

The objects stored through the SSL VPN access are served as a hit when accessed through the regular connection.

When using weblogging, how do I differentiate entries that indicate response served from cache from those served by the server?

For responses served from the integrated cache, the server log field contains the value IC. For responses served from a server, the server log field contains the value sent by the server. Following is a sample log entry for an integrated caching transaction:

"10.102.1.52 - "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 4.0; .NET CLR 1.0.3705)" "GET /" 200 0 "IC" 10.102.1.45"

Along with a client request, the response logged is the one sent to the client and not necessarily the one sent by the server.

What do you mean by configuring relexpiry and absexpiry?

By configuring relexpiry and absexpiry, it means that you are overriding the header irrespective of what appears in the header. You can configure different expiry setting and the content group level. With relexpiry, expiration of the header is based on the time at which the object was received by the NetScaler. With absexpiry, expiration is based on the time configured on the NetScaler. Relexpiry is configured in terms of seconds. Absexpiry is a time of day.

What do you mean by configuring weakpos and heuristic?

The weakpos and heuristic are like fall back values. If there is an expiry header, it is considered only if the last-modified header is present. The NetScaler appliance sets expiry on the basis of the last-modified header and the heuristic parameter. The heuristic expiry calculation determines the time to expiry by checking the last-modified header. Some percentage of the duration since the object was last modified is used as time to expiry. The heuristic of an object that remains unmodified for longer periods of time and is likely to have longer expiry periods. The –heurExpiryParam specifies what percentage value to use in this calculation. Otherwise, the appliance uses the weakpos value.

What should I consider before configuring dynamic caching?

If there is some parameter that is in name-value form and does not have the full URL query, or the appliance receives the parameter in a cookie header or POST body, consider configuring dynamic caching. To configure dynamic caching, you have to configure hitParams parameter.

How is hexadecimal encoding supported in the parameter names?

On the NetScaler appliance, the %HEXHEX encoding is supported in the parameter names. In the names that you specify for hitParams or invalParams, you can specify a name that contains %HEXHEX encoding in the names. For example, name, nam%65, and n%61m%65 are equivalent.

What is the process for selecting a hitParam parameter?

Consider the following excerpt of an HTTP header for a POST request:

How do we select a hitparam?

```
POST /data2html.asp?param1=value1&param2=&param3&param4=value4
HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-powerpoint, application/vnd.ms-excel,
application/msword, application/x-shockwave-flash, */*
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Referer: http://10.102.3.97/forms.html
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: 10.102.3.97
Content-Length: 153
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDQGQGGRNY=NNLLKDADEENOAFLCCDGFGDMO
S1=This+text+is+only+text%2C+not+more+and+not+less%2C+%0D%0Ajust+text+to+be+itself%2C+namely+%22Text%22+to+be+posted+as+text+%28what+else...%29&**B1**=Submit
In the above request, you can use S1 and B1, highlighted in bold face for your reference, as hitParams depending on your requirements. Additionally, if you use -matchCookies YES in the ASPSESSIONIDQGQGGRNY content group, then you can also use these parameters as hitParams.

What happens to the queued clients if the response is not cacheable?

If the response is not cacheable, all of the clients in the queue receive the same response that the first client receives.

Can I enable the Poll every time (PET) and Flash Cache features on the same content group?

No. You cannot enable PET and Flash Cache on the same content group. The integrated cache does not perform AutoPET function on Flash Cache content groups. The PET feature ensures that the integrated cache does not serve a stored object without consulting the server. You can configure PET explicitly for a content group.

When are the log entries created for the queued clients?

The log entries are created for the queued clients soon after the appliance receives the response header. The log entries are created only if the response header does not make the object non-cacheable.

What is the meaning of the DNS, HOSTNAME, and HOSTNAME_AND_IP values of the Verify cached object using parameter of the cache configuration?

The meanings are as follows:

- set cache parameter -verifyUsing HOSTNAME
  This ignores the destination IP address.
- set cache parameter -verifyUsing HOSTNAME_AND_IP
  This matches the destination IP address.
- set cache parameter -verifyUsing DNS
  This uses the DNS server.

I have set weakNegRelExpiry to 600, which is 10 minutes. I noticed that 404 responses are not getting cached. What is the reason ?

This completely depends on your configuration. By default, 404 responses are cached for 10 minutes. If you want all 404 responses to be fetched form the server, specify–weakNegRelExpiry 0. You can fine tune the –weakNegRelExpiry to a desired value, such as higher or lower to get the 404 responses cached appropriately. If you have configured –absExpiry for positive responses, then it might not yield desired results.

When the user accesses the site by using the Mozilla Firefox browser, the updated content is served. However, when the user accesses the site by using the Microsoft Internet Explorer browser, stale content is served. What could be the reason?

The Microsoft Internet Explorer browser might be taking the content from its local cache instead of the NetScaler integrated cache. The reason could be that the Microsoft Internet Explorer browser is not respecting the expiry related header in the response.

To resolve this issue, you can disable the local cache of the Internet Explorer and clear the offline content. After clearing the offline content, the browser should display the updated content

What if Hits are zero?

Check to see if the server time and NS time are in sync. And the weakPosrelexpiry limit set should bear the time difference between NS and server as shown below

root@ns180# date

Tue May 15 18:53:52 IST 2012

Why are policies getting hits but nothing is being cached?

Verify that memory is allocated to the integrated cache and that the allocation is greater than zero.

Is it possible to zero the cache counters?

There is no command line or GUI option for setting the cache counters to zero, and flushing the cache does not do so either. Rebooting the box automatically sets these counters to zero.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing

Nov 10, 2016

What are the various load balancing policies I can create on the NetScaler appliance?

You can create the following types of load balancing policies on the NetScaler appliance:

- Least Connections
- Round Robin
- Least response time
- Least bandwidth
- Least packets
- URL hashing
- Domain name hashing
- Source IP address hashing
- Destination IP address hashing
- Source IP - Destination IP hashing
- Token
- LRTM

Can I achieve the Web farm security by implementing load balancing using the NetScaler appliance?

Yes. You can achieve Web farm security by implementing load balancing using the NetScaler appliance. NetScaler appliance enables you to implement the following options of the load balancing feature:

- IP Address hiding: Enables you to install the actual servers to be on private IP address space for security reasons and for IP address conservation. This process is transparent to the end-user because the NetScaler appliance accepts requests on behalf of the server. While in the address hiding mode, the appliance completely isolates the two networks. Therefore, a client can access a service running on the private subnet, such as FTP or a Telnet server, through a different VIP on the appliance for that service.
- Port Mapping: Enables the actual TCP services to be hosted on non-standard ports for security reasons. This process is transparent to the end-user as the NetScaler appliance accepts requests on behalf of the server on the standard advertised IP address and port number.

What are various devices that I can use to load balance with a NetScaler appliance?

You can load balance following devices with a NetScaler appliance:

- Server farms
- Caches or Reverse Proxies
- Firewall devices
- Intrusion detection systems
- SSL offload devices
- Compression devices
- Content Inspection servers

Why should I implement the load balancing feature for the website?

You can implement the Load balancing feature for the website to take the following advantages:

- Reduce the response time: When you implement the load balancing feature for the website, one of the major benefits is the boost you can look forward to in load time. With two or more servers sharing the load of the web traffic, each of the servers runs less traffic load than a single server alone. This means there are more resources available to fulfill the client requests. This results in a faster website.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Redundancy: Implementing the load balancing feature introduces a bit of redundancy. For example, if the website is balanced across three servers and one of them does not respond at all, the other two can keep running and the website visitors do not even notice any downtime. Any load balancing solution immediately stops sending traffic to the backend server that is not available.

Why do I need to disable the Mac Based Forwarding (MBF) option for Link Load Balancing (LLB)?

- If you enable the MBF option, the NetScaler appliance considers that the incoming traffic from the client and the outgoing traffic to the same client flow through the same upstream router. However, the LLB feature requires that the best path be chosen for the return traffic.
- Enabling the MBF option breaks this topology design by sending the outgoing traffic through the router that forwarded the incoming client traffic.

What are the various persistence types available on the NetScaler appliance?

The NetScaler appliance supports the following persistence types:

- Source IP
- Cookie insert
- SSL session ID
- URL passive
- Custom Server ID
- Rule
- DESTIP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Migration

May 13, 2016

How is the rollback procedure performed on a NetScaler appliance?

The rollback procedure is similar to the basic upgrade procedure. Select the target build that you want to roll back to and perform the downgrade.

Before rolling back to a different release, Citrix recommends that you create a copy of your current configuration files. To downgrade from release 10.1, see Downgrading from Release 10.1 or to downgrade to an earlier build in 10.1, see Downgrading to an Earlier Build within Release 10.1 .

Can I upgrade a NetScaler appliance directly from release 10.0 to 10.5, or should I upgrade to 10.1 first and then to 10.5?

You can directly upgrade a NetScaler appliance regardless of the version or the build number.

Is the procedure for upgrading a NetScaler from classic to nCore different?

Yes. Only version 9.3 or earlier supports classic builds. For details, see Upgrading from a Classic to an nCore Release .

Does 10.0 support classic builds?

No. Release 10.0 supports only nCore builds. You must have a multi-processor Netscaler (MPX or higher) to upgrade to 10.0.

Can the primary appliance and secondary appliance have separate builds?

Recommended practice is to use the same version and build number on both the primary and the secondary appliance.

Can both the appliances in an High Availability (HA) pair be upgraded at the same time?

No. In an HA pair, first upgrade the secondary node and then upgrade the primary node. For details, refer Upgrading a High Availability Pair or Upgrading a NetScaler High Availability Pair to a Later Build.

Does Citrix support firmware upgrades in the amazon AWS cloud?

Yes.

Can I upgrade the NetScaler instance independently of the SDX version.

It is not required to upgrade the SDX version when the NetScaler appliance is upgraded. However, some features might not work.

Can I use the FTP server to upgrade the NetScaler appliance?

No. You must first download the firmware from the Citrix download site, save it on your local computer and then upgrade the appliance.

Is the procedure for upgrading the NetScaler appliance with GSLB configurations different from an upgrade of an appliance that is not involved in GSLB?

No. The upgrade procedure is similar to the basic upgrade procedure. The only difference is that you can upgrade the standalone or HA appliances on different sites in a phased manner.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SDX

Sep 04, 2013

What is SDX?

SDX is a true service delivery networking platform for enterprises and cloud datacenters. SDX features an advanced virtualization architecture that supports multiple NetScaler instances on a single hardware appliance.

When do I need SDX?

If you have multiple enterprise applications that have independent life cycle needs for L4–L7 networking services, or if you have a need to consolidate multiple underutilized load balancing appliances, you benefit from SDX.

What's unique about SDX?

SDX uniquely delivers key benefits from advancements in server hardware virtualization, hardware-assisted SSL acceleration, and the market-proven, award-winning NetScaler product line. The Management Service features an advanced control plane to unify provisioning, monitoring, and management in the most demanding multitenant environments, while providing full resource isolation for data separation and to meet service level agreement guarantees, such as availability, reliability, and performance.

How will I benefit from SDX?

SDX delivers isolated multitenancy with up to 40:1 consolidation. As a key pillar in Citrix's TriScale technology framework, SDX addresses the growing need to "scale in" within virtual data centers and cloud network infrastructures. The TriScale scale-in factor enables IT to provide the foundation for consolidating L4–L7 network services today, thereby simplifying the build-out of cloud based services down the line, in accordance with business requirements.

Will I need to go outside my normal procurement procedure to purchase SDX?

SDX is a fully contained networking appliance, designed for network deployment. SDX is not designed to be managed through standard hypervisor management tools such as XenCenter.

How do I purchase an SDX?

An SDX order has three basic product components: SDX appliance SKU, SDX support contract SKU, and Add-On Instance Packs. SKUs are also available for platform conversion (MPX-to-SDX) and platform upgrade (SDX-to-SDX). SDX today is available in Platinum Edition only.

Is there SDX-specific documentation?

Yes, please visit http://support.citrix.com/proddocs/topic/netscaler/sdx-ag-wrapper-con.html.

Do NetScaler editions apply to NetScaler SDX?

The editions do not apply from a packaging perspective. NetScaler SDX appliances and the instance 5-packs are priced the same regardless of the edition. However, when provisioning new instances, the administrator is free to deploy the Standard, Enterprise, or Platinum edition of the NetScaler software.

How much memory can I assign to each instance?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

There is no maximum limit to the memory that can be assigned to each instance. Minimum memory required per instance is 2GB.

Can we migrate the existing configuration (ns.conf) from the MPX platform to SDX VPX instance?

Yes, but some configuration, such as RBA policies and SNMP community configuration, is deleted.

What NetScaler features do I get with SDX?

All NetScaler features are available on SDX.

Does SDX accelerate SSL in hardware like MPX does?

Yes. You can assign SSL cores to an instance during provisioning.

What changes to my network are required for me to deploy SDX?

SDX fits into your network environment through standard Ethernet interfaces. You must disable link aggregation control protocol (LACP) on any external switch ports connected to the appliance.

Is SDX interoperable with my routing and switching infrastructure?

Yes, although link aggregation control protocol (LACP) is currently not supported. However, SDX supports manual link aggregation.

Is SDX interoperable with my existing NetScaler deployment?

Yes, although standard VPX-to-MPX limitations apply. For example, high availability is supported only across homogeneous devices (you cannot pair a virtual device with a physical device), some configuration, such as RBA policies and SNMP configuration, is deleted, and license transfer is not supported.

Can I manage SDX from Command Center?

Yes. You can identify SDX appliances and provision and de-provision VPX instances by using Command Center.

How does SDX deliver multitenancy?

Each instance runs as a separate virtual machine with its own dedicated NetScaler kernel, CPU resources, memory resources, address space, and bandwidth allocation. Network I/O is done in a way that not only maintains aggregate system performance but also enables complete segregation of each tenant's data-plane and management-plane traffic.

Do I need to manage an SDX through XenCenter?

No. XenCenter is not supported. Use the Management Service to manage XenServer.

We are a VMware shop. We have no infrastructure available to support XenServer, do you have a VMware variant of SDX?

No additional XenServer infrastructure is necessary. SDX is a fully contained networking appliance with its own control plane, and the virtualization layer is transparent to the deployment.

Why is the system health monitoring page not showing any data?

You have to install the supplemental pack before you can use this feature. For installation instructions for the supplemental pack, see http://support.citrix.com/article/CTX132877.

How do I verify that the supplemental pack installation was successful?

After installation, a pop up window shows whether installation was successful or if there was an error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Why is the VPX instance not reachable after interfaces on the appliance are modified?

When you provision a NetScaler VPX instance with L2VLAN configuration, physical interfaces on the SDX appliance are mapped to virtual interfaces on the VPX instance. If you remove an interface, you might change the mapping between the physical interfaces and VPX instances, and therefore you might lose connectivity to the VPX instance.
For example,

1. You provision a VPX instance, by using the Management Service, with interfaces 10/1, 10/2, 10/7, and tag VLAN 512 to interface 10/2. When you log on to that VPX instance, you see that interfaces 10/1, 10/2, and 10/3 are configured.
2. If you later modify the instance and remove interface 10/1, you lose connectivity to the instance, because interface 10/2 is renamed to 10/1 in the VPX instance.

Are IPv6 addresses supported on the NetScaler SDX appliance?

Yes. All NetScaler-supported IPv6 functionality is available on the SDX appliance.

Where are link parameters, such as speed and duplex, configured?

Link parameters are configured from the Management Service.

Should the appliance be restarted if the platform license is upgraded?

No. You do not need to restart the appliance for the new license to apply.

Do I need to restart the appliance to upgrade the device-level firmware?

Yes, this upgrade is handled through the Management Service and requires that the appliance be restarted. This is the only time that the SDX appliance needs a complete restart.

Do I need to restart the appliance when I upgrade it by using a Pay-As-You-Grow license?

No. Upgrading the appliance upgrades the platform license. Restart the Management Service but not the instances running on the SDX appliance. Once upgraded, the Management Service detects the higher throughput available for the instances. If you decide to increase the bandwidth limit for an instance, restart that instance after modifying the bandwidth limit.

What happens to production instances if I remove my platform license?

There is no change to the production instances. However, you cannot add new instances.

How can we readd a gadget to the Home page?

Click the << button in the top-right corner of the Home page. Then, type the name of the gadget, or press Enter for all gadgets. Click "Add to Dashboard".

Should member interfaces in manual link aggregation be part of same VLAN?

Yes. Member interfaces in manual link aggregation should be part of the same VLAN.

How many VLANs are supported per interface with VLAN filtering enabled? What happens if I configure more?

With VLAN filtering enabled, 10G interfaces support up to 63 VLANs, and 1G interfaces suppport up to 31 VLANs. This is a hard limit based on the number of the queues supported by the NIC. An error message appears if the limit is exceeded.

How many instances can be shared on a single NIC?

For a 10G interface, SDX supports up to 63 virtual functions per physical port, which translates to 63 instances per 10G NIC. For 1G interfaces, the maximum number of shared instances per NIC is 7.

Why is the XenServer password the same as the Management Service password?

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The XenServer password and the Management Service password are the same to maintain administrative consistency. Changing the XenServer password causes the internal communication between the Management Service and XenServer to fail.

If I have separate management networks, do I need to manually add these networks to the Management Service?

No. Communication is over an external device.

Why can't I modify the default administrator profile?

The default administrator profile enables multiple administrative roles to exist on the SDX. You cannot change the password of the nsroot administrator profile, but you can create a new administrator profile and make it the default profile.

Why does Core usage show 50% when I'm not passing any traffic through my NetScaler instance?

CPU core usage shows, from the hypervisor perspective, the CPU utilization of one physical CPU, which has two hyperthreads: one for the packet engine and one for the management CPU. For example, assume a single instance with one dedicated core. Even if you are not passing any traffic through your appliance, PE CPU utilization will be 100%, and average core utilization will be 50%.

Will restarting the Management Service interrupt my production instances?

No. Your production instances will continue to pass traffic without interruption while the Management Service restarts. The same applies when you upgrade the Management Service.

Can I configure the Management Service to send syslog?

Syslog through the Management Service is currently not supported.

Am I required to upgrade all VPX instances if I upgrade the Management Service?

No, instance life cycles can be managed independently of one other and of the life cycle of the Management Service.

If my Management Service and VPX instances are on different networks, how can I manage the VPX instance through HTTPS?

The same way as if they are on the same network.

If my Management Service and VPX instances are on different networks, how can I manage the VPX instance through the Management Service?

If the Management Service and the VPX instance are in different networks but the instance can be reached from Management Service, the Management Service shows the instance as UP. If an instance is UP, you can manage it from the Management Service. However, if communication between the two fails, the Management Service shows the instance as "Out of Service".

I forgot the IP address of my Management Service. What can I do?

Log on to XenServer, and then use the default IP address (169.254.0.10) to log on to the Management Service. At the shell prompt, type networkconfig to view or modify the IP address of the Management Service.

Can I specify VLANs on management interfaces?

VLANs on management interfaces are currently not supported.

How do I restart XenServer?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The only supported method for restarting XenServer is from the Management Service. It is equivalent to restarting the appliance.

How many instances can I provision on the SDX appliance? How much aggregate throughput can I expect?

This number is dependent on the hardware and the license that you purchased, as shown below:

- 11500, 13500, 14500, 16500, 18500, 20500—5 to 20 instances. Throughput ranges from 8 to 42 Gbps.
- 17500, 19500, 21500—5 to 20 instances. Throughput ranges from 20 to 50 Gbps.
- 17550, 19550, 20550, 21550—5 to 40 instances. Throughput ranges from 20 to 50 Gbps.
- 8400, 8600—2 to 5 instances. Throughput ranges from 4 to 6 Gbps.

Note: For more information, see the NetScaler datasheet at http://www.citrix.com/content/dam/citrix/en_us/documents/products/netscaler-data-sheet.pdf

Can I restrict functionality on the VPX instances?

Some functionality can be restricted by specifying the license (Standard, Enterprise, or Platinum) when you provision the instance.

How many SDX models are there, and how do they differ?

The NetScaler SDX appliance comes in the following variants:

- SDX 11500/13500/14500/16500/18500/20500—8 to 42 Gbps, maximum 20 instances, 8x1G ports, 4x10G ports.
- SDX 17500/19500/21500—20 to 50 Gbps, maximum 20 instances, 8x10G ports.
  Note: This platform is going EOS this year.
- SDX 17550/19550/20550/ 21550—20 to 50 Gbps, maximum 40 instances, 8x10G ports.
- SDX 8400/8600—4 to 6 Gbps, maximum 5 instances, (6x10/100/1000Base-T copper Ethernet ports + 6x1G SFP) and (6x10/100/1000Base-T copper Ethernet ports + 2x10G SFP+)

What is the minimum NetScaler software version required for SDX instances?

NetScaler VPX instances should run release 9.3 and later to be able to work on SDX.

How many physical interfaces will I need to use?

If you have a single management network, you'll need on an average 1 or 2 physical NICs per instance. For 2 or more management networks (multiple VLANs for NetScaler IP addresses), you'll need a dedicated separate physical NIC for each management VLAN trunk. You can share physical NICs among multiple instances with L2 separation. Therefore, depending on your topology, you can offset the management VLAN trunk count with multiple instances sharing a physical NIC.

Can I upgrade my MPX to an SDX? What about my MPX FIPS platform?

A non-FIPS MPX platform that supports the SDX architecture can be converted to a similar class of SDX platform. The MPX platform must have a platinum license to be eligible for this upgrade. This is a one way upgrade, and it wipes out the entire configuration on that MPX platform. For more information about this upgrade, see http://support.citrix.com/article/CTX129423.

How many SSL cards (cores) are supported on a NetScaler SDX appliance?

The number of SSL cards supported varies by the platform as follows:

- SDX 17500/19500/21500—16 cards.
- SDX 11500/13500/14500/16500/18500/20500—16 cards.
- SDX 17550/19550/20550/21550—36 cards.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- SDX 8400/8600—4 cards.

Note: Instances cannot share SSL cores. Any SSL cores that are allocated at the time of provisioning an instance are dedicated to that instance.

Can I apply my VPX license to SDX?

No. NetScaler SDX and NetScaler VPX have different licensing models. One license cannot be used for the other.

Why are the hardware sensors not displayed on the NetScaler SDX 17500/19500/21500 appliance?

The NetScaler SDX 17500/19500/21500 is built on the MPX 17500/19500/21500 hardware platform. These appliance configurations do not support monitoring of hardware components.

When I upgraded my MPX to an SDX, the LCD panel went dark. Is that expected?

Yes, that is normal behavior. SDX does not support the LCD panel.

What are RX and TX errors on the NetScaler SDX appliance?

RX and TX errors include cyclic redundancy check (CRC) errors and small or runt packet errors.

What happens if a hardware component is removed from the SDX appliance?

If a hardware component is physically removed from the appliance, it no longer appears in the Management Service user interface.

Do I need to restart my appliance after I reconfigure VLAN filtering?

No. However, you need to restart the VPX instances that are affected by this change. The Management Service restarts the affected instances if you select "Reboot associated Instances" in the Enable/Disable VLAN Filter dialog box.

What is the NMI button for on the SDX appliance?

The NMI button is not operational on the SDX appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL

Jan 23, 2018

For the list of FAQs, see SSL FAQs.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing, Upgrading, and Downgrading

May 13, 2016

What is the use of the zebos.conf file available in a NetScaler release?

A NetScaler appliance uses Zebos as the routing suite. The zebos.conf file available in a NetScaler release is the configuration file for Zebos.

I want to change the SSH port (22) on the NetScaler appliance to some other port. Is it possible to change the SSH port on the appliance?

Yes. You can change the SSH port on the NetScaler appliance by editing the sshd_config file in the /nsconfig directory. If the file does not exist in the /nsconfig directory, copy it from the /etc directory.
In the sshd_config file, edit the entry for Port 22 to Port <Number>, where <Number> is the target port number. If you do not want to restart the appliance and make the changes effective, terminate the sshd process by using the kill command, and then restart the process.

The flash directory is missing from the NetScaler appliance. What procedure should I follow to mount the flash directory?

To mount the flash directory, do the following:

1. Start the NetScaler appliance in single-user mode.
   When the appliance starts, the following message appears:

   Hit [Enter] to boot immediately, or any other key for command prompt. Booting [kernel] in 10 seconds...” Hit space and you should see the following prompt:

   Type '?' for a list of commands, 'help' for more detailed help.

2. Enter the following command to start FreeBSD in single-user mode:
   boot –s

   After the appliance starts, the following message appears:

   Enter full pathname of shell or RETURN for /bin/sh:

3. Press Enter to display the # prompt.
4. Run the following command to mount the flash directory:
   mount /dev/ad0s1a /flash

   Note: If the preceding command displays an error message about permissions, run the following command to check the disk for consistency:
   fsck /dev/ad0s1a

   Run the mount command again to mount the flash directory.

5. Restart the appliance.
6. From the shell prompt, run the following command to verify that the flash directory is mounted:
   df -kh

I want to log on to the NetScaler appliance without entering the password. Is it possible to configure SSH on the appliance to allow that?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Yes. You can configure SSH on the NetScaler appliance to log on without a password. However, you must provide your user name. To configure SSH for logging in without a password, do the following:

1. Run the following command to generate the public and private keys:
   # ssh-keygen -t rsa
2. Run the following command to copy the id_rsa.pub file to the .ssh directory of the remote host that you want to log on to:
   # scp id_dsa.pub <user>@<remote_host>/.ssh/id_dsa.pub
3. Log on to the remote host.
4. Change to the .ssh directory.
5. Run the following commands to add the public key of the client to the known public keys:
   # cat id_dsa.pub >> authorized_keys2

   # chmod 640 authorized_keys2

   # rm id_dsa.pub

What is the procedure to reset the NetScaler appliance BIOS? Under what circumstances should I reset the BIOS?

To reset BIOS of the NetScaler appliance, complete the following procedure:

1. Connect to the appliance through the serial port.
2. Start the appliance and press Delete as soon as the boot-up process starts.
   Pressing Delete during the POST process displays the appliance's BIOS settings.
3. Activate the Exit page of the BIOS settings.
4. Select the Load Optimal Defaults option.
   The Load Optimal Settings message box appears.
5. Select OK.
6. Make the following changes to the BIOS settings on the various tabs:

Tab	Group	Component	Set to...
Advanced	SuperIO Configuration	Parallel Port Address	Disabled
	Floppy Configuration	Floppy A	Disabled
	Boot Settings Configuration	Quiet Boot	Disabled
		PS/2 Mouse Support	Disabled
		Parity Check	Enable
	Remote Access Configuration	Remote Access	COM1
		Serial Port Mode	9600 8'n'1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tab	Group	Component	Set to...
Chipset		Hyper-threading	Disabled
PCI PnP		Allocate IRQ to PCIVGA	NO
		USB Function	Disabled
		Legacy USB Support	Disabled
Power		ACPI Aware OS	NO
		Power Management	Enabled

Note: The BIOS options differs by appliance model.

7. Activate the Exit page of the BIOS settings.
8. Select Save changes and Exit.
9. Select OK to confirm.
10. Verify that the appliance starts cleanly and the serial console displays output after the appliance starts.

You need to reset BIOS when the serial console does not respond. This usually happens after you upgrade the appliance and the serial console is disabled. However, you can still access the appliance by using the telnet or SSH utility.

I need to reset the NetScaler appliance to the factory defaults. What procedure should I follow?

To reset the NetScaler appliance to the factory defaults, you need to reset two environments: the NetScaler application environment and the base FreeBSD environment.

To reset the NetScaler application environment of the appliance to the factory defaults, do the following:

1. Make a backup of the appliance's /nsconfig/ns.conf.
2. Delete the /nsconfig/ns.conf file.
3. Restart the appliance.

To reset the FreeBSD environment of the appliance to the factory defaults, do the following:

1. Install a fresh NetScaler code image on the appliance. This overwrites a number of FreeBSD-level configuration files with default values.
2. Delete any users and groups that are added to the appliance, that is, all except the default users.
3. Delete the /etc/resolv.conf file.
4. Delete the entries that you have added to the /etc/hosts file.
5. If the /etc/rc.netscaler file exists, delete it.
6. Open the /etc/nsperm_group_suser file and make sure that all IOCTL entries are comment entries.
7. Open the /etc/rc.conf file and make sure that the syslogd_enable=NO entry is not changed to syslogd_enable=YES.
8. Open the /etc/syslog.conf file and make sure that there are no additional entries in the file.
9. Delete the contents of the /var/nslog, /var/nstrace, and /var/crash files.
10. If the syslog process is enabled on the appliance and the appliance creates log files locally, delete the contents of the log files listed in the /etc/syslog.conf file. The files are created in the /var/log directory. For example, if syslog process writes system events to the /var/log/events file, and sslvpn access events to the /var/log/sslvpnevents file, delete these files.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The appliance displays a message similar to the “Jun 21 12:20:18 ns /flash/ns-10.0-47.15: [1/2]dc0: NIC hang condition #663: TX 10000/10000, RX 0, HF 0” message on the console. What is the meaning of this message?

The message consists of the following components (shown here as examples):

- #663: Number of times this condition has occurred on the appliance.
- TX 10000/10000: Number of packets that the appliance attempted to transmit, and number of packets transmitted. If both numbers are the same, as in this example, the NIC transmitted all the packets that the appliance attempted to transmit.
- RX 0: Number of packets received. In this example, no packet was received.
- HF0: Number of hardware issues reported by the NIC. In this example, the NIC did not report any hardware issue.

If the appliance does not receive any packets, it reports a hang condition, because on a network it is very unlikely not to receive any packets. However, if the appliance is plugged into quite interface, you can ignore this error message.

After I upgraded the NetScaler release on the appliance, the appliance still displays the earlier release/build. What could be the reason?

The appliance displays the software version number from the /flash/boot/loader.conf file. If the kernel entry for the current NetScaler release is missing from that file, the appliance displays the last NetScaler release version for which the entry was available.

To resolve this issue, do the following:

1. Verify that the kernel file exists in the /nsconfig directory.
2. Check the /flash/boot/loader.conf file for an entry for the kernel.
   (You can expect the entry for the kernel of the release/build that you just installed to be missing from the file.)
3. Open the loader.conf file in a text editor, such as the vi editor, and update the kernel entry for the new release/build.
4. Save and close the file.
5. Repeat step 2 through step 4 for the /flash/boot/loader.conf.local file.
6. Update the release/build entry in the ns.conf file.
7. Restart the appliance.

Since I upgraded the NetScaler release on the appliance, the LCD display on the front panel of the appliance displays the out of service message or does not display anything. How can I resolve this issue?

Run the following command from the appliance’s shell prompt:

/netscaler/nslcd –k

I have upgraded the NetScaler release/build. However, after the upgrade process, the appliance fails to start. Can I downgrade the appliance’s software to the previous release/build?

Yes. You can start the appliance with the kernel.old kernel file. When you restart the appliance, press the F1 key as soon as the appliance console displays the Press F1 message. Type kernel.old and press Enter.

After upgrading the NetScaler release on the appliance, I accidently deleted the kernel file from the /flash directory. As a result, I am not able to start the appliance. Is there a method for starting the appliance in this situation?

Yes. You can start the appliance by using the kernel.GENERIC kernel file, as follows:

1. When you restart the appliance, press the F1 key as soon as the appliance console displays the Press F1 message.
2. Type kernel.GENERIC and press Enter.
3. Login as the root user.
4. Reinstall the NetScaler release.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Restart the appliance.

I have received a NetScaler appliance with the latest NetScaler release installed on it. However, I want to downgrade the software release. Can I do so?

No. If you attempt to downgrade the software release, the appliance might not work as expected, because the ns.conf file of the later release might not be compatible with the earlier release, and the appliance might restore to the factory settings.

When downgrading the NetScaler release, I followed the instructions. However, the appliance displays the following message:

```
root@LBCOL03B# ./installns
installns version (10.0-47.7) kernel (ns-10.0-47.7.gz)
Note:
Installation may pause for up to 3 minutes while data is written to the flash.
Caution:
Do not interrupt the installation process.
Doing so may cause the system to become unusable.
Installation will proceed in 5 seconds, CTRL-C to abort
No Valid Netscaler Version Detected
root@LBCOL03B#
```

Am I doing something incorrectly?

This issue could be the result of incorrect version information in the ns.conf file. To resolve this issue, open the ns.conf file in a text editor, such as the vi editor. Update the release-specific entry in the ns.conf file to #NS<Release_No> Build <Build_No>. Here, <Release_No> is the NetScaler release number to which you want to downgrade the software, and the <Build_No> is the build number of the software release to which you want to downgrade the software.

After upgrading the appliance software to NetScaler release 10.0, I am not able to log on to the appliance, and the following message is appears:

```
login: nsroot
Password:
connect: No such file or directory
nsnet_connect: No such file or directory
Login incorrect
```

I tried to resolve this issue by using the password recovery procedure, but I was not successful. Have I done something incorrectly?

You cannot resolve this issue by using the password recovery procedure. NetScaler releases 8.0 and later use the new licensing system, based on the Imgrd daemon, which runs during the startup procedure. For this daemon to work properly, the host name of NetScaler appliance, which is set in the /nsconfig/rc.conf file, must be resolved by a name server to the NetScaler IP address. Alternately, you can create a hosts file in the /nsconfig directory and add the 127.0.0.1 <Host_Name> entry in file.

Additionally, make sure that you have copied the license files to the /nsconfig/license/ directory.

During an upgrade of a high availability pair, the following message appears repeatedly:

```
<auth.err> ns sshd[5035]: error: Invalid username or password
```

What could be the reason?

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

This error message appears when the appliances involved in the high availability pairing have either a different NetScaler release or a different builds of the same release installed. The appliances can have different version installed if you have upgraded or downgraded one appliance but not the other.

I want to change the netmask of the NetScaler IP address on a NetScaler appliance. Can I do so without causing an outage?

Changing the netmask of the NetScaler IP might result in a short outage. Make sure that you change the netmask on the secondary appliance, and then break the high availability pairing. Check the functionality of the appliance. If everything works as expected, rebuild the high availability pairing.

To change the netmask on the appliance, run the configns command from the CLI prompt, and then choose the second option in the menu.

I have configured a High Availability pair of NetScaler appliances. After upgrading the software release from a beta release to a final release, I noticed that some of the appliance configurations are missing. Can I retrieve the lost configurations?

You can use the following procedure to restore the configuration:

1. Log on to the command line interface of the primary appliance.
2. Run the following commands:

   save config

   shell

   #cp /nsconfig/ns.conf /nsconfig/ns.conf.bkup

   The ns.conf.bkup file is a backup for the running configuration.

3. Upgrade software of both the appliances to the final release.
4. Log on to the command line interface of the primary appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler Solutions

May 13, 2016

NetScaler solutions simplify the task of setting up frequently deployed configurations. Check this space from time to time for additional solutions.

This section includes the following solutions:

- Solutions for Telecom Service Providers
- Setting Up NetScaler for XenApp/XenDesktop
- RISE Integration: NetScaler ADC and Cisco Nexus 7000 Series Switch
- Global Server Load Balancing (GSLB) Powered Zone Preference

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Solutions for Telecom Service Providers

May 13, 2016

Information and Communication Technology (ICT) is about bringing the Internet user closer to the apps and data. The latest datacenter technologies have enabled the user, apps and data to be located anywhere. A user can access apps and data from the office or from home, or from a location such as an airport. The apps and data can be located either on the enterprise's premises, in a public or private cloud, or on a hybrid host. The result has been on only increased productivity, but also reduced costs of ownership and maintenance.

Service provides offer the core infrastructure needed for carrying the user's apps and data over the network. Because the core infrastructure serves millions of subscribers and a wide variety of apps and data, requirements for scale and protocol support are very high. The core infrastructure handles two major types of traffic: data plane and control plane. Each of these planes has its own scale and protocol-support requirements.

The data plane is the part of the core infrastructure that carries user apps and data from end to end, that is, between end-user equipment and the application server. The number of users accessing apps and data is in the thousands of millions, so throughput and IP-addressing requirements are very high. Every user in the network must be uniquely identifiable. Only then can the service provider control the traffic, monitor network usage, deliver user-specific services, and log information correctly. Many of the today's client devices and application servers support IPv6 natively. The core infrastructure must not only support a mix of IPv4 and IPv6 clients and servers, but also provide the technologies for cross-communication between IPv4 and IPv6. Finally, a service provider is measured by the quality of service (directly related to end-user experience) and the availability of service without disruptions. The data plane should be resilient enough to provide both quality and availability at the same time.

The control-plane infrastructure manages user traffic and maintains the business and network operations services. The most important of the many protocols that run in this plane are Diameter, Radius, and SMPP. Diameter is a base protocol over which several other function-specific protocols have been developed. For example:

- Gx interface between the Policy and Charging Enforcement Function (PCEF) and the Policy and Charging Rules Function (PCRF)
- Gy interface between the Online Charging System (OCS) and the Cisco Packet Data Network Gateway (PGW)/Policy and Charging Enforcement Function (PCEF)

The volume of control plane traffic is in direct proportion to user activity. To manage the control plane traffic, service providers use several ADC functionalities, such as load balancing and content switching. They need fine-grain control of control plane traffic, which equals data-plane traffic in complexity.

Service providers must meet demanding service-level agreements (SLAs), and are scrutinized thoroughly by regulators for compliance. Adhering to requirements while managing the data and control plane traffic requires a service provider to keep its infrastructure nimble, within budget, easily upgradable, and flexible. As the most powerful and advanced ADCs in the market today, Citrix NetScaler products are a natural fit for the service-provider environment.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Large Scale NAT

May 13, 2016

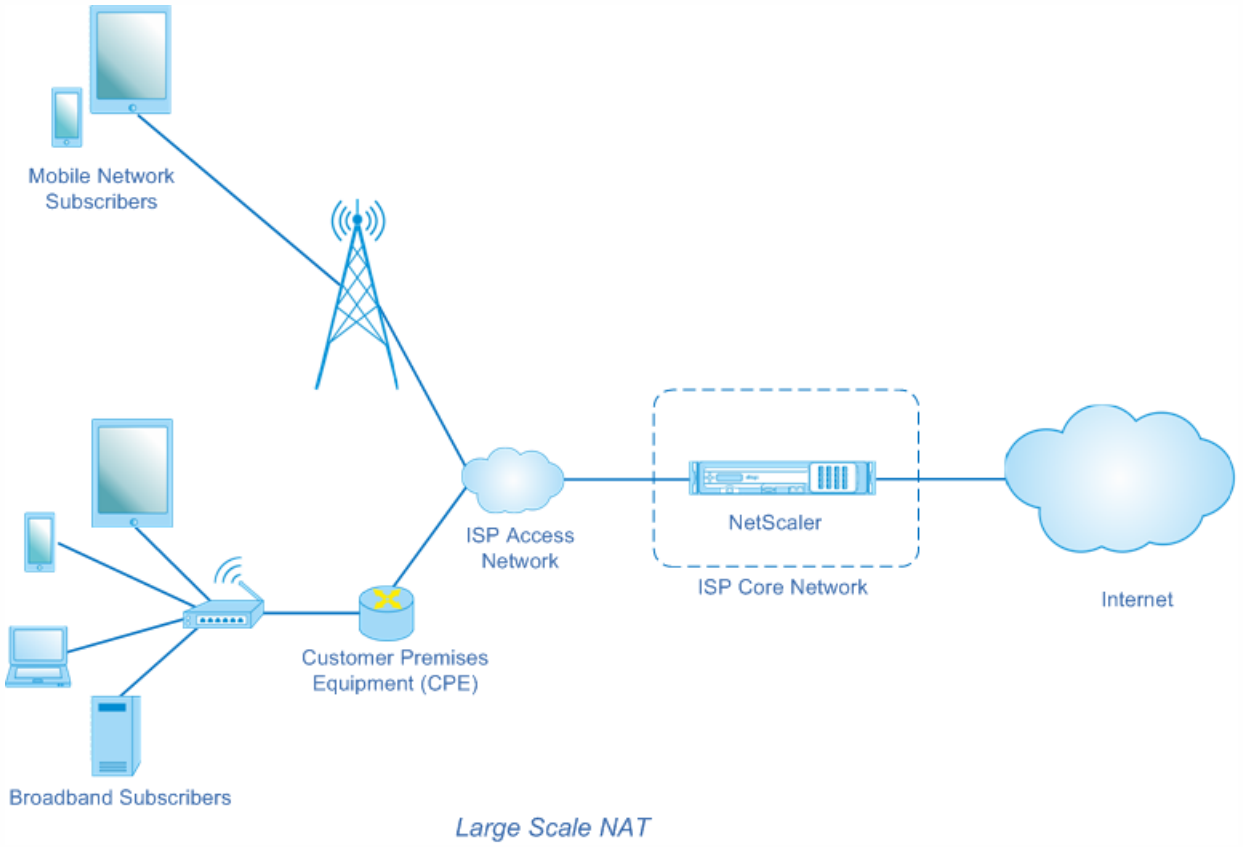
The Internet's phenomenal growth has resulted in a shortage of public IPv4 addresses. Large Scale NAT (LSN/CGNAT) provides a solution to this issue, maximizing the use of available public IPv4 addresses by sharing a few public IPv4 addresses among a large pool of Internet users.

LSN translates private IPv4 addresses into public IPv4 addresses. It includes network address and port translation methods to aggregate many private IP addresses into fewer public IPv4 addresses. LSN is designed to handle NAT on a large scale. The NetScaler LSN feature is very useful for Internet Service Providers (ISPs) and carriers providing millions of translations to support a large number of users (subscribers) and at very high throughput.

LSN Architecture

The LSN architecture of an ISP using Citrix products consists of subscribers (Internet users) in private address spaces accessing the Internet through a NetScaler appliance deployed in ISP's core network. Subscribers are connected to the ISP through the ISP's access network. Usually, subscribers for commercial use of the Internet are directly connected to the ISP's access network. Serving those subscribers requires only one level of NAT (NAT44).

Noncommercial subscribers, however, are typically behind customer-premises equipment (CPE), such as routers and modems, that also implements NAT. These two levels of NAT create the NAT444 model. Deploying a NetScaler appliance in an ISP's core network for LSN functionality is transparent to the subscribers and requires no configuration changes to subscribers or the CPEs.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

The NetScaler appliance receives all subscriber packets destined to the Internet. The appliance is configured with a pool of pre-defined NAT IP addresses to use for LSN. The NetScaler appliance uses its LSN feature to translate the source IP address (private) and port of the packet to the NAT IP address (public) and NAT port, and then sends the packet to its destination on the Internet. The appliance maintains a record of all active sessions that use the LSN feature. These sessions are called LSN sessions. The NetScaler appliance also maintains the mappings between subscriber IP address and port, and NAT IP address and port, for each session. These mappings are called LSN mappings. From LSN sessions and LSN mappings, the NetScaler appliance recognizes a response packet (received from the Internet) belonging to a particular session. The appliance translates the destination IP address and port of the response packet from NAT IP address:port to the subscriber IP address:port, and sends the translated packet to the subscriber.

LSN Features Supported on NetScaler Appliance

The following describes some of the LSN features supported on NetScaler appliance:

NAT Resource Allocation

The NetScaler appliance allocates NAT IP addresses and ports, from its pre-defined NAT resource pool, to subscribers to translate their packets for transmission to external hosts (Internet). The NetScaler appliance supports the following types of NAT IP address and port allocation for subscribers:

- **Deterministic.** The NetScaler appliance allocates a NAT IP address and a block of ports to each subscriber. The appliance sequentially allocates NAT resources to these subscribers. It assigns the first block of ports on the beginning NAT IP address to the beginning subscriber IP address. The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. At that point, the first port block on the next NAT address is assigned to the subscriber, and so on.

  The NetScaler appliance logs the allocated NAT IP address and the port block for a subscriber. For a connection, a subscriber can be identified just by its mapped NAT IP address and port block. Because of this reason, the NetScaler appliance does not log any LSN session created or deleted. If the entire block of ports is being used, the NetScaler appliance drops any new connection from the subscriber.

- **Dynamic.** The NetScaler appliance allocates a random NAT IP address and a port from the LSN NAT pool for a subscriber's connection. When port block allocation is enabled in the configuration, the NetScaler allocates a random NAT IP address and a block of ports for a subscriber when it initiates a connection for the first time. The NetScaler appliance then allocates this NAT IP address and one of the ports from the allocated block to each subsequent connection from this subscriber. If the entire block of ports is being used, the appliance allocates a new random port block to the subscriber when it initiates a new connection. One of the port in the new port block is allocated for the new connection.

IP Pooling

The following NAT resource allocation options are available for subsequent sessions of a subscriber who was allocated a random NAT IP address and port for an existing session.

- **Paired.** The NetScaler appliance allocates the same NAT IP address for all sessions associated with the same subscriber. When no more ports are available for that address, the appliance drops any new connections from the subscriber. This option is needed for proper functioning of certain applications that require creation of multiple sessions on the same source IP address (for example in peer-to-peer applications that use RTP or RTCP protocol.
- **Random.** The NetScaler appliance allocates random NAT IP addresses, from the pool, for different sessions associated with the same subscriber.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Reusing LSN Mappings

The NetScaler appliance can reuse an existing LSN map for new connections originating from the same subscriber IP address and port. The NetScaler LSN feature supports the following types of LSN mapping reuse:

1. **Endpoint Independent.** The NetScaler appliance reuses the LSN mapping for subsequent packets sent from the same subscriber IP address and port (X:x) to any external IP address and port. This type of LSN map reuse is useful for proper functioning of VOIP and peer-to-peer applications.
2. **Address dependent.** The NetScaler appliance reuses the LSN mapping for subsequent packets sent from the same subscriber IP address and port (X:x) to the same external IP address (Y), regardless of the external port.
3. **Address port dependent.** The NetScaler appliance reuses the LSN mapping for subsequent packets sent from the same internal IP address and port (X:x) to the same external IP address and port (Y:y) while the mapping is still active.

LSN Filtering

The NetScaler appliance can filter packets from external hosts based on the active LSN sessions and LSN mappings. Consider an example of an LSN mapping that includes the mapping of subscriber IP:port (X:x), NAT IP:port (N:n), and external host IP:port (Y:y). The NetScaler LSN feature supports the following types of filtering:

1. **Endpoint Independent.** The NetScaler appliance filters out only those packets that are not destined to NAT IP:port (N:n), which represents subscriber IP:port (X:x), regardless of the external host IP address and port source (Z:z). The NetScaler appliance forwards any packets destined to X:x. In other words, sending packets from the subscriber to any external IP address is sufficient to allow packets from any external host to the subscriber. This type of filtering is useful for proper functioning of VOIP and peer-to-peer applications.
2. **Address dependent.** The NetScaler appliance filters out packets not destined to NAT IP:port (N:n), which represents subscriber IP:port (X:x). In addition, the appliance filters out packets from external host IP address and port (Y:y) destined for N:n if the subscriber has not previously sent packets to Y:anyport (external port independent). In other words, receiving packets from a specific external host requires that the subscriber first send packets to that specific external host's IP address.
3. **Address port dependent.** The NetScaler appliance filters out packets not destined to NAT IP:port (N:n), which represents subscriber IP:port (X:x). In addition, the appliance filters out packets from external host IP address and port (Y:y) destined for N:n if the subscriber has not previously sent packets to Y:y. In other words, receiving packets from a specific external host requires that the subscriber first send packets to that specific external IP address and port.

Quotas

The NetScaler appliance can limit the number of NAT ports and sessions for each subscriber to ensure fair distribution of resources among subscribers. The NetScaler appliance can also limit the number of session for a subscriber group to ensure fair distribution of resources among different subscriber groups.

- **Port quota.** The NetScaler appliance can limit the LSN NAT ports to be used at a time by each subscriber for a specified protocol. For example, you could limit each subscriber to a maximum of 500 TCP NAT ports. When the LSN NAT mappings for a subscriber reach the limit, the NetScaler appliance does not allocate additional NAT ports of the specified protocol to that subscriber.
- **Subscriber Session Limit.** The number of concurrent session for a subscriber can be more than it port quota. The NetScaler appliance can limit the LSN sessions allowed for each subscriber for a specified protocol. When the number of LSN sessions reaches the limit for a subscriber, the NetScaler appliance does not allow the subscriber to open additional sessions of the specified protocol.
- **Group Session Limit.** The NetScaler appliance can limit the total number of LSN sessions allowed for a subscriber group for a specified protocol. When the total number of LSN sessions reaches the limit for a group for a specified protocol,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the NetScaler appliance does not allow any subscriber of the group to open additional sessions of the specified protocol. For example, You limit a group to a maximum of 10000 UDP sessions. When the total number of UDP sessions for this group reaches 10000, the NetScaler appliance does not allow any subscriber of the group to open additional UDP sessions.

Application Layer Gateways

For some Application layer protocols, the IP addresses and protocol port numbers are also communicated in the packet's payload. Application Layer Gateway for a protocol parses the packet’s payload and does necessary changes to ensure that the protocol continues to work over LSN.

The NetScaler appliance supports ALG for the following protocols:

- FTP
- ICMP
- TFTP
- PPTP
- SIP
- RTSP

Hairpin Support

The NetScaler appliance supports communication between subscribers or internal hosts using NAT IP addresses. This type of communication between two subscribers using NAT IP addresses is called hairpin flow. Hairpin flow is enabled by default, and you cannot disable it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Points to Consider before Configuring LSN

Apr 26, 2017

Consider the following points before configuring LSN on a NetScaler appliance:

- Make sure that you understand the different components of Large Scale NAT, described in RFCs 6888, 5382, 5508, and 4787.
- Endpoint independent mapping (EIM) and endpoint independent filtering (EIF) are disabled by default. These options must be enabled for proper functioning of VoIP and peer-to-peer (P2P) applications.
- **Logging LSN**: Following are the consideration points for logging LSN information:
  - Citrix recommends logging the LSN information on external log servers instead of on the NetScaler appliance. Logging on external servers facilitates optimal performance when the appliance creates large numbers of LSN log entries (in order of millions).
  - Citrix recomends using SYSLOG over TCP, or NSLOG. By default SYSLOG uses UDP, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data.
  - The following limitations apply to SYSLOG over TCP:
    - The Syslog over TCP solution does not provide authentication, integrity check, and privacy.
    - The NetScaler appliance relies on the TCP protocol to provide confirmation of SYSLOG message delivery to external log servers.
- **High Availability**: Following are the consideration points for high availability of NetScaler appliances for LSN:
  - Citrix recommends configuring the LSN feature in a high availability deployment of two NetScaler appliances for uninterrupted and seamless operation of all LSN sessions.
  - In a high availability deployment, Citrix recommends:
    - Setting the SYNC VLAN parameter for dedicating a VLAN for all HA related communication.
    - Synchronizing the symmetric RSS key of the primary node to the secondary node for stateful synchronization of a large number of LSN mappings and sessions.
    - Binding the subnet of LSN IP addresses to a VLAN to avoid flooding of GARP broadcasts on all VLANs after a failover.
  - In a high availability deployment of NetScaler appliances, ALG-related sessions are not mirrored to the secondary appliance.
- **Application Layer Gateways (ALGs)**: Following are the consideration points related for ALGs on a NetScaler appliance:
  - The following are not supported for SIP ALG:
    - Multicast IP addresses
    - Encrypted SDP
    - SIP messages over TLS
    - FQDN translation in SIP messages
    - Authentication of SIP messages
    - Traffic domains, admin partitions, and NetScaler clusters.
    - SIP messages with multipart bodies.
  - The following are not supported for RTSP ALG:
    - Multicast RTSP sessions
    - RTSP session over UDP
    - NetScaler traffic domains, admin partitions, and NetScaler clusters
  - The NetScaler appliance does not support ALG for the IPSec protocol.
- If you disable the LSN feature when some LSN sessions exist on the NetScaler appliance, these sessions continue to exist for the duration of the configured timeout interval.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- LSN takes precedence over RNAT. If a packet from a specified LSN subscriber also matches a RNAT rule, the packet is translated according to the LSN configuration.
- Forwarding of packets related only to the LSN sessions is based on the NetScaler appliance's routing table.
- Unlike with subnet IP addresses, selection of an LSN NAT IP address for a subscriber's connection is not based on the routing entry for the destination IP address.
- For inbound packets, static LSN mappings take precedence over dynamic LSN mappings.
- For outbound packets, LSN application profiles take precedence over static mapping.
- When a large number of LSN sessions (> 1 million) exist on the NetScaler appliance, Citrix recommends displaying selected LSN sessions instead of all of them. In the command line interface or the configuration utility, use the selection parameters for showing LSN session operation.
- LSN is not supported in a NetScaler cluster.
- To reduce the amount of active memory allocated to the LSN feature, you must warm restart the NetScaler appliance after changing the configured-memory setting. Without a warm restart, you can only increase the amount of active memory.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuration Steps for LSN

Jan 04, 2016

Configuring LSN on a NetScaler appliance consists of the following tasks:

1. **Set the global LSN parameters.** Global parameters include the amount of NetScaler memory reserved for the LSN feature and synchronization of LSN sessions in a high availability setup.
2. **Create an LSN client entity and bind subscribers to it.** An LSN client entity is a set of subscribers on whose traffic you want the NetScaler appliance to perform LSN. The client entity includes IPv4 addresses and extended ACL rules for identifying subscribers. An LSN client can be bound to only one LSN group. The command line interface has two commands for creating an LSN client entity and binding a subscriber to the LSN client entity. The configuration utility combines these two operations on a single screen.
3. **Create an LSN pool and bind NAT IP addresses to it.** An LSN pool defines a pool of NAT IP addresses to be used by the NetScaler appliance to perform LSN. The pool is assigned parameters, such as port block allocation and NAT type (Deterministic or Dynamic). An LSN pool bound to an LSN group applies to all subscribers of an LSN client entity bound to the same group. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiple LSN pools can be bound to an LSN group. For Dynamic NAT, an LSN pool can be bound to multiple LSN groups. For Deterministic NAT, pools bound to an LSN group cannot be bound to other LSN groups. The command line interface has two commands for creating an LSN pool and binding NAT IP addresses to the LSN pool. The configuration utility combines these two operations on a single screen.
4. **(Optional) Create an LSN Transport Profile for a specified protocol.** An LSN transport profile defines various timeouts and limits, such as maximum LSN sessions and maximum ports usage, that a subscriber can have for a given protocol. You bind an LSN transport profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. A profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN transport profile with default settings for TCP, UDP, and ICMP protocols is bound to an LSN group during its creation. This profile is called default transport profile. An LSN transport profile that you bind to an LSN group overrides the default LSN transport profile for that protocol.
5. **(Optional) Create an LSN Application Profile for a specified protocol and bind a set of destination ports to it.** An LSN application profile defines the LSN mapping and LSN filtering controls of a group for a given protocol and for a set of destination ports. For a set of destination ports, you bind an LSN profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. An LSN application profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN application profile with default settings for TCP, UDP, and ICMP protocols for all destination ports is bound to an LSN group during its creation. This profile is called a default application profile. When you bind an LSN application profile, with a specified set of destination ports, to an LSN group, the bound profile overrides the default LSN application profile for that protocol at that set of destination ports. The command line interface has two commands for creating an LSN application profile and binding a set of destination ports to the LSN application profile. The configuration utility combines these two operations on a single screen.
6. **Create an LSN Group and bind LSN pools, (optional) LSN transport profiles, and (optional) LSN application profiles to the LSN group.** An LSN group is an entity consisting of an LSN client, LSN pool(s), LSN transport profile(s), and LSN application profiles(s). A group is assigned parameters, such as port block size and logging of LSN sessions. The parameter settings apply to all the subscribers of an LSN client bound to the LSN group. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group. For Dynamic NAT, an LSN pool can be bound to multiple LSN groups. For Deterministic NAT, pools bound to an LSN group cannot be bound to other LSN groups. Only one LSN client entity can be bound to an LSN group, and an LSN client entity bound to an LSN group cannot be bound to other LSN groups. The command line interface has two commands

© 1999-2017 Citrix Systems, Inc. All rights reserved.

for creating an LSN group and binding LSN pools, LSN transport profiles, LSN application profiles to the LSN group. The configuration utility combines these two operations in a single screen.

The following table lists the maximum numbers of different LSN entities and bindings that can be created on a NetScaler appliance. These limits are also subject to memory available on the NetScaler appliance.

LSN entities and bindings	Limit
LSN clients	1024
LSN pools	128
LSN groups	1024
Subscriber networks that can be bound to an LSN client	64
Extended ACLs that can be bound to an LSN client	1024
NAT IP addresses in a Pool	4096
LSN pools that can be bound to an LSN group	8
LSN groups that can use the same LSN pool	16
LSN transport profiles that can be bound to an LSN group	3 (one each for TCP, UDP, and ICMP protocols)
LSN groups that can use same LSN transport profile	8
LSN application profiles that can be bound to an LSN group	64
LSN groups that can use same LSN application profile	8
Port ranges that can be bound to an LSN application profile	8

Configuration Using the Command Line Interface

To create an LSN client by using the command line interface

At the command prompt, type:

- **add lsn client** <clientname>
- **show lsn client**

To bind a network address or an ACL rule to an LSN client by using the command line interface

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

- **bind lsn client** <clientname> ((-**network** <ip_addr> [-**netmask** <netmask>] [-**td**<positive_integer>]) | -**aclname** <string>)
- **show lsn client**

To create an LSN pool by using the command line interface

At the command prompt, type:

- **add lsn pool** <poolname> [-**nattype** (**DYNAMIC** | **DETERMINISTIC**)] [-**portblockallocation** (**ENABLED** | **DISABLED**)] [-**portrealloctimeout** <secs>] [-**maxPortReallocTmq** <positive_integer>]
- **show lsn pool**

To bind an IP address range to an LSN pool by using the command line interface

At the command prompt, type:

- **bind lsn pool** <poolname> <lsnip>
- **show lsn pool**

Note: For removing LSN IP addresses from an LSN pool, use the unbind lsn pool command.

To create an LSN transport profile by using the command line interface

At the command prompt, type:

- **add lsn transportprofile** <transportprofilename> <transportprotocol> [-**sessiontimeout** <secs>] [-**finrsttimeout** <secs>] [-**portquota** <positive_integer>] [-**sessionquota** <positive_integer>] [-**portpreserveparity** (**ENABLED** | **DISABLED**)] [-**portpreserverange** (**ENABLED** | **DISABLED**)] [-**syncheck** (**ENABLED** | **DISABLED**)]
- **show lsn transportprofile**

To create an LSN application profile by using the command line interface

At the command prompt, type:

- **add lsn appsprofile** <appsprofilename> <transportprotocol> [-**ippooling** (**PAIRED** | **RANDOM**)] [-**mapping** <mapping>] [-**filtering** <filtering>][-**tcpproxy** (**ENABLED** | **DISABLED**)] [-**td** <positive_integer>]
- **show lsn appsprofile**

To bind an application protocol port range to an LSN application profile by using the command line interface

At the command prompt, type:

- **bind lsn appsprofile** <appsprofilename> <lsnport>
- **show lsn appsprofile**

To create an LSN group by using the command line interface

At the command prompt, type:

- **add lsn group** <groupname> -**clientname** <string> [-**nattype** (**DYNAMIC** |**DETERMINISTIC**)] [-**portblocksize** <positive_integer>] [-**logging** (**ENABLED** | **DISABLED**)] [-**sessionLogging** (**ENABLED** | **DISABLED**)]

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

[-**sessionSync** (**ENABLED** | **DISABLED**)] [-**snmptraplimit** <positive_integer>] [-**ftp** (**ENABLED** | **DISABLED**)]

- **show lsn group**

To bind LSN profiles and LSN pools to an LSN group by using the command line interface

At the command prompt, type:

- **bind lsn group** <groupname> (-**poolname** <string> | -**transportprofilename** <string> | -**appsprofilename** <string>)
- **show lsn group**

Configuration Using the Configuration Utility

To configure an LSN client and bind an IPv4 network address or an ACL rule by using the configuration utility

Navigate to **System** > **Large Scale NAT** > **Clients**, and add a client and then bind an IPv4 network address or an ACL rule to the client.

To configure an LSN pool and bind NAT IP addresses by using the configuration utility

Navigate to **System > Large Scale NAT** > **Pools**, and add a pool and then bind an NAT IP address or a range of NAT IP addresses to the pool.

To configure an LSN transport profile by using the configuration utility

1. Navigate to **System > Large Scale NAT > Profiles**.
2. On the details pane, click **Transport** tab, and then add a transport profile.

To configure an LSN application profile by using the configuration utility

1. Navigate to **System > Large Scale NAT > Profiles**.
2. On the details pane, click **Application** tab, and then add an application profile.

To configure an LSN group and bind an LSN client, pools, transport profiles, and application profiles by using the configuration utility

Navigate to **System > Large Scale NAT > Groups**, and add a group and then bind an LSN client, pools, transport profiles, and application profiles to the group.

Parameter Descriptions (of commands listed in the CLI procedure)

add lsn client

clientname

Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1').

This is a mandatory argument. Maximum Length: 127

Parameter Descriptions (of commands listed in the CLI procedure)

bind lsn client

clientname

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1').

This is a mandatory argument. Maximum Length: 127

network

IPv4 address(es) of the LSN subscriber(s) or subscriber network(s) on whose traffic you want the NetScaler appliance to perform Large Scale NAT.

netmask

Subnet mask for the IPv4 address specified in the Network parameter.

Default value: 255.255.255.255

td

ID of the traffic domain on which this subscriber or the subscriber network (as specified by the network parameter) belongs.

If you do not specify an ID, the subscriber or the subscriber network becomes part of the default traffic domain.

Default value: 0

Minimum value: 0

Maximum value: 4094

aclname

Name(s) of any configured extended ACL(s) whose action is ALLOW. The condition specified in the extended ACL rule identifies the traffic from an LSN subscriber for which the NetScaler appliance is to perform large scale NAT. Maximum Length: 127

Parameter Descriptions (of commands listed in the CLI procedure)

add lsn pool

poolname

Name for the LSN pool. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN pool is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn pool1" or 'lsn pool1').

This is a mandatory argument. Maximum Length: 127

nattype

Type of NAT IP address and port allocation (from the LSN pools bound to an LSN group) for subscribers (of the LSN client entity bound to the LSN group):

Available options function as follows:

- **Deterministic**—Allocate a NAT IP address and a block of ports to each subscriber (of the LSN client bound to the LSN group). The NetScaler appliance sequentially allocates NAT resources to these subscribers. The NetScaler appliance assigns the first block of ports (block size determined by the port block size parameter of the LSN group) on the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

beginning NAT IP address to the beginning subscriber IP address. The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. In this case, the first port block on the next NAT address is used for the subscriber, and so on. Because each subscriber now receives a deterministic NAT IP address and a block of ports, a subscriber can be identified without any need for logging. For a connection, a subscriber can be identified based only on the NAT IP address and port, and the destination IP address and port.

- **Dynamic**—Allocate a random NAT IP address and a port from the LSN NAT pool for a subscribers connection. If port block allocation is enabled (in LSN pool) and a port block size is specified (in the LSN group), the NetScaler appliance allocates a random NAT IP address and a block of ports for a subscriber when it initiates a connection for the first time. The appliance allocates this NAT IP address and a port (from the allocated block of ports) for different connections from this subscriber. If all the ports are allocated (for different subscribers connections) from the subscribers allocated port block, the appliance allocates a new random port block for the subscriber. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group.
  Possible values: DYNAMIC, DETERMINISTIC

  Default value: DYNAMIC

portblockallocation

Allocate a random NAT port block, from the available NAT port pool of an NAT IP address, for each subscriber when the NAT allocation is set as Dynamic NAT. For any connection initiated from a subscriber, the NetScaler appliance allocates a NAT port from the subscribers allocated NAT port block to create the LSN session.

You must set the port block size in the bound LSN group. For a subscriber, if all the ports are allocated from the subscribers allocated port block, the NetScaler appliance allocates a new random port block for the subscriber.

For Deterministic NAT, this parameter is enabled by default, and you cannot disable it.

Possible values: ENABLED, DISABLED

Default value: DISABLED

portreallocTimeout

The waiting time, in seconds, between deallocating LSN NAT ports (when an LSN mapping is removed) and reallocating them for a new LSN session. This parameter is necessary in order to prevent collisions between old and new mappings and sessions. It ensures that all established sessions are broken instead of redirected to a different subscriber. This is not applicable for ports used in:

- Deterministic NAT
- Address-Dependent filtering and Address-Port-Dependent filtering
- Dynamic NAT with port block allocation

In these cases, ports are immediately reallocated.

Default value: 0

Maximum value: 600

maxPortReallocTmq

Maximum number of ports for which the port reallocation timeout applies for each NAT IP address. In other words, the maximum deallocated-port queue size for which the reallocation timeout applies for each NAT IP address.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

When the queue size is full, the next port deallocated is reallocated immediately for a new LSN session.

Default value: 65536

Maximum value: 65536

Parameter Descriptions (of commands listed in the CLI procedure)

bind lsn pool

poolname

Name for the LSN pool. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN pool is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn pool1" or 'lsn pool1').

This is a mandatory argument. Maximum Length: 127

lsnip

IPv4 address or a range of IPv4 addresses to be used as NAT IP address(es) for LSN.

After the pool is created, these IPv4 addresses are added to the NetScaler appliance as NetScaler owned IP address of type LSN. An LSN IP address associated with an LSN pool cannot be shared with other LSN pools. IP addresses specified for this parameter must not already exist on the NetScaler appliance as any NetScaler owned IP addresses. In the command line interface, separate the range with a hyphen. For example: 10.102.29.30-10.102.29.189. You can later remove some or all the LSN IP addresses from the pool, and add IP addresses to the LSN pool.

Parameter Descriptions (of commands listed in the CLI procedure)

add lsn transportprofile

transportprofilename

Name for the LSN transport profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN transport profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn transport profile1" or 'lsn transport profile1').

This is a mandatory argument. Maximum Length: 127

transportprotocol

Protocol for which to set the LSN transport profile parameters.

This is a mandatory argument.

Possible values: TCP, UDP, ICMP

sessiontimeout

Timeout, in seconds, for an idle LSN session. If an LSN session is idle for a time that exceeds this value, the NetScaler appliance removes the session.

This timeout does not apply for a TCP LSN session when a FIN or RST message is received from either of the endpoints.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Default value: 120

Minimum value: 60

finrsttimeout

Timeout, in seconds, for a TCP LSN session after a FIN or RST message is received from one of the endpoints.

f a TCP LSN session is idle (after the NetScaler appliance receives a FIN or RST message) for a time that exceeds this value, the NetScaler appliance removes the session.

Since the LSN feature of the NetScaler appliance does not maintain state information of any TCP LSN sessions, this timeout accommodates the transmission of the FIN or RST, and ACK messages from the other endpoint so that both endpoints can properly close the connection.

Default value: 30

portquota

Maximum number of LSN NAT ports to be used at a time by each subscriber for the specified protocol. For example, each subscriber can be limited to a maximum of 500 TCP NAT ports. When the LSN NAT mappings for a subscriber reach the limit, the NetScaler appliance does not allocate additional NAT ports for that subscriber.

Default value: 0

Minimum value: 0

Maximum value: 65535

sessionquota

Maximum number of concurrent LSN sessions allowed for each subscriber for the specified protocol. When the number of LSN sessions reaches the limit for a subscriber, the NetScaler appliance does not allow the subscriber to open additional sessions.

Default value: 0

Minimum value: 0

Maximum value: 65535

portpreserveparity

Enable port parity between a subscriber port and its mapped LSN NAT port. For example, if a subscriber initiates a connection from an odd numbered port, the NetScaler appliance allocates an odd numbered LSN NAT port for this connection. You must set this parameter for proper functioning of protocols that require the source port to be even or odd numbered, for example, in peer-to-peer applications that use RTP or RTCP protocol.

Possible values: ENABLED, DISABLED

Default value: DISABLED

portpreserverange

If a subscriber initiates a connection from a well-known port (0-1023), allocate a NAT port from the well-known port range (0-1023) for this connection. For example, if a subscriber initiates a connection from port 80, the NetScaler appliance can allocate port 100 as the NAT port for this connection.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

This parameter applies to dynamic NAT without port block allocation. It also applies to Deterministic NAT if the range of ports allocated includes well-known ports.

When all the well-known ports of all the available NAT IP addresses are used in different subscribers connections (LSN sessions), and a subscriber initiates a connection from a well- known port, the NetScaler appliance drops this connection.

Possible values: ENABLED, DISABLED

Default value: DISABLED

syncheck

Silently drop any non-SYN packets for connections for which there is no LSN-NAT session present on the NetScaler appliance.

If you disable this parameter, the NetScaler appliance accepts any non-SYN packets and creates a new LSN session entry for this connection.

Following are some reasons for the NetScaler appliance to receive such packets:

- LSN session for a connection existed but the NetScaler appliance removed this session because the LSN session was idle for a time that exceeded the configured session timeout.
- Such packets can be a part of a DoS attack.

Possible values: ENABLED, DISABLED

Default value: ENABLED

Parameter Descriptions (of commands listed in the CLI procedure)

add lsn appsprofile

appsprofilename

Name for the LSN application profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN application profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn application profile1" or 'lsn application profile1').

This is a mandatory argument. Maximum Length: 127

transportprotocol

Name of the protocol for which the parameters of this LSN application profile applies.

This is a mandatory argument.

Possible values: TCP, UDP, ICMP

ippooling

NAT IP address allocation options for sessions associated with the same subscriber.

Available options function as follows:

- **Paired**—The NetScaler appliance allocates the same NAT IP address for all sessions associated with the same

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

subscriber. When all the ports of a NAT IP address are used in LSN sessions (for same or multiple subscribers), the NetScaler appliance drops any new connection from the subscriber.
- **Random**—The NetScaler appliance allocates random NAT IP addresses, from the pool, for different sessions associated with the same subscriber.

This parameter is applicable to dynamic NAT allocation only.

Possible values: PAIRED, RANDOM

Default value: RANDOM

mapping

Type of LSN mapping to apply to subsequent packets originating from the same subscriber IP address and port.

Consider an example of an LSN mapping that includes the mapping of the subscriber IP:port (X:x), NAT IP:port (N:n), and external host IP:port (Y:y).

Available options function as follows:

- **ENDPOINT-INDEPENDENT**—Reuse the LSN mapping for subsequent packets sent from the same subscriber IP address and port (X:x) to any external IP address and port.
- **ADDRESS-DEPENDENT**—Reuse the LSN mapping for subsequent packets sent from the same subscriber IP address and port (X:x) to the same external IP address (Y), regardless of the external port.
- **ADDRESS-PORT-DEPENDENT**—Reuse the LSN mapping for subsequent packets sent from the same internal IP address and port (X:x) to the same external IP address and port (Y:y) while the mapping is still active.

Possible values: ENDPOINT-INDEPENDENT, ADDRESS-DEPENDENT, ADDRESS-PORT-DEPENDENT

Default value: ADDRESS-PORT-DEPENDENT

filtering

Type of filter to apply to packets originating from external hosts.

Consider an example of an LSN mapping that includes the mapping of subscriber IP:port (X:x), NAT IP:port (N:n), and external host IP:port (Y:y).

Available options function as follows:

- **ENDPOINT INDEPENDENT**—Filters out only packets not destined to the subscriber IP address and port X:x, regardless of the external host IP address and port source (Z:z). The NetScaler appliance forwards any packets destined to X:x. In other words, sending packets from the subscriber to any external IP address is sufficient to allow packets from any external hosts to the subscriber.
- **ADDRESS DEPENDENT**—Filters out packets not destined to subscriber IP address and port X:x. In addition, the appliance filters out packets from Y:y destined for the subscriber (X:x) if the client has not previously sent packets to Y:anyport (external port independent). In other words, receiving packets from a specific external host requires that the subscriber first send packets to that specific external host's IP address.
- **ADDRESS PORT DEPENDENT** (the default)—Filters out packets not destined to subscriber IP address and port (X:x). In addition, the NetScaler appliance filters out packets from Y:y destined for the subscriber (X:x) if the subscriber has not previously sent packets to Y:y. In other words, receiving packets from a specific external host requires that the subscriber first send packets first to that external IP address and port.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Possible values: ENDPOINT-INDEPENDENT, ADDRESS-DEPENDENT, ADDRESS-PORT-DEPENDENT

Default value: ADDRESS-PORT-DEPENDENT

tcpproxy

Enable TCP proxy, which enables the NetScaler appliance to optimize the TCP traffic by using Layer 4 features.

Possible values: ENABLED, DISABLED

Default value: DISABLED

td

ID of the traffic domain through which the NetScaler appliance sends the outbound traffic after performing LSN.

If you do not specify an ID, the appliance sends the outbound traffic through the default traffic domain, which has an ID of 0.

Default value: 65535

Maximum value: 65535

Parameter Descriptions (of commands listed in the CLI procedure)

bind lsn appsprofile

appsprofilename

Name for the LSN application profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN application profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn application profile1" or 'lsn application profile1').

This is a mandatory argument. Maximum Length: 127

lsnport

Port numbers or range of port numbers to match against the destination port of the incoming packet from a subscriber. When the destination port is matched, the LSN application profile is applied for the LSN session. Separate a range of ports with a hyphen. For example, 40-90.

Parameter Descriptions (of commands listed in the CLI procedure)

add lsn group

groupname

Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1').

This is a mandatory argument. Maximum Length: 127

clientname

Name of the LSN client entity to be associated with the LSN group. You can associate only one LSN client entity with an

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

LSN group. You cannot remove this association or replace with another LSN client entity once the LSN group is created.

This is a mandatory argument. Maximum Length: 127

nattype

Type of NAT IP address and port allocation (from the bound LSN pools) for subscribers:

Available options function as follows:

- **Deterministic**—Allocate a NAT IP address and a block of ports to each subscriber (of the LSN client bound to the LSN group). The NetScaler appliance sequentially allocates NAT resources to these subscribers. The NetScaler appliance assigns the first block of ports (block size determined by the port block size parameter of the LSN group) on the beginning NAT IP address to the beginning subscriber IP address. The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. In this case, the first port block on the next NAT address is used for the subscriber, and so on. Because each subscriber now receives a deterministic NAT IP address and a block of ports, a subscriber can be identified without any need for logging. For a connection, a subscriber can be identified based only on the NAT IP address and port, and the destination IP address and port.
- **Dynamic**—Allocate a random NAT IP address and a port from the LSN NAT pool for a subscriber's connection. If port block allocation is enabled (in LSN pool) and a port block size is specified (in the LSN group), the NetScaler appliance allocates a random NAT IP address and a block of ports for a subscriber when it initiates a connection for the first time. The appliance allocates this NAT IP address and a port (from the allocated block of ports) for different connections from this subscriber. If all the ports are allocated (for different subscribers connections) from the subscribers allocated port block, the appliance allocates a new random port block for the subscriber.

Possible values: DYNAMIC, DETERMINISTIC

Default value: DYNAMIC

portblocksize

Size of the NAT port block to be allocated for each subscriber.

To set this parameter for Dynamic NAT, you must enable the port block allocation parameter in the bound LSN pool. For Deterministic NAT, the port block allocation parameter is always enabled, and you cannot disable it.

In Dynamic NAT, the NetScaler appliance allocates a random NAT port block, from the available NAT port pool of an NAT IP address, for each subscriber. For a subscriber, if all the ports are allocated from the subscribers allocated port block, the appliance allocates a new random port block for the subscriber.

logging

Log mapping entries and sessions created or deleted for this LSN group. The NetScaler appliance logs LSN sessions for this LSN group only when both logging and session logging parameters are enabled.

The appliance uses its existing syslog and audit log framework to log LSN information. You must enable global level LSN logging by enabling the LSN parameter in the related NSLOG action and SYLOG action entities. When the Logging parameter is enabled, the NetScaler appliance generates log messages related to LSN mappings and LSN sessions of this LSN group. The appliance then sends these log messages to servers associated with the NSLOG action and SYSLOG actions entities.

A log message for an LSN mapping entry consists of the following information:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- NSIP address of the NetScaler appliance
- Time stamp
- Entry type (MAPPING or SESSION)
- Whether the LSN mapping entry is created or deleted
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
  - Destination IP address and port are not logged for Endpoint-Independent mapping
  - Only Destination IP address (and not port) is logged for Address-Dependent mapping
  - Destination IP address and port are logged for Address-Port-Dependent mapping

Possible values: ENABLED, DISABLED

Default value: DISABLED

sessionLogging

Log sessions created or deleted for the LSN group. The NetScaler appliance logs LSN sessions for this LSN group only when both logging and session logging parameters are enabled.

A log message for an LSN session consists of the following information:

- NSIP address of the NetScaler appliance
- Time stamp
- Entry type (MAPPING or SESSION)
- Whether the LSN session is created or removed
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

Possible values: ENABLED, DISABLED

Default value: DISABLED

sessionSync

In a high availability (HA) deployment, synchronize information of all LSN sessions related to this LSN group with the secondary node. After a failover, established TCP connections and UDP packet flows are kept active and resumed on the secondary node (new primary).

For this setting to work, you must enable the global session synchronization parameter.

Possible values: ENABLED, DISABLED

Default value: ENABLED

snmptraplimit

Maximum number of SNMP Trap messages that can be generated for the LSN group in one minute.

Default value: 100

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Minimum value: 0

Maximum value: 10000

ftp

Enable Application Layer Gateway (ALG) for the FTP protocol. For some application-layer protocols, the IP addresses and protocol port numbers are usually communicated in the packets payload. When acting as an ALG, the NetScaler changes the packets payload to ensure that the protocol continues to work over LSN.

Note: The NetScaler appliance also includes ALG for ICMP and TFTP protocols. ALG for the ICMP protocol is enabled by default, and there is no provision to disable it. ALG for the TFTP protocol is disabled by default. ALG is enabled automatically for an LSN group when you bind a UDP LSN application profile, with endpoint-independent-mapping, endpoint-independent filtering, and destination port as 69 (well-known port for TFTP), to the LSN group.

Possible values: ENABLED, DISABLED

Default value: ENABLED

Parameter Descriptions (of commands listed in the CLI procedure)

bind lsn group

groupname

Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1').

This is a mandatory argument. Maximum Length: 127

poolname

Name of the LSN pool to bind to the specified LSN group. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group.

For Deterministic NAT, pools bound to an LSN group cannot be bound to other LSN groups. For Dynamic NAT, pools bound to an LSN group can be bound to multiple LSN groups. Maximum Length: 127

transportprofilename

Name of the LSN transport profile to bind to the specified LSN group. Bind a profile for each protocol for which you want to specify settings.

By default, one LSN transport profile with default settings for TCP, UDP, and ICMP protocols is bound to an LSN group during its creation. This profile is called a default transport.

An LSN transport profile that you bind to an LSN group overrides the default LSN transport profile for that protocol. Maximum Length: 127

appsprofilename

Name of the LSN application profile to bind to the specified LSN group. For each set of destination ports, bind a profile for each protocol for which you want to specify settings.

By default, one LSN application profile with default settings for TCP, UDP, and ICMP protocols for all destination ports is

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

bound to an LSN group during its creation. This profile is called a default application profile.

When you bind an LSN application profile, with a specified set of destination ports, to an LSN group, the bound profile overrides the default LSN application profile for that protocol at that set of destination ports. Maximum Length: 127

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Sample LSN Configurations

Aug 17, 2015

The following table shows examples of configuring LSN through command line interface.

Task	Steps
Create a simple LSN configuration with a single subscriber network, single LSN NAT IP address, and default settings.	>set lsn parameter -memLimit 4000 Done >add lsn client LSN-CLIENT-1 Done >bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0 Done >add lsn pool LSN-POOL-1 Done >bind lsn pool LSN-POOL-1 203.0.113.3 Done >add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 Done >bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1 Done
Create an LSN configuration with an extended ACL for identifying LSN subscribers	>add ns acl LSN-ACL-2 ALLOW -srcIP 192.0.2.10-192.0.2.20 Done >apply acls Done >add lsn client LSN-CLIENT-2 Done >bind lsn client LSN-CLIENT-2 –aclname LSN-ACL-2 Done >add lsn pool LSN-POOL-2

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	Done >bind lsn pool LSN-POOL-2 203.0.113.5-203.0.113.10 Done >add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2 Done >bind lsn group LSN-GROUP-2 -poolname LSN-POOL-2 Done
Create an LSN configuration with endpoint-independent mapping for HTTP protocol (port 80) and address-port dependent mapping for SSH protocol (port 22). Also, restrict each subscriber to use a maximum of 1000 NAT ports for TCP protocol and 100 NAT ports for UDP protocol. Restrict each subscriber to have a maximum of 2000 concurrent sessions for TCP protocol. Restrict the group to have a maximum of 30000 concurrent sessions for TCP protocol.	>add lsn client LSN-CLIENT-3 Done >bind lsn client LSN-CLIENT-3 -network 192.0.3.0 -netmask 255.255.255.0 Done >add lsn pool LSN-POOL-3 Done >bind lsn pool LSN-POOL-3 203.0.113.11 Done >add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-3 Done >bind lsn group LSN-GROUP-3 -poolname LSN-POOL-3 Done >add lsn appsprofile LSN-APPS-HTTPPROFILE-3 TCP -mapping ENDPOINT-INDEPENDENT Done >bind lsn appsprofile LSN-APPS-HTTPPROFILE-3 80 Done >bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-HTTPPROFILE-3 Done >add lsn appsprofile LSN-APPS-SSHPROFILE-3 TCP -mapping ADDRESS-PORT-DEPENDENT Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	>bind lsn appsprofile LSN-APPS-SSHPROFILE-3 22 Done >bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-SSHPROFILE-3 Done > add lsn transportprofile LSN-TRANS-PROFILE-TCP-3 TCP -portquota 1000 -sessionquota 2000 -groupSessionLimit 30000 Done > bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-TCP-3 Done >add lsn transportprofile LSN-TRANS-PROFILE-UDP-3 UDP -portquota 100 Done >bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-UDP-3 Done
Create an LSN configuration for a large set of subscribers.	>set lsn parameter -memLimit 22462 Done >add lsn client LSN-CLIENT-4 Done >bind lsn client LSN-CLIENT-4 -network 192.0.4.0 -netmask 255.255.255.0 Done >bind lsn client LSN-CLIENT-4 -network 192.0.5.0 -netmask 255.255.255.0 Done >bind lsn client LSN-CLIENT-4 -network 192.0.6.0 -netmask 255.255.255.0 Done >bind lsn client LSN-CLIENT-4 -network 192.0.7.0 -netmask 255.255.255.0 Done >bind lsn client LSN-CLIENT-4 -network 192.0.8.0 -netmask 255.255.255.0 Done >add lsn pool LSN-POOL-4

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	Done >bind lsn pool LSN-POOL-4 203.0.113.30-203.0.113.40 Done >bind lsn pool LSN-POOL-4 203.0.113.45-203.0.113.50 Done >bind lsn pool LSN-POOL-4 203.0.113.55-203.0.113.60 Done >add lsn group LSN-GROUP-4 -clientname LSN-CLIENT-4 Done >bind lsn group LSN-GROUP-4 -poolname LSN-POOL-4 Done >add lsn appsprofile LSN-APPS-WELLKNOWNPROFILE-4 TCP -mapping ENDPOINT-INDEPENDENT Done >bind lsn appsprofile LSN-APPS-WELLKNOWN-PORTS-PROFILE-4 1- 1023 Done >bind lsn group LSN-GROUP-4 -applicationprofilename LSN-APPS-WELLKNOWN-PORTS-PROFILE-4 Done
Create an LSN configuration with sharing of NAT resources among multiple LSN groups. In this example, LSN pool LSN-POOL-5 is shared with LSN groups LSN-GROUP-5 and LSN-GROUP-6.	>add lsn client LSN-CLIENT-5 Done >bind lsn client LSN-CLIENT-5 -network 192.0.15.0 -netmask 255.255.255.0 Done >add lsn pool LSN-POOL-5 Done >bind lsn pool LSN-POOL-5 203.0.113.12-203.0.113.14 Done >add lsn group LSN-GROUP-5 -clientname LSN-CLIENT-5 Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	>bind lsn group LSN-GROUP-5 -poolname LSN-POOL-5 Done >add lsn client LSN-CLIENT-6 Done >bind lsn client LSN-CLIENT-6 -network 192.0.16.0 -netmask 255.255.255.0 Done >add lsn pool LSN-POOL-6 Done >bind lsn pool LSN-POOL-6 203.0.113.15-203.0.113.18 Done >add lsn group LSN-GROUP-6 -clientname LSN-CLIENT-6 Done >bind lsn group LSN-GROUP-6 -poolname LSN-POOL-6 Done >bind lsn group LSN-GROUP-6 -poolname LSN-POOL-5 Done
Create an LSN configuration with deterministic NAT resource allocation.	>add lsn client LSN-CLIENT-7 Done >bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask 255.255.255.0 Done >add lsn pool LSN-POOL-7 -nattype DETERMINISTIC Done >bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23 Done >add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype DETERMINISTIC -portblocksize 1024 Done >bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7 Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Create an LSN configuration with multiple subscriber networks having the same network address but each network belonging to a different traffic domain. Also, restrict the outbound traffic related to HTTP protocol (port 80), sending it through a particular traffic domain (td 5).	>add lsn client LSN-CLIENT-8 Done >bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0 -td 1 Done >bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0 -td 2 Done >bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0 -td 3 Done >add lsn pool LSN-POOL-8 Done >bind lsn pool LSN-POOL-8 203.0.113.80-203.0.113.86 Done >add lsn group LSN-GROUP-8 -clientname LSN-CLIENT-8 Done >bind lsn group LSN-GROUP-8 -poolname LSN-POOL-8 Done >add lsn appsprofile LSN-APPS-HTTP-PROFILE-8 TCP -td 5 Done >bind lsn appsprofile LSN-APPS-HTTP-PROFILE-8 80 Done >bind lsn group LSN-GROUP-8 -applicationprofilename LSN-APPS-HTTP-PROFILE-8 Done
Create an LSN configuration that restricts the outbound traffic of a specific protocol (TCP), sending it through a particular traffic domain (td 5). With endpoint-independent filtering, receive inbound traffic related to this protocol (TCP) on	>add lsn client LSN-CLIENT-9 Done >bind lsn client LSN-CLIENT-9 -network 192.0.9.0 -netmask 255.255.255.0 -td 1 Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

any traffic domain.	>add lsn pool LSN-POOL-9 Done >bind lsn pool LSN-POOL-9 203.0.113.90 Done >add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9 Done >bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9 Done >add lsn appsprofile LSN-APPS-PROFILE-9 TCP -filtering ENDPOINT-INDEPENDENT -td 5 Done >bind lsn group LSN-GROUP-9 -approfile LSN-APPS-PROFILE-9 Done
Create an LSN configuration that restricts outbound HTTP (port 80) traffic, sending it through a particular traffic domain (td 10). With address-dependent filtering, receive inbound traffic related to this protocol (HTTP) on the specified traffic domain (td 10).	>add lsn client LSN-CLIENT-10 Done >bind lsn client LSN-CLIENT-10 -network 192.0.10.0 -netmask 255.255.255.0 -td 1 Done >add lsn pool LSN-POOL-10 Done >bind lsn pool LSN-POOL-10 203.0.113.100 Done >add lsn group LSN-GROUP-10 -clientname LSN-CLIENT-10 Done >bind lsn group LSN-GROUP-10 -poolname LSN-POOL-10 Done >add lsn appsprofile LSN-APPS-PROFILE-10 TCP -mapping ENDPOINT -INDEPENDENT -filtering ADDRESS-DEPENDENT -td 10 Done >bind lsn appsprofile LSN-APPS-PROFILE-10 80

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done

>bind lsn group LSN-GROUP-10 -approfile LSN-APPS-PROFILE-10

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Static LSN Maps

Aug 17, 2015

The NetScaler appliance supports manual creation of a one-to-one LSN mapping between a subscriber IP address:port and a NAT IP address:port. Static LSN mappings are useful in cases where you want to ensure that the connections initiated to a NAT IP:Port maps to the subscriber IP address:Port. For example, Web servers located in the internal network.

To create a static LSN mapping by using the command line interface

At the command prompt, type:

- add lsn static <name> <transportprotocol> <subscrIP> <subscrPort> [-td <positive_integer>] [<natIP> [<natPort>]] [-destIP <ip_addr> [-dsttd <positive_integer>]]
- show lsn static

To create a static LSN mapping by using the configuration utility

Navigate to System > Large Scale NAT > Static, and add a new static mapping.

Parameter Descriptions (of commands listed in the CLI procedure)

add lsn static

name

Name for the LSN static mapping entry. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn static1" or 'lsn static1'). This is a mandatory argument. Maximum Length: 127

transportprotocol

Protocol for the LSN mapping entry. This is a mandatory argument. Possible values: TCP, UDP, ICMP

subscrIP

IPv4 address of an LSN subscriber for the LSN mapping entry. This is a mandatory argument.

subscrPort

Port of the LSN subscriber for the LSN mapping entry. This is a mandatory argument. Maximum value: 65535

td

ID of the traffic domain to which the subscriber belongs. If you do not specify an ID, the subscriber is assumed to be a part of the default traffic domain. Default value: 0, Minimum value: 0, Maximum value: 4094

natIP

IPv4 address, already existing on the NetScaler appliance as type LSN, to be used as NAT IP address for this mapping entry.

natPort

NAT port for this LSN mapping entry.

destIP

Destination IP address for the LSN mapping entry.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dsttd

ID of the traffic domain through which the destination IP address for this LSN mapping entry is reachable from the NetScaler appliance. If you do not specify an ID, the destination IP address is assumed to be reachable through the default traffic domain, which has an ID of 0. Default value: 0, Minimum value: 0, Maximum value: 4094

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Application Layer Gateways

Aug 17, 2015

For some Application layer protocols, the IP addresses and protocol port numbers are also communicated in the packet's payload. Application Layer Gateway for a protocol parses the packet’s payload and does necessary changes to ensure that the protocol continues to work over LSN.

The NetScaler appliance supports ALG for the following protocols:

- FTP
- ICMP
- TFTP
- PPTP
- SIP
- RTSP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Layer Gateway for FTP, ICMP, and TFTP Protocols

Jan 07, 2016

You can enable or disable ALG for the FTP protocol for an LSN configuration by enabling or disabling the FTP option of the LSN group of the LSN configuration.

ALG for the ICMP protocol is enabled by default, and there is no provision to disable it.

ALG for the TFTP protocol is disabled by default. TFTP ALG is enabled automatically for an LSN configuration when you bind a UDP LSN application profile, with endpoint-independent-mapping, endpoint-independent filtering, and destination port as 69 (well-known port for TFTP), to the LSN group.

Sample LSN Configuration for FTP ALG COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

In the following sample LSN configuration, FTP ALG is enabled for subscribers that have IP address in the range 192.0.2.30-192.0.2.100.

```
> add ns acl LSN-ACL-1 ALLOW -srcIP 192.0.2.30-192.0.2.100

Done

> apply acls

Done

> add lsn client LSN-CLIENT-1

Done

> bind lsn client LSN-CLIENT-1 –aclname LSN-ACL

Done

> add lsn pool LSN-POOL-1

Done

> bind lsn pool LSN-POOL-1 203.0.113.10

Done

> add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -FTP ENABLED

Done

> bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done
```

Sample LSN Configuration for TFTP ALG

COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

In the following sample LSN configuration, endpoint-independent mapping and endpoint-independent filtering are enabled for TFTP proto

```
> add lsn client LSN-CLIENT-2

Done

> bind lsn client LSN-CLIENT-2 -network 198.51.100.0 -netmask 255.255.255.0

Done

> add lsn pool LSN-POOL-2

Done

> bind lsn pool LSN-POOL-2 203.0.113.10-203.0.113.11

Done

> add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2

Done

> bind lsn group LSN-GROUP-2 -poolname pool1 LSN-POOL-2

Done

> add lsn appsprofile LSNAPPSPROFILE-TFTP-2 UDP -mapping ENDPOINT-INDEPENDENT –filtering  ENDPOINT-INDEPENDENT

Done

> bind lsn appsprofile LSNAPPSPROFILE-TFTP-2 69

Done

>bind lsn group LSN-GROUP-1 -applicationprofilename LSNAPPSPROFILE-TFTP-2

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Layer Gateway for PPTP Protocol

Aug 17, 2015

The NetScaler appliance supports Application Layer Gateways (ALGs) for the Point-to-Point Tunneling Protocol (PPTP).

PPTP is a network protocol that enables secure transfer of data from a remote client to an enterprise server by creating a tunnel across TCP/IP-based data networks. PPTP encapsulates PPP packets into IP packets for transmission over the Internet. PPTP establishes a tunnel for each communicating PPTP network server (PNS)-PPTP Access Concentrator (PAC) pair. After the tunnel is set up, enhanced generic routing encapsulation (GRE) is used to exchange PPP packets. A call ID in the GRE header indicates the session to which a particular PPP packet belongs.

The NetScaler appliance recognizes PPTP packets that arrive on the default TCP port, 1723. The appliance parses PPTP control packets, translates the call ID, and assigns a NAT IP address. For two-way data communication between the client and server, the NetScaler appliance creates an LSN session entry based on the server call ID, and an LSN session based on the client call ID. The appliance then parses the GRE data packets and translates call IDs on the basis of the two LSN session entries.

For PPTP protocol, the NetScaler also includes timeout setting for any idle PPTP LSN sessions. If a PPTP LSN session is idle for a time that exceeds the timeout setting, the NetScaler appliance removes the session.

Limitations

The following are the limitations of PPTP ALG on a NetScaler appliance:

- PPTP ALG is not supported for hairpin LSN flow.
- PPTP ALG is not supported to work with any RNAT configuration.
- PPTP ALG is not supported in NetScaler clusters.

Configuring PPTP ALG

Configuring PPTP ALG on the NetScaler appliance consist of the following tasks:

- Create an LSN configuration and enable PPTP ALG on it. In an LSN configuration, the LSN group includes the PPTP ALG setting. For instructions on creating an LSN configuration, see Configuration Steps for LSN.
- (Optional) Set the global timeout for idle PPTP LSN sessions.

To enable PPTP ALG for an LSN configuration by using the NetScaler command line

At the command prompt, type:

- **add lsn group** <groupname> -**clientname** <string> [-**pptp** (ENABLED | DISABLED)]
- **show lsn group**

To set the global timeout for idle PPTP LSN sessions by using the NetScaler command line

At the command prompt, type:

- **set appAlgParam** -**pptpGreIdleTimeout** <positive_integer>
- **show appAlgParam**

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

In the following sample LSN configuration, PPTP ALG is enabled for subscribers in the 192.0.2.0/24 network.

Also idle PPTP LSN session timeout is set to 200 secs.

```
>set lsn parameter -memLimit 4000

Done

>add lsn client LSN-CLIENT-1

Done

>bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

>add lsn pool LSN-POOL-1

Done

>bind lsn pool LSN-POOL-1 203.0.113.3

Done

>add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -pptp ENABLED

Done

>bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done

>set appAlgParam -pptpGreIdleTimeout 200

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Layer Gateway for SIP Protocol

Aug 17, 2015

Using Large Scale NAT (LSN) with Session Initiation Protocol (SIP) is complicated, because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When LSN is used with SIP, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media.

SIP ALG adheres to the following RFCs:

- RFC 3261
- RFC 3581
- RFC 4566
- RFC 4475

How SIP ALG Works

How IP address translation is performed depends on the type and direction of the message. A message can be any of the following:

- Inbound request
- Outbound response
- Outbound request
- Inbound response

For an outgoing message, the private IP address and port number of the SIP client are replaced with the NetScaler-owned public IP address and port number, called the *LSN pool IP address and port number*, specified during LSN configuration. For an incoming message, the LSN pool IP address and the port number are replaced with the private address of the client. If the message contains any public IP addresses, the NetScaler SIP ALG retains them. Also, a pinhole is created on the:

- LSN pool IP address and port on behalf of the private client, so that the messages that arrive at this IP address and port from the public network are treated as SIP messages.
- Public IP address and port on behalf of the public clients, so that the messages that arrive at this IP address and port from the private network are treated as SIP messages.

When a SIP message is sent out across the network, the SIP Application Layer Gateway (ALG) collects information from the message and translates the IP addresses in the following headers into LSN pool IP addresses:

- Via
- Contact
- Route
- Record-Route

In the following sample SIP request message, LSN replaces the IP addresses in the header fields to hide them from the outside network.

INVITE adam@10.102.185.156 SIP/2.0 Via: SIP/2.0/UDP 192.170.1.161:62914 From: eve@10.120.210.3 To: adam@10.102.185.156 Call-ID: a12abcde@10.120.210.3 Contact: adam@10.102.185.156 I

When a message containing SDP information arrives, the SIP ALG collects information from the message and translates the IP addresses in the following fields into LSN pool IP addresses and port numbers:

- c= (connection information)

  This field can appear at the session or media level. It appears in the following format:

  c=<network-type><address-type><connection-address>

  If the destination IP address is a unicast IP address, the SIP ALG creates pinholes by using the IP address and port numbers specified in the m= field.

- m= (media announcement)

  This field appears at the media level and contains the description of the media. It appears in the following format:

  m=<media><port><transport><fmt list>

- a= (information about the media field)

  This field can appear at the session or media level, in the following format:

  a=<attribute>

  a=<attribute>:<value>

The following excerpt from a sample SDP section shows the fields that are translated for resource allocation.

o=user 2344234 55234434 IN IP4 10.150.20.3

c=IN IP4 10.150.20.3

m=audio 43249 RTP/AVP 0

The following table shows how SIP payload is translated.

Inbound Request (from public to private)	To:	None
	From:	None
	Call-ID:	None
	Via:	None

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	Request-URI:	Replace LSN pool IP address with private IP address
	Contact:	None
	Record-Route	None
	Route:	None
Outbound Response (from private to public)	To:	None
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	Replace private IP address with LSN pool IP address
	Contact:	Replace private IP address with LSN pool IP address
	Record-Route	None
	Route:	None
Outbound Request (from private to public)	To:	None
	From:	None
	Call-ID:	None
	Via:	Replace private IP address with LSN pool IP address
	Request-URI:	None
	Contact:	Replace private IP address with LSN pool IP address
	Record-Route	None
	Route:	None
Inbound Response (from public to private)	To:	None
	From:	None
	Call-ID:	None
	Via:	Replace LSN pool IP address with private IP address
	Request-URI:	None
	Contact:	Retain public IP address, if present
	Record-Route	None
	Route:	None

Limitations of SIP ALG

A SIP ALG has the following limitations:

- Only SDP payload is supported.
- The following are not supported:
  - Multicast IP addresses

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Encrypted SDP
- SIP TLS
- FQDN translation
- SIP layer authentication
- TD/partitioning/cluster
- Multipart body
- SIP messages over IPv6 network
- Line folding

Tested SIP Clients and Proxy Servers

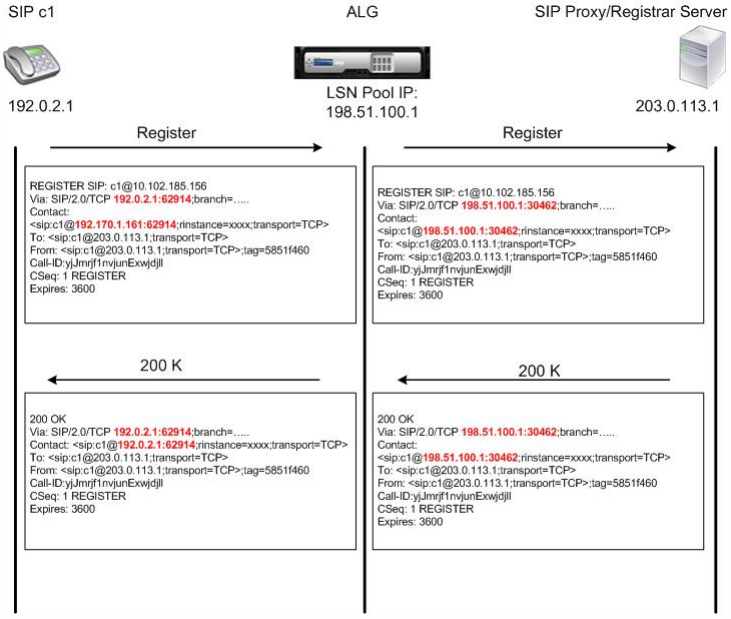
The following SIP clients and proxy server have been tested with SIP ALG:

- **SIP Clients:** X-Lite, Zoiper, Ekiga. Avaya
- **Proxy Server:** openSIPS

LSN SIP Scenario: SIP Proxy Outside the Private Network (Public Network)

SIP Client Registration

For a typical SIP call, SIP client must register with the SIP registrar by composing a REGISTER request and sending it to the SIP registrar. The NetScaler appliance's SIP ALG intercepts the request, replaces the IP address and port number in the request with the LSN pool IP address and port number provided in the LSN configuration, and forwards the request to the SIP registrar. The SIP ALG then opens a pinhole in the NetScaler configuration to allow further SIP communication between the SIP client and the SIP registrar. The SIP registrar sends a 200 OK response to the SIP client over the LSN pool IP address and port number. The NetScaler appliance captures this response in the pinhole, and the SIP ALG replaces the SIP header, putting the original Contact, Via, Route, and Record-Route SIP fields back into the message. The SIP ALG then forwards the message to the SIP client. The following figure shows how SIP ALG uses LSN in a SIP call registration flow.

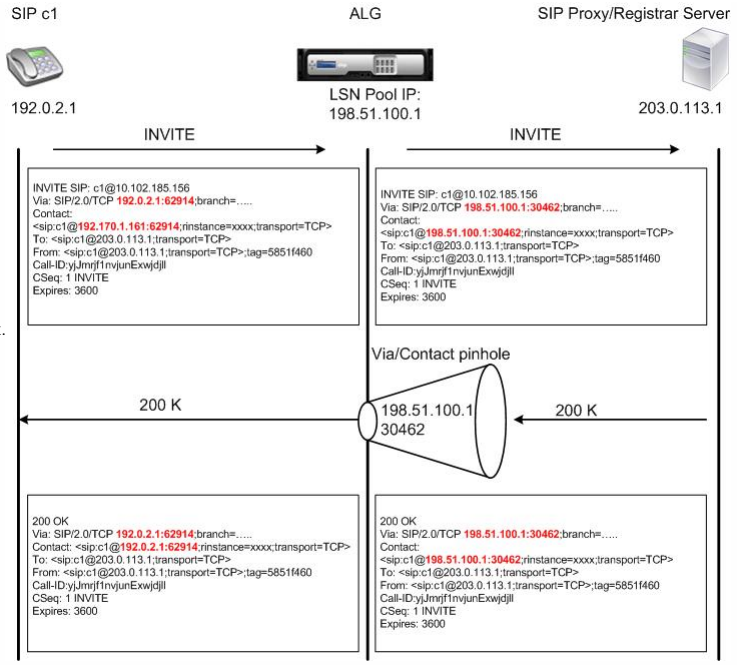


Outgoing Calls

A SIP call is initiated with a SIP INVITE message sent from the internal to the external network. The SIP ALG performs NAT on the IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, replacing them with the LSN pool IP address and port number. LSN stores these mappings for subsequent SIP messages in the SIP call. The SIP ALG then opens separate pinholes in the NetScaler configuration to allow SIP and media through the NetScaler appliance on the dynamically assigned ports specified in the SDP and SIP headers. When a 200 OK message arrives at the NetScaler, it is captured by one of the created pinholes. The SIP ALG replaces the SIP header, restoring the original Contact, Via, Route, and Record-Route SIP fields, and then

© 1999-2017 Citrix Systems, Inc. All rights reserved.

forwards the message to the internal SIP client.



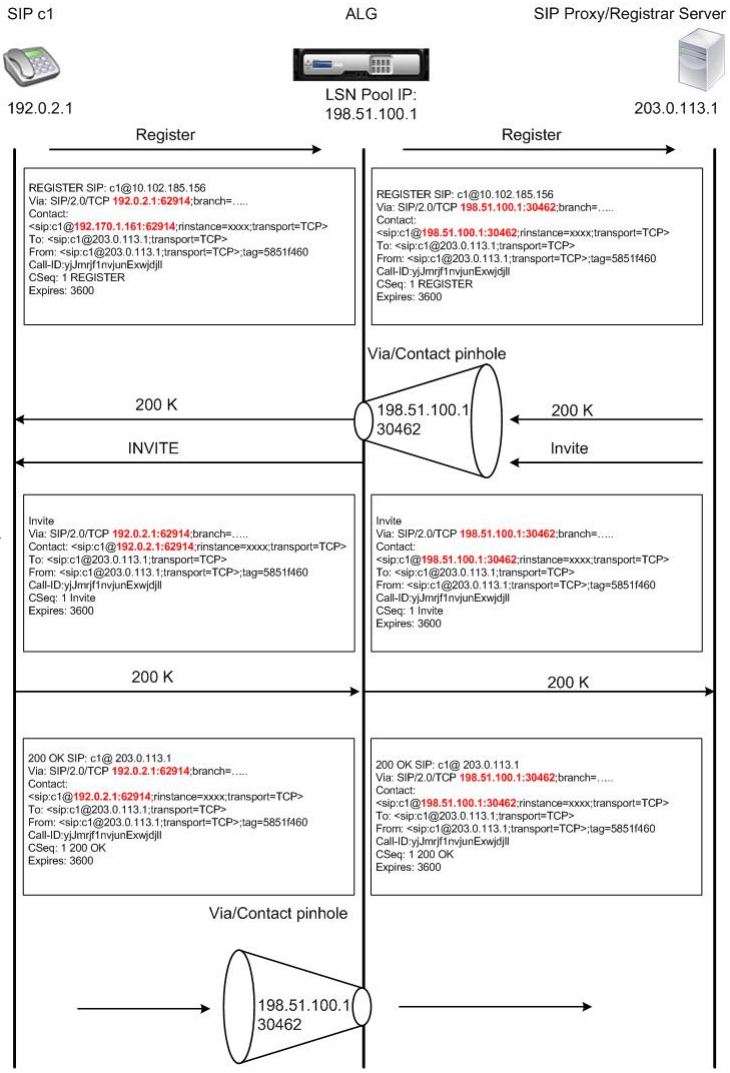
Incoming Calls

A SIP incoming call is initiated with a SIP INVITE message from the external client to the internal network. The SIP registrar forwards the INVITE message to the SIP client in the internal network, using the pinhole that was created when the Internal SIP client registered with the SIP registrar.

The SIP ALG performs NAT on the LSN IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, translating them to the IP address and port number of the internal SIP client, and forwards the request to the SIP client. When the 200 OK response message sent by the internal SIP client arrives at the NetScaler appliance, the SIP ALG performs NAT on the IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, translating them to the LSN pool IP address and port number, forwards the response message to the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SIP registrar, and then opens a pinhole in the outbound direction for further SIP communication.



Call Termination

The BYE message terminates a call. When the device receives a BYE message, it translates the header fields in the message just as it does for any other message. But because a BYE message must be acknowledged by the receiver with a 200 OK, the ALG delays call teardown for 15 seconds to allow time for transmission of the 200 OK.

Call Between Clients in the Same Network

When both client A and client B in the same network initiate a call, the SIP messages are routed through the SIP proxy in the outside network. The SIP ALG processes the INVITE from client A as a normal outgoing call. Since client B is in the same network, the SIP proxy sends the INIVITE back to the NetScaler appliance. The SIP ALG examines the INIVITE message, determines that it contains the NAT IP address of client A, and replaces that address with the private IP address of client A before sending the message to client B. Once the call is established between the clients, the NetScaler is not involved in the media transmission between the clients.

More LSN SIP Scenarios: SIP Proxy Inside the Private Network

If you want to host the SIP Proxy server inside the private network, Citrix recommends that you do one of the following:

- Configure a static LSN Mapping for the private SIP proxy. For more information, see Configuring Static LSN Maps. Make sure that the NAT port is the same as the port configured in the SIP ALG profile.
- Configure the SIP Proxy server inside a demilitarized zone (DMZ).

Figure 1. SIP Call Registration

© 1999-2017 Citrix Systems, Inc. All rights reserved.

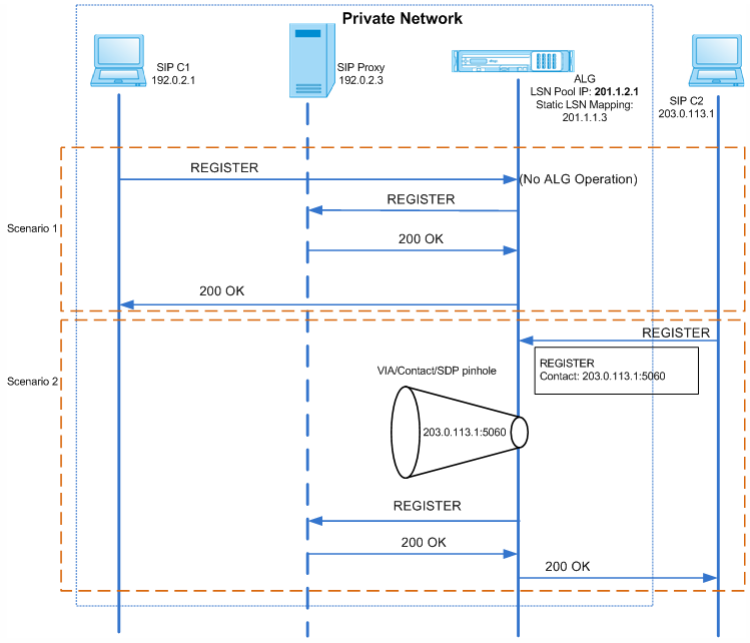
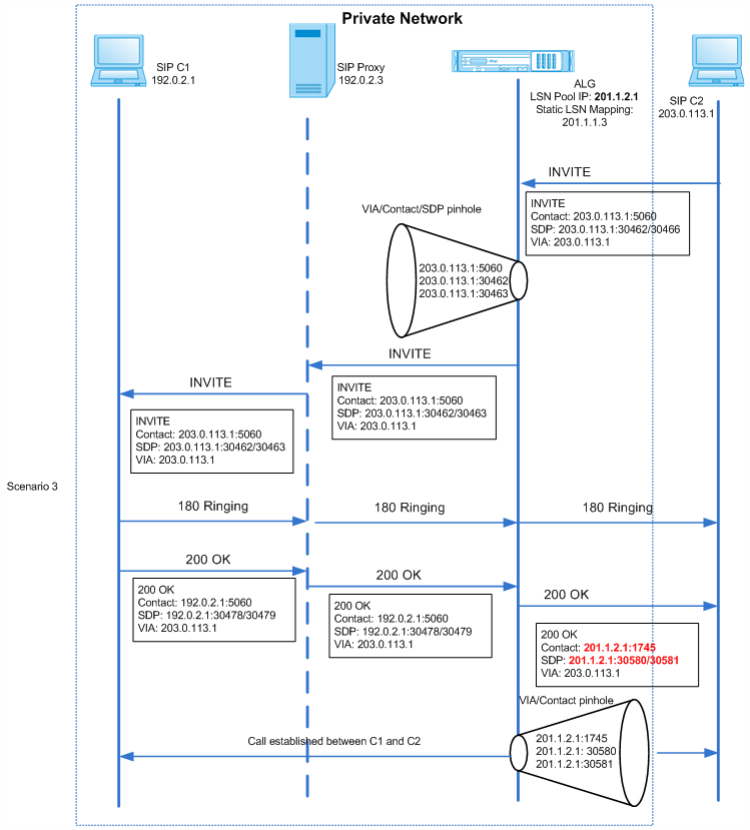


Figure 2. SIP Incoming Call Flow



Figures 1 and 2 show the following scenarios:

- Scenario 1—SIP client in the private network registers with the SIP proxy server in the same network. ALG operations are not performed, because the SIP client and SIP proxy server are in the same network.
- Scenario 2—SIP client in the public network registers with the SIP proxy server in the private network. The REGISTER message from the public SIP client is sent to the NetScaler appliance by using the static LSN mapping configured on the appliance, and the appliance creates a pinhole for further SIP operations.
- Scenario 3— SIP Incoming call flow. A SIP incoming call is initiated with a SIP INVITE message from the external to the internal network. The Netscaler appliance receives the INVITE message from SIP client C2, which is in the external network, through the static LSN maps configured on the NetScaler appliance.
  The appliance creates a pinhole and forwards the INVITE message to the SIP proxy. The SIP proxy then forwards the INVITE message to SIP client C1 in the internal network. SIP client C1 then sends 180 and 200 OK messages to the SIP proxy, which in turn forwards the message to SIP client C2 through the NetScaler appliance.
  When the 200 OK response message sent by internal SIP client C1 arrives at the NetScaler, the SIP ALG performs NAT on the IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, and in the SDP fields, replacing them with the LSN pool IP address and port number. The SIP ALG then forwards the response message to SIP client C2 and opens a pinhole in the outbound direction for further SIP communication.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Support for Audit Logs

You can log ALG information as part of LSN logging by enabling ALG in the LSN audit logging configuration. For more information on LSN logging, see Logging and Monitoring LSN. A log message for an ALG entry in the LSN log consists of the following information:

- Time stamp
- Type of SIP message (for example, SIP request)
- Source IP address and port of the SIP client
- Destination IP address and port of the SIP proxy
- NAT IP address and port
- SIP method
- Sequence number
- Whether or not the SIP client is registered
- Caller's user name and domain
- Receiver's user name and domain

Sample audit log:

Request:

07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG ALG_SIP_INFO_PACKET_EVENT 169 0 : Infomsg: "SIP request" - Group: g2 - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE

Response:

07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG ALG_SIP_INFO_PACKET_EVENT 170 0 : Infomsg: "SIP response" - Group: g2 - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZT

Configuring SIP ALG

You need to configure the SIP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuration Steps for LSN. While configuring LSN, make sure that you:

- Set the following parameters while adding the LSN application profile:
  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT

  Important: For the SIP ALG to work, a full cone NAT configuration is mandatory.

  Example

  add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT

- Create a SIP ALG profile and make sure that you define either the source port range or destination port range.

  Example

  add lsn sipalgprofile sipalgprofile_tcp -sipsrcportrange 1-65535 -sipdstportrange 5060 -openViaPinhole ENABLED -openRecordRoutePinhole ENABLED –sipTransportProtocol TCP

- Set SIP ALG = ENABLED, while creating the LSN group.

  Example

  add lsn group g1 -clientname c1 -sipalg ENABLED

- Bind the SIP ALG profile to the LSN group.

Sample SIP ALG Configuration

The following sample configuration shows how to create a simple LSN configuration with a single subscriber network, single LSN NAT IP address, SIP ALG specific setting, and configure SIP ALG:

```
>add lsn pool p1
Done
>bind lsn pool p1 10.102.185.190
Done
>add lsn client c1
Done
>bind lsn client c1 -network 192.170.1.0 -netmask 255.255.255.0
Done
>add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
>add lsn appsprofile app_udp UDP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
>bind lsn appsprofile app_tcp 1-65535
Done
>bind lsn appsprofile app_udp 1-65535
Done
>add lsn sipalgprofile sipalgprofile_tcp -sipdstportrange 5060 -openViaPinhole ENABLED -openRecordRoutePinhole ENABLED –sipTransportProtocol TCP
Done
>add lsn sipalgprofile sipalgprofile_udp -sipdstportrange 5060 -openViaPinhole ENABLED -openRecordRoutePinhole ENABLED -sipTransportProtocol UDP
Done
>add lsn group g1 -clientname c1 -sipalg ENABLED
Done
>bind lsn group g1 -poolname p1
Done
>bind lsn group g1 -appsprofilename app_tcp
Done
>bind lsn group g1 -appsprofilename app_udp
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>bind lsn group g1 -sipalgprofilename sipalgprofile_tcp
Done
>bind lsn group g1 -sipalgprofilename sipalgprofile_udp
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Layer Gateway for RTSP Protocol

Aug 17, 2015

Real Time Streaming Protocol (RTSP) is an application-level protocol for the transfer of real-time media data. Used for establishing and controlling media sessions between end points, RTSP is a control channel protocol between the media client and the media server. The typical communication is between a client and a streaming media server.

Streaming media from a private network to a public network requires translating IP addresses and port numbers over the network. NetScaler functionality includes an Application Layer Gateway (ALG) for RTSP, which can be used with Large Scale NAT (LSN) to parse the media stream and make any necessary changes to ensure that the protocol continues to work over the network.

How IP address translation is performed depends on the type and direction of the message, and the type of media supported by the client-server deployment. Messages are translated as follows:

- Outbound request—Private IP address to NetScaler-owned public IP address called an LSN pool IP address.
- Inbound response—LSN pool IP address to private IP address.
- Inbound request—No translation.
- Outbound response—Private IP address to LSN pool IP address.

Limitations of RTSP ALG

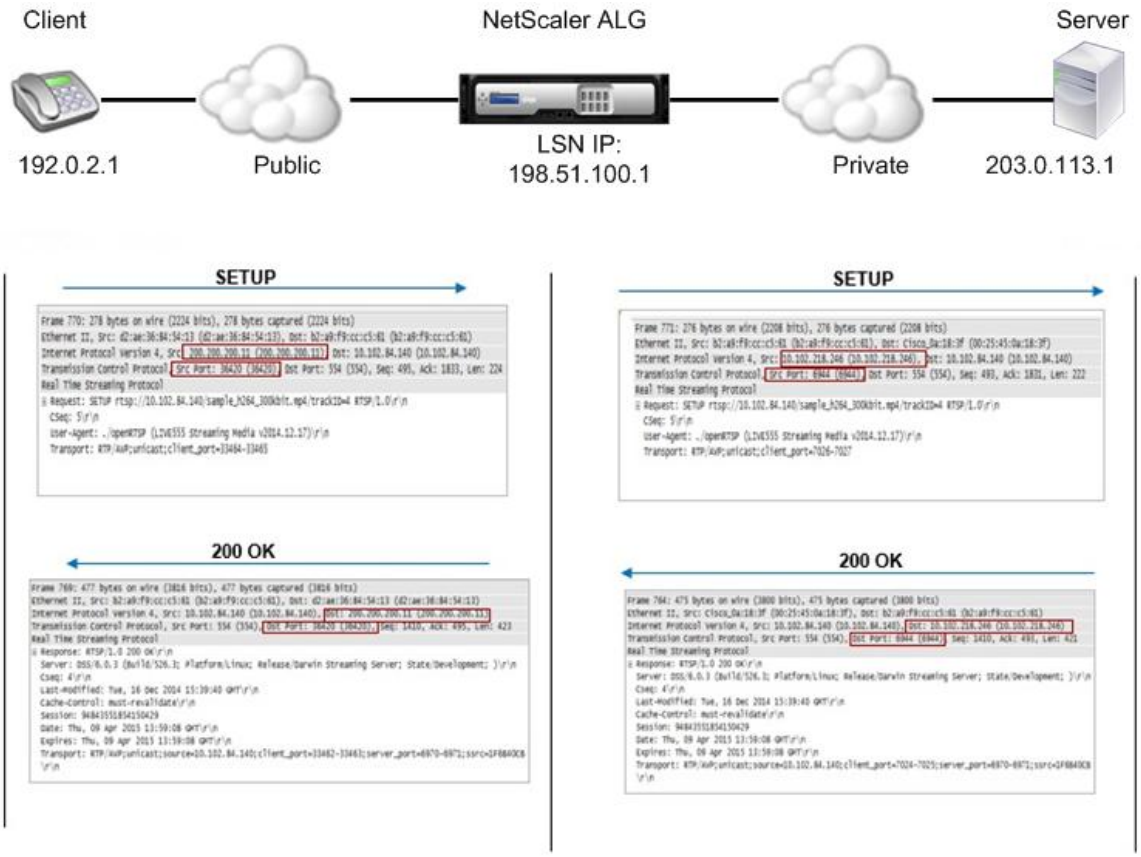
The RTSP ALG does not support the following:

- Multicast RTSP sessions
- RTSP session over UDP
- TD/admin partitioning/cluster deployments
- RSTP Authentication
- HTTP tunneling

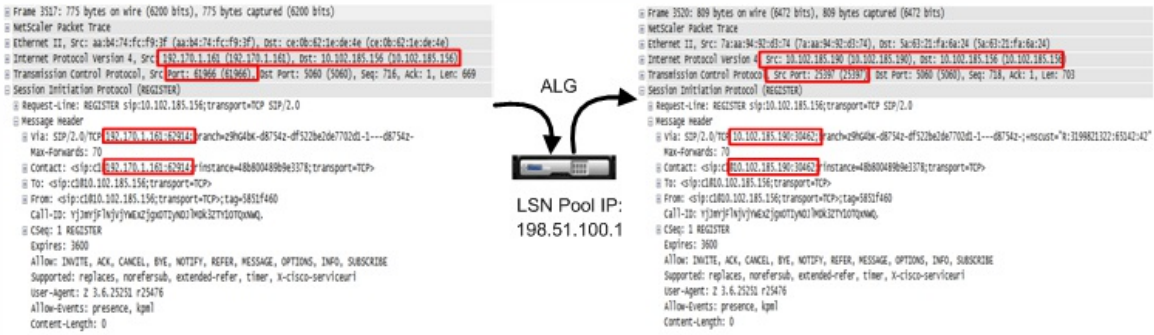
RTSP and LSN scenario

The following figure shows an RTSP SETUP request flow. Typically, a SETUP request specifies how a single media stream must be transported. The request contains the media stream URL and a transport specifier. This specifier typically includes one local port for receiving RTP data (audio or video), and another for receiving RTCP data (meta information). The server reply usually confirms the chosen parameters and fills in the missing parts, such as the server's chosen ports. Each media stream must be configured by using the SETUP command before an aggregate play request can be sent.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

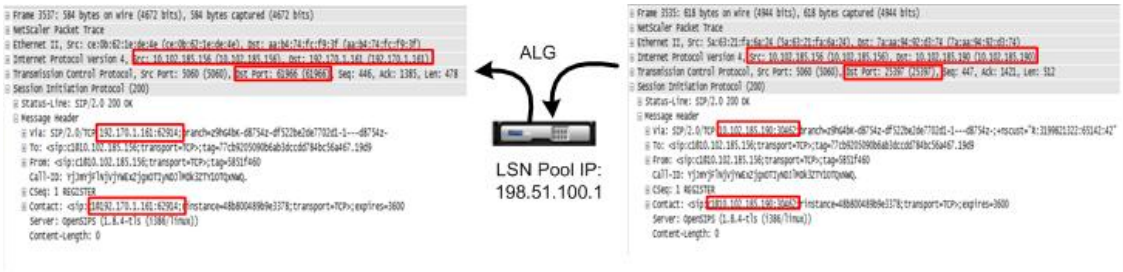


In a typical RTSP communication, the media client in the public network sends a SETUP request to the media server in the private network. RSTP ALG intercepts the request and, in the media stream, replaces the public IP address and port number with the LSN pool IP address and LSN port number. The following figure shows the translation performed by a NetScaler appliance in the media stream for an outbound request:



The media server in the private network uses the LSN pool IP address and LSN port number to send a 200 OK response to the media client in the public network. The NetScaler RTSP ALG intercepts the response and replaces the LSN pool IP address and LSN port number with the public IP address and port number of the media client. The following figure shows the translation performed by a NetScaler appliance in the media stream for an inbound response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Configuring RTSP ALG

Configure RTSP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuration Steps for LSN. While configuring LSN, make sure that you:

- Set the **NAT Type** as DETERMINSTIC or DYNAMIC while adding the LSN pool.
- Set the following parameters while adding the LSN application profile:
  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT
- Create a RTSP ALG profile and bind the RTSP ALG profile to the LSN group

Sample RTSP ALG Configuration

The following sample configuration shows how to create a simple LSN configuration with a single subscriber network, single LSN NAT IP address, and RTSP ALG settings:

```
>enable ns feature WL SP LB CS LSN
Done
>set lsn parameter -memLimit 2100
Done
>add lsn pool pool1 -nattype DETERMINISTIC
Done
>bind lsn pool pool1 10.102.218.246
Done
>add lsn client client1
Done
>bind lsn client client1 -network 200.200.200.11 -netmask 255.255.255.0
Done
>add lsn appsprofile app1 TCP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
>add lsn appsprofile app2 UDP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
>bind lsn appsprofile app1 1-65535
Done
>bind lsn appsprofile app2 1-65535
Done
>add lsn rtspalgprofile rtspalgprofiledefault -rtspIdleTimeout 1000 -rtspportrange 554
Done
>add lsn group group1 -clientname client1 -nattype DETERMINISTIC -portblocksize 512 -rtspalg ENABLED
Done
>bind lsn group group1 -poolname pool1
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>bind lsn group group1 -appsprofilename app1
Done
>bind lsn group group1 -appsprofilename app2
Done
>bind lsn group group1 -rtspalgprofilename rtspalgprofiledefault
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Logging and Monitoring LSN

Jan 04, 2016
You can log LSN information to diagnose, troubleshoot problems, and to meet legal requirements. You can monitor the performance of the LSN feature by using LSN statistical counters and displaying current LSN sessions.

This section includes the following details:

- Logging LSN
- Minimal Logging
- Load Balancing SYSLOG Servers
- Logging HTTP Header Information
- Logging MSISDN Information
- Displaying Current LSN Sessions
- Displaying LSN Statistics

Logging LSN

Updated: 2015-06-29

Logging LSN information is one of the important functions required by the ISPs to meet legal requirements and for identifying the source of traffic at any given time.

A NetScaler appliance logs LSN mapping entries and the LSN sessions created or deleted for each LSN group. You can control logging of LSN information for an LSN group by using the logging and session logging parameters of the LSN group. These are group level parameters and are disabled by default. The NetScaler appliance logs LSN sessions for an LSN group only when both logging and session logging parameters are enabled.

The following table displays the logging behavior for an LSN group for various settings of logging and session logging parameters.

Logging	Session Logging	Logging Behavior
Enabled	Enabled	Logs LSN mapping entries as well as LSN sessions.
Enabled	Disabled	Logs LSN mapping entries but not LSN sessions.
Disabled	Enabled	Logs neither mapping entries nor LSN sessions.

A log message for an LSN mapping entry consists of the following information:
- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced.
- Time stamp
- Entry type (MAPPING)
- Whether the LSN mapping entry was created or deleted
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
  - Destination IP address and port are not logged for Endpoint-Independent mapping.
  - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
  - Destination IP address and port are logged for Address-Port-Dependent mapping.

A log message for an LSN session consists of the following information:
- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced.
- Time stamp
- Entry type (SESSION)
- Whether the LSN session is created or removed
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The appliance uses its existing syslog and audit log framework to log LSN information. You must enable global level LSN logging by enabling the LSN parameter in the related NSLOG action and SYLOG action entities. When the Logging parameter is enabled, the NetScaler appliance generates log messages related to LSN mappings and LSN sessions of this LSN group. The appliance then sends these log messages to servers associated with the NSLOG action and SYSLOG action entities.

For logging LSN information, Citrix recommends:
- Logging the LSN information on external log servers instead of on the NetScaler appliance. Logging on external servers facilitates optimal performance when the appliance creates large amounts of LSN log entries (in order of millions).
- Using SYSLOG over TCP, or NSLOG. By default SYSLOG uses UDP, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data.

  Note:
  - The SYSLOG generated on NetScaler appliance are dynamically sent to the external log servers.
  - When using SYSLOG over TCP, if the TCP connection is down or the SYSLOG server is busy, then the NetScaler appliances stores the logs in buffer and send the data once the connection is active.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

For more information about configuring logging, see Audit Logging.

Configuring LSN logging consists of the following tasks:

- **Configuring the NetScaler appliance for logging.** This task involves creating and setting various entities and parameters of the NetScaler appliance:
  - **Create a SYSLOG or NSLOG audit logging configuration**. Creating an audit logging configuration involves the following tasks:
    - Create a NSLOG or SYSLOG audit action and enable the LSN parameter. Audit actions specify the IP addresses of log servers.
    - Create a SYSLOG or NSLOG audit policy and bind the audit action to the audit policy. Audit actions specify the IP addresses of log servers. Optionally, you can set the transport method for log messages that are sent to the external log servers. By default UDP is selected, you can set the transport method as TCP for a reliable transport mechanism.Bind the audit policy to system global.
    - Create a SYSLOG or NSLOG audit policy and bind the audit action to the audit policy.
    - Bind the audit policy to system global.

    Note: For an existing audit logging configuration, just enable the LSN parameter for logging LSN information in the server specified by the audit action.
  - **Enable logging and session logging parameters.** Enable logging and session logging parameters either as you add LSN groups or after you have created the groups. The NetScaler appliance generates log messages related to these LSN groups and sends them to the server of those audit actions that have the LSN parameter enabled.
- **Configuring log servers.** This task involves installing SYSLOG or NSLOG packages on the desired servers. This task also involves specifying the NSIP address of the NetScaler appliance in the configuration file of SYSLOG or NSLOG. Specifying the NSIP address enables the server to identify the log information sent by the NetScaler appliance for storing them in a log file.

For more information about configuring logging, see Audit Logging.

SYSLOG Configuration Using the Command Line Interface

To create a SYSLOG server action for LSN logging by using the command line interface

At the command prompt, type:

add audit syslogAction <name> <serverIP> [-**serverPort** <port>] -**logLevel** <logLevel>... [-**transport** (TCP)] [-**lsn** (ENABLED | DISABLED)]
To create a SYSLOG server policy for LSN logging by using the command line interface

At the command prompt, type:

add audit syslogPolicy <name> <rule> <action>
To bind a SYSLOG server policy to system global for LSN logging by using the command line interface

At the command prompt, type:

bind system global [<policyName> [-**priority** <positive_integer>]]

SYSLOG Configuration Using the Configuration Utility

To configure a SYSLOG server action for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Syslog** and, on the Servers tab, add a new auditing server or edit an existing server.
2. To enable LSN logging, select the **Large Scale NAT Logging** option.
3. (Optional) To enable SYSLOG over TCP, select the **TCP Logging** option.

To configure a SYSLOG server policy for LSN logging by using the configuration utility

Navigate to **Systems** > **Auditing** > **Syslog** and, on the **Policies** tab, add a new policy or edit an existing policy.
To bind a SYSLOG server policy to system global for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Syslog**.
2. On the **Policies** tab, in the **Action** list, click **Global Bindings** to bind the audit global policies.

NSLOG Configuration Using the Command Line Interface

To create a NSLOG server action for LSN logging by using the command line interface

At the command prompt, type:

add audit nslogAction <name> <serverIP> [-**serverPort** <port>] -**logLevel** <logLevel> ... [-**lsn** (ENABLED | DISABLED)]
To create a NSLOG server policy for LSN logging by using the command line interface

At the command prompt, type:

add audit nslogPolicy <name> <rule> <action>
To bind a NSLOG server policy to system global for LSN logging by using the command line interface

At the command prompt, type:

bind system global [<policyName>]

NSLOG Configuration Using the Configuration Utility

To configure a NSLOG server action for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Nslog** and, on the **Servers** tab, add a new auditing server or, edit an existing server.
2. To enable LSN logging, select the **Large Scale NAT Logging** option.

To configure a NSLOG server policy for LSN logging by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Navigate to **Systems** > **Auditing** > **Nslog** and, on the **Policies** tab, add a new policy or edit an existing policy.
To bind a NSLOG server policy to system global for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Nslog**.
2. On the **Policies** tab, in the **Action** list, click **Global Bindings** to bind the audit global policies.

Example

The following configuration specifies two SYSLOG and two NSLOG servers for storing log entries including LSN logs. LSN Logging is configured for LSN groups LSN-GROUP-2 and LSN-GROUP-3.

The NetScaler appliance generates log messages related to LSN mappings and LSN sessions of these LSN groups, and sends them to the specified log servers.

```
>add audit syslogAction SYS-ACTION-1 198.51.101.10 -logLevel ALL -lsn ENABLED
Done
>add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done
>bind system global SYSLOG-POLICY-1
Done

>add audit syslogAction SYS-ACTION-2 198.51.101.20 -logLevel ALL -lsn ENABLED
Done
>add audit syslogPolicy SYSLOG-POLICY-2 ns_true SYS-ACTION-2
Done
>bind system global SYSLOG-POLICY-2
Done

>add audit nslogAction NSLOG-ACTION-1 198.51.101.30 -logLevel ALL -lsn ENABLED
Done
>add audit nslogPolicy NSLOG-POLICY-1 ns_true NSLOG-ACTION-1
Done
>bind system global NSLOG-POLICY-1
Done

>add audit nslogAction NSLOG-ACTION-2 198.51.101.40 -logLevel ALL -lsn ENABLED
Done
>add audit nslogPolicy NSLOG-POLICY-2 ns_true NSLOG-ACTION-2
Done
>bind system global NSLOG-POLICY-2
Done

> add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-2 –logging ENABLED –sessionLogging ENABLED
Done
> set lsn group LSN-GROUP-2  –logging ENABLED –sessionLogging ENABLED
Done
```

The following configuration specifies SYSLOG configuration for sending log messages to
the external SYSLOG server 192.0.2.10 using TCP.

```
> add audit syslogAction SYS-ACTION-1 192.0.2.10 -logLevel ALL -transport TCP
Done

> add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done

>bind system global SYSLOG-POLICY-1
Done
```

The following table displays sample LSN log entries of each type stored on the configured log servers. These LSN log entries are generated by a NetScaler appliance whose NSIP address is 10.102.37.115.

LSN Log Entry Type	Sample Log Entry
LSN session creation	Local4.Informational 10.102.37.115 08/05/2014:09:59:48 GMT 0-PPE-0 : LSN LSN_SESSION 2581750 : SESSION CREATED Client IP:Port:TD 192.0.2.10: 15136:0, NatIP:NatPort 203.0.113.6: 6234, Destination IP:Port:TD 198.51.100.9: 80:0, Protocol: TCP
LSN session deletion	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_SESSION 3871790 : SESSION DELETED Client IP:Port:TD 192.0.2.11: 15130:0, NatIP:NatPort 203.0.113.6: 7887, Destination IP:Port:TD 198.51.101.2:80:0, Protocol: TCP
LSN mapping creation	Local4.Informational 10.102.37.115 08/05/2014:09:59:47 GMT 0-PPE-0 : LSN LSN_MAPPING 2581580 : EIM CREATED Client IP:Port 192.0.2.15: 14567, NatIP:NatPort 203.0.113.5: 8214, Protocol: TCP
LSN mapping deletion	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_MAPPING 3871700 : EIM DELETED Client IP:Port 192.0.3.15: 14565, NatIP:NatPort 203.0.113.11: 8217, Protocol: TCP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Minimal Logging

Deterministic LSN configurations and Dynamic LSN configurations with port block significantly reduces the LSN log volume. For these two types of configuration, the NetScaler appliance allocates a NAT IP address and a block of ports to a subscriber. The NetScaler appliance generates a log message for a port block at the time of allocation to a subscriber. The NetScaler appliance also generates a log message when a NAT IP address and port block is freed. For a connection, a subscriber can be identified just by its mapped NAT IP address and port block. Because of this reason, the NetScaler appliance does not log any LSN session created or deleted. The appliance also neither logs any mapping entry created for a session nor when the mapping entry gets removed.

The minimal logging feature for deterministic LSN configurations and dynamic LSN configurations with port block is enabled by default and there is no provision to disable it. In other words, the NetScaler appliance automatically do minimal logging for deterministic LSN configurations and dynamic LSN configurations with port block. There is no option available for disabling this feature.

The NetScaler sends the log messages to all the configured log servers. For more information on configuring Logging for LSN on the NetScaler appliance, see Logging LSN.

A log message for each port block consists of the following information:

- NSIP address of the NetScaler appliance
- Time stamp
- Entry type as DETERMINISTIC or PORTBLOCK
- Whether a port block is allocated or is freed
- Subscriber's IP address and the assigned NAT IP address and port block
- Protocol name

Minimal Logging for Deterministic LSN Configuration

Consider an example of a simple deterministic LSN configuration for four subscribers having the IP address 192.0.17.1, 192.0.17.2, 192.0.17.3, and 192.0.17.4.

In this LSN configuration, the port block size is set to 32768 and LSN NAT IP address pool has IP addresses in the range 203.0.113.19-203.0.113.23.

```
>add lsn client LSN-CLIENT-7
Done
>bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask 255.255.255.253
Done
>add lsn pool LSN-POOL-7 -nattype DETERMINISTIC
Done
>bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23
Done
>add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype DETERMINISTIC -portblocksize 32768
Done
>bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7
Done
```

The NetScaler appliance sequentially preallocates, from the LSN NAT IP pool and on the basis of the set port block size, an LSN NAT IP address and a block of ports to each subscriber. It assigns the first block of ports (1024-33791) on the beginning NAT IP address (203.0.113.19) to the beginning subscriber IP address (192.0.17.1). The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. At that point, the first port block on the next NAT IP address is assigned to the subscriber, and so on. The NetScaler logs the NAT IP address and the block of ports allocated for each subscriber.

The NetScaler appliance does not log any LSN session created or deleted for these subscribers. The NetScaler generates the following log messages for the LSN configuration.

```
1) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN LSN_DETERMINISTIC 79201453 0 :  Dtrstc ALLOC Client 12.0.0.241, NatInfo 50.0.0.2:59904 to 60415
2) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN LSN_DETERMINISTIC 79201454 0 :  Dtrstc ALLOC Client 12.0.0.242, NatInfo 50.0.0.2:60416 to 60927
3) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243, NatInfo 50.0.0.2:60928 to 61439
4) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243, NatInfo 50.0.0.2:60928 to 61439
```

When you remove the LSN configuration, the allocated NAT IP address and block of ports is freed from each subscriber. The NetScaler logs NAT IP address and block of ports freed from each subscriber. The NetScaler generates the following log messages for each subscriber when you remove the LSN configuration.

```
1) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN LSN_DETERMINISTIC 79201706 0 :  Dtrstc FREE Client 12.0.0.238, NatInfo 50.0.0.2:58368 to 58879
2) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN LSN_DETERMINISTIC 79201707 0 :  Dtrstc FREE Client 12.0.0.239, NatInfo 50.0.0.2:58880 to 59391
3) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN LSN_DETERMINISTIC 79201708 0 :  Dtrstc FREE Client 12.0.0.240, NatInfo 50.0.0.2:59392 to 59903
```

Minimal Logging for Dynamic LSN Configuration with Port Block

Consider an example of a simple dynamic LSN configuration with port block for any subscriber in the network 192.0.2.0/24. In this LSN configuration, the port block size is set to 1024 and LSN NAT IP address pool has IP addresses in the range 203.0.113.3-203.0.113.4.

```
> set lsn parameter -memLimit 4000
Done
>add lsn client LSN-CLIENT-1
Done
>bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done
>add lsn pool LSN-POOL-1
Done
>bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
Done
>add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Done
>bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done
The NetScaler appliance allocates a random NAT IP address and a block of ports, from the LSN NAT IP pool and on the basis of the set port block size, for a subscriber when it initiates a session for the first time. The NetScaler logs the NAT IP address and block of ports allocated to this subscriber. The NetScaler does not log any LSN session created or deleted for this subscriber. If all the ports are allocated (for different subscriber's sessions) from the subscriber's allocated port block, the NetScaler allocates a new random NAT IP address and port block for the subscriber for additional sessions. The NetScaler logs every NAT IP address and port block allocated to a subscriber.

The NetScaler generates the following log message when the subscriber, having the IP address 192.0.2.1, initiates a session. The log message shows that the NetScaler has allocated NAT IP address 203.0.113.3 and port block 1024-2047 to the subscriber.

03/23/2015:00:07:12 GMT Informational 0-PPE-3 : default LSN LSN_PORTBLOCK 106725793 0 : Portblock ALLOC Client 12.0.2.72, NatInfo 203.0.113.3:1024 to 2047, Proto:TCP
Once there are no more sessions left that is using the allocated NAT IP address and one of the ports in the allocated port block, the allocated NAT IP address and block of ports is freed from the subscriber. The NetScaler logs that the NAT IP address and the block of ports is freed from the subscriber. The NetScaler generates the following log messages for the subscriber, having the IP address 192.0.2.1 , when no more sessions are left that is using the allocated NAT IP address (203.0.113.3) and a port from the allocated port block (1024-2047). The log message shows that the NAT IP address and port block are freed from the subscriber.

03/23/2015:00:11:09 GMT Informational 0-PPE-3 : default LSN LSN_PORTBLOCK 106814342 0 : Portblock FREE Client 12.0.3.122, NatInfo 203.0.113.3: 1024 to 2047, Proto:TC

Load Balancing SYSLOG Servers

The NetScaler appliance send its SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the NetScaler appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastConnection, LeastPackets, and AuditlogHash.

Load balancing of SYSLOG servers using the command line interface

At the command prompt, type:

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.
   add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP | SYSLOGUDP)> <port>

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.
   add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod <AUDITLOGHASH>]

3. Bing the service to the load balancing virtual server.
   Bind lb vserver <name> <serviceName>

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.
   add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel <logLevel>]

5. Add a SYSLOG policy by specifying the rule and action.
   add syslogpolicy <name> <rule> <action>

6. Bind the SYSLOG policy to the system global for the policy to take effect.
   bind system global <policyName>

Load balancing of SYSLOG servers using the configuration utility

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.
   Navigate to Traffic Management > Services, click Add and select SYLOGTCP or SYSLOGUDP as protocol.

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.
   Navigate to Traffic Management > Virtual Servers, click Add and select SYLOGTCP or SYSLOGUDPas protocol.

3. Bing the service to the load balancing virtual server to the service.
   Bing the service to the load balancing virtual server.

   Navigate to Traffic Management > Virtual Servers, select a virtual server and then selectAUDITLOGHASH in the Load Balancing Method.

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.
   Navigate to System > Auditing, click Servers and add a server by selecting LB Vserver option inServers.

5. Add a SYSLOG policy by specifying the rule and action.
   Navigate to System > Syslog, click Policies and add a SYSLOG policy.

6. Bind the SYSLOG policy to the system global for the policy to take effect.
   Navigate to System > Syslog, select a SYSLOG policy and click Action, and then click Global Bindings and bind the policy to system global.

Example

The following configuration specifies load balance of SYSLOG messages among the external log servers using the AUDITLOGHASH as load balancing method. The NetScaler appliance generates SYSLOG events and messages that are load balanced amongst the services, service1, service2, and service 3.

>add service service1 192.0.2.10 SYSLOGUDP 514

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done

>add service service2 192.0.2.11 SYSLOGUDP 514
Done

>add service service3 192.0.2.11 SYSLOGUDP 514
Done

>add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
Done

>bind lb vserver lbvserver1 service1
Done

>bind lb vserver lbvserver1 service2
Done

>bind lb vserver lbvserver1 service3
Done

>add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
Done

>add syslogpolicy syspol1 ns_true sysaction1
Done

>bind system global syspol1
Done
```

Logging HTTP Header Information

Updated: 2015-05-12

The NetScaler appliance can now log request header information of an HTTP connection that is using the LSN fucntionality of the NetScaler. The following header information of an HTTP request packet can be logged:

- URL that the HTTP request is destined to.
- HTTP Method specified in the HTTP request.
- HTTP version used in the HTTP request.
- IP address of the subscriber that sent the HTTP request.

The HTTP header logs can be used by ISPs to see the trends related to the HTTP protocol among a set a subscribers. For example, an ISP can use this feature to find out the most popular websites among a set of subscribers.

An HTTP header log profile is a collection of HTTP header attributes (for example, URL and HTTP method) that can be enabled or disabled for logging. The HTTP header log profile is then bound to an LSN group. The NetScaler appliance then logs HTTP header attributes, which are enabled in the bound HTTP header log profile for logging, of any HTTP requests related to the LSN group. The NetScaler then sends the log messages to the configured log servers.

An HTTP header log profile can be bound to multiple LSN groups but an LSN group can have only one HTTP header log profile.

To create an HTTP header log profile by using the the command line interface

At the command prompt, type:

- add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL (ENABLED | DISABLED)] [-logMethod (ENABLED | DISABLED)] [-logVersion (ENABLED | DISABLED)] [-logHost (ENABLED | DISABLED)]
- show lsn httphdrlogprofile

To bind an HTTP header log profile to an LSN group by using the the command line interface

At the command prompt, type:

- bind lsn group <groupname> -httphdrlogprofilename <string>
- show lsn group <groupname>

Example

In the following example of an LSN configuration, HTTP header log profile HTTP-Header-LOG-1 is bound to LSN group LSN-GROUP-1. The log profile has all the HTTP attributes (URL, HTTP method, HTTP version, and HOST IP address) enabled for logging so that all these attributes are logged for any HTTP requests from subscribers (in the network 192.0.2.0/24) related to the LSN group.

```
> add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done

> set lsn parameter -memLimit 4000
Done
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>add lsn client LSN-CLIENT-1
Done

>bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done

>add lsn pool LSN-POOL-1
Done

>bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
Done

>add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
Done

>bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done

> bind lsn group  LSN-GROUP-1  -httphdrlogprofilename HTTP-HEADER-LOG-1
Done
```

The NetScaler generates the following HTTP header log message when one of the subscriber belonging to the LSN configuration example sends an HTTP request.

The log message tells us that a client having the IP address 192.0.2.33 sends an HTTP request to URL example.com using HTTP method GET and HTTP version 1.1.

03/19/2015:16:24:04 GMT Informational 0-PPE-1 : default LSN Message 59 0 : "LSN Client IP:TD 10.102.37.118:0 URL: example.com Host: 192.0.2.33 Version: HTTP1.1 Method: GET"

Logging MSISDN Information

A Mobile Station Integrated Subscriber Directory Number (MSISDN) is a telephone number uniquely identifying a subscriber across multiple mobile networks. The MSISDN is associated with a country code and a national destination code identifying the subscriber's operator.

You can configure a NetScaler appliance to include MSISDNs in LSN log entries for subscribers in mobile networks. The presence of MSISDNs in the LSN logs helps the administrator in faster and accurate back tracing of a mobile subscriber who has violated a policy or law, or whose information is required by lawful interception agencies.

The following sample LSN log entries include MSISDN information for a connection from a mobile subscriber in an LSN configuration. The log entries show that a mobile subscriber whose MSISDN is E164:5556543210 was connected to destination IP:port 23.0.0.1:80 through the NAT IP:port 203.0.113.3:45195.

Log Entry Type	Sample Log Entry
LSN session creation	Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN mapping creation	Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN session deletion	Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN mapping	Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP

Perform the following tasks for including MSISDN information in LSN logs:

- **Create an LSN log profile**. An LSN log profile includes the log subscriber ID parameter, which specifies whether to or not to include the MSISDN information in the LSN logs of an LSN configuration. Enable the log subscriber ID parameter when creating the LSN log profile.
- **Bind the LSN log profile to an LSN group of an LSN configuration**. Bind the created LSN log profile to an LSN group of an LSN configuration by setting the log profile name parameter to the created LSN log profile name.For instructions on configuring Large Scale NAT, see Configuration Steps for LSN.

To create an LSN log profile by using the NetScaler command line

At the command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **add lsn logprofile** <logprofilename> -**logSubscriberID** (**ENABLED** | **DISABLED**)
- **show lsn logprofile**

To bind an LSN log profile to an LSN group of an LSN configuration by using the NetScaler command line

At the command prompt, type:

- **bind lsn group** <groupname> -logProfileName <lsnlogprofilename>
- **show lsn group**

Sample Configuration COPY

```
In this example of LSN configuration, the LSN log profile has the log subscriber ID parameter enabled. The profile is bound to LSN group LSN-GROUP-9. MSISDN information is included in the LSN sessio

> add lsn logprofile LOG-PROFILE-MSISDN-9 -logSubscriberID ENABLED

Done

> add lsn client LSN-CLIENT-9

Done

> bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0

Done

> add lsn pool LSN-POOL-9

Done

> bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4

Done

> add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

Done

> bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done

> bind lsn group LSN-GROUP-9 -logprofilename LOG-PROFILE-MSISDN-9

Done
```

Displaying Current LSN Sessions

You can display the current LSN sessions for detecting any unwanted or inefficient LSN sessions on the NetScaler appliance. You can display all or some LSN sessions on the basis of selection parameters.

Note: When more than a million LSN sessions exist on the NetScaler appliance, Citrix recommends displaying selected LSN sessions instead of all by using the selection parameters.

Configuration Using the Command Line Interface

To display all LSN sessions by using the command line interface

At the command prompt, type:

show lsn session

To display selective LSN sessions by using the command line interface

At the command prompt, type:

show lsn session [-clientname <string>] [-network <ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <port>]]

Example

To display all LSN sessions existing on a NetScaler ADC

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> show lsn session
        SubscrIP  SubscrPort SubscrTD  DstIP    DstPort DstTD   NatIP NatPort Proto Dir
1. 192.0.2.10  15136  0  198.51.100.9  80  0 203.0.113.6 6234  TCP OUT
2. 192.0.2.11  15130  0  198.51.101.2  80  0 203.0.113.6 7887  TCP OUT
3. 192.0.2.12  16136  0  198.51.100.3  80  0 203.0.113.6 9807  TCP OUT
4. 192.0.2.13  18148  0  198.51.101.6  80  0 203.0.113.6 4657  TCP OUT
5. 192.0.2.14  13560  0  198.51.101.7  80  0 203.0.113.7 9341  TCP OUT
6. 192.0.2.15  14567  0  198.51.100.8  80  0 203.0.113.5 8214  TCP OUT
7. 192.0.2.15  16890  0  198.51.101.1  80  0 203.0.113.5 8214  TCP OUT
8. 192.0.2.16  12345  0  198.51.102.9  80  0 203.0.113.5 1678  TCP OUT
9. 192.0.2.19  19876  0  198.51.103.8  80  0 203.0.113.5 1567  TCP OUT
10. 192.0.2.20  10989  0  198.51.104.19  80  0 203.0.113.11 1343  TCP OUT
11. 192.0.3.13  18149  0  198.51.101.61  80  0 203.0.113.11 4653  TCP OUT
12. 192.0.3.14  13510  0  198.51.101.74  80  0 203.0.113.11 9344  TCP OUT
13. 192.0.3.15  14565  0  198.51.100.82  80  0 203.0.113.11 8217  TCP OUT
14. 192.0.3.15  16899  0  198.51.101.12  80  0 203.0.113.11 8219  TCP OUT
15. 192.0.3.16  12343  0  198.51.102.99  80  0 203.0.113.11 1673  TCP OUT

Done
```

To display all LSN sessions related to an LSN client entity LSN-CLIENT-2

```
> show lsn session -clientname LSN-CLIENT-2
   SubscrIP  SubscrPort SubscrTD  DstIP   DstPort DstTD   NatIP NatPort Proto Dir
1. 192.0.2.10  15136  0  198.51.100.9  80  0 203.0.113.6 6ß234  TCP OUT
2. 192.0.2.11  15130  0  198.51.101.2  80  0 203.0.113.6 7887  TCP OUT
3. 192.0.2.12  16136  0  198.51.100.3  80  0 203.0.113.6 9807  TCP OUT
4. 192.0.2.13  18148  0  198.51.101.6  80  0 203.0.113.6 4657  TCP OUT
5. 192.0.2.14  13560  0  198.51.101.7  80  0 203.0.113.7 9341  TCP OUT
6. 192.0.2.15  14567  0  198.51.100.8  80  0 203.0.113.5 8214  TCP OUT
7. 192.0.2.15  16890  0  198.51.101.1  80  0 203.0.113.5 8214  TCP OUT
8. 192.0.2.16  12345  0  198.51.102.9  80  0 203.0.113.5 1678  TCP OUT
9. 192.0.2.19  19876  0  198.51.103.8  80  0 203.0.113.5 1567  TCP OUT
10. 192.0.2.20  10989  0  198.51.104.19  80  0 203.0.113.11 1343  TCP OUT

Done
```

To display all LSN sessions that uses 203.0.113.5 as the NAT IP address

```
> show lsn session –natIP 203.0.113.5
SubscrIP  SubscrPort SubscrTD DstIP   DstPort DstTD   NatIP NatPort Proto Dir
1. 192.0.2.15  14567  0  198.51.100.8  80  0 203.0.113.5 8214  TCP OUT
2. 192.0.2.15  16890  0  198.51.101.1  80  0 203.0.113.5 8214  TCP OUT
3. 192.0.2.16  12345  0  198.51.102.9  80  0 203.0.113.5 1678  TCP OUT
4. 192.0.2.19  19876  0  198.51.103.8  80  0 203.0.113.5 1567  TCP OUT
Done
```

Configuration Using the Configuration Utility

To display all or selected LSN sessions by using the configuration utility

1. Navigate to System > Large Scale NAT > Sessions, and click the NAT44 tab.
2. For displaying LSN sessions on the basis of selection parameters, click Search.

Parameter Descriptions (of commands listed in the CLI procedure)

show lsn session

clientname

Name of the LSN Client entity. Maximum Length: 127

network

IP address or network address of subscriber(s).

netmask

Subnet mask for the IP address specified by the network parameter.

Default value: 255.255.255.255

td

Traffic domain ID of the LSN client entity.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Default value: 0

Minimum value: 0

Maximum value: 4094

natIP

Mapped NAT IP address used in LSN sessions.

Displaying LSN Statistics

You can display statistics related to the LSN feature for evaluating the performance of the LSN feature or to troubleshoot problems. You can display a summary of statistics of the LSN feature or of a particular LSN group. The statistical counters reflect events since the NetScaler appliance was last restarted. All these counters are reset to 0 when the NetScaler appliance is restarted.

To display all LSN statistics by using the command line interface

At the command prompt, type:

```
stat lsn
```

To display statistics for a specified LSN group by using the command line interface

At the command prompt, type:

```
stat lsn group [<groupname>]
```

Example

```
> stat lsn

Large Scale NAT statistics
                                    Rate(/s)        Total
LSN TCP Received Packets                 0           40
LSN TCP Received Bytes                   0         3026
LSN TCP Transmitted Packets              0           40
LSN TCP Transmitted Bytes                0         3026
LSN TCP Dropped Packets                  0            0
LSN TCP Current Sessions                 0            0
LSN UDP Received Packets                 0            0
LSN UDP Received Bytes                   0            0
LSN UDP Transmitted Packets              0            0
LSN UDP Transmitted Bytes                0            0
LSN UDP Dropped Packets                  0            0
LSN UDP Current Sessions                 0            0
LSN ICMP Received Packets                0          982
LSN ICMP Received Bytes                  0        96236
LSN ICMP Transmitted Packets             0            0
LSN ICMP Transmitted Bytes               0            0
LSN ICMP Dropped Packets                 0          982
LSN ICMP Current Sessions                0            0
LSN Subscribers                          0            1

Done

> stat lsn group LSN-GROUP-1

LSN Group Statistics
                                    Rate (/s)       Total
TCP Translated Pkts                      0           40
TCP Translated Bytes                     0         3026
TCP Dropped Pkts                         0            0
TCP Current Sessions                     0            0
UDP Translated Pkts                      0            0
UDP Translated Bytes                     0            0
UDP Dropped Pkts                         0            0
UDP Current Sessions                     0            0
ICMP Translated Pkts                     0            0
ICMP Translated Bytes                    0            0
ICMP Dropped Pkts                        0            0
ICMP Current Sessions                    0            0
Current Subscribers                      0            1

Done
```

Parameter Descriptions (of commands listed in the CLI procedure)

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

stat lsn group

groupname

Name of the LSN Group. Maximum Length: 127

detail

Specifies detailed output (including more statistics). The output can be quite voluminous. Without this argument, the output will show only a summary.

fullValues

Specifies that numbers and strings should be displayed in their full form. Without this option, long strings are shortened and large numbers are abbreviated.

ntimes

The number of times, in intervals of seven seconds, the statistics should be displayed.

Default value: 1

logFile

The name of the log file to be used as input.

clearstats

Clear the statsistics / counters

Possible values: basic, full

© 1999-2017 Citrix Systems, Inc. All rights reserved.

STUN Timeout

Aug 17, 2015

STUN (Session Traversal Utilities for NAT) enables an end host operating behind a NAT device to discover its NAT IP address and NAT port allocated by the NAT device. Interactive communication applications (for example real-time voice, video, and messaging) running on these hosts use the STUN protocol for discovering NAT IP address and port information. This information is used by these applications to connect to their peer applications in the Internet. STUN protocol includes servers, known as STUN servers, residing in the Internet. Using the STUN protocol, an application of an end host sends a request to a known STUN server, which in turn then embeds the NAT IP address and port in the payload of its response packet.

In an LSN deployment of a NetScaler appliance for an ISP, interactive communication applications (for example real-time voice, video, and messaging) running on a subscribers can use the STUN protocol to discover whether it is behind a NAT (NetScaler appliance) device or not. These applications send a request to a known STUN server. On receiving the request, the NetScaler allocates a NAT IP address and a port for this request, creates an LSN session and an LSN mapping entry, translates the packet with the allocated NAT IP address and port, and then forwards the packet to the STUN server. The STUN server embeds the allocated NAT IP address and port in the payload of its response packet. When the subscriber finally receives the packet, from the payload of the packet it learns that it is behind a NAT device, and the NAT IP address and port allocated for the session.

The application then notifies the peer applications that it is reachable at the NAT IP address and the port of the LSN mapping entry created for the STUN session. It notifies by embedding the NAT IP address and port in the payload of the packets sent to the peer applications. For making the application reachable at the same LSN mapping entry for any external application, full Cone NAT (endpoint Independent mapping and Endpoint Independent filtering) is enabled for the LSN configuration on the NetScaler.

The NetScaler detects an LSN session of type STUN if the request packets are destined to TCP or UDP port 3478, and then marks the created mapping entry of type STUN. The NetScaler applies a timeout called, STUN timeout, to the created STUN LSN mapping entry. A STUN timeout is the maximum time that the NetScaler maintains an idle STUN LSN mapping entry since it was last used by any LSN session. If the STUN LSN mapping session is unused for a time that exceeds the STUN timeout, the NetScaler removes the mapping entry.

For an application on a subscriber that use STUN LSN mapping entry to stay available to other peer applications on the Internet, the application periodically sends keep-alive messages to the NetScaler appliance so that the STUN LSN mapping entry does not timeout. A higher frequency of keep-alive messages can have an affect on the performance a subscriber, especially, if the subscriber is a mobile device. A higher value of STUN timeout reduces the frequency of keep-alive messages from a subscriber.

ALGs on the NetScaler appliance do not apply to an LSN session that use a STUN LSN mapping entry because NAT IP address and NAT port are communicated in payload of the packets related to the session.

For subscribers' applications that use STUN protocol, the LSN configuration must have the following settings:

- STUN timeout. In an LSN configuration, the LSN group includes the STUN timeout setting.
- Endpoint-independent mapping and endpoint-independent filtering for STUN protocol ports.

For instructions on creating an LSN configuration, see Configuration Steps for LSN.

Example

The following sample LSN configuration applies to applications that use STUN protocol over TCP or UDP. STUN timeout is set to 10 mins. Endpoint-independent mapping and endpoint-independent filte

```
>add lsn client LSN-CLIENT-1
Done
>bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done
>add lsn pool LSN-POOL-1
Done
>bind lsn pool LSN-POOL-1 203.0.113.3
Done
>add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -stuntimeout 10
Done
>bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done
>add lsn appsprofile LSNAPPSPROFILE-TCP-STUN-1 TCP -mapping ENDPOINT-INDEPENDENT –filtering  ENDPOINT-INDEPENDENT
Done
>bind lsn appsprofile LSNAPPSPROFILE-TCP-STUN-1 3748
Done
>bind lsn group LSN-GROUP-1 -applicationprofilename LSNAPPSPROFILE-TCP-STUN-1
Done
>add lsn appsprofile LSNAPPSPROFILE-UDP-STUN-1 UDP -mapping ENDPOINT-INDEPENDENT –filtering  ENDPOINT-INDEPENDENT
Done
>bind lsn appsprofile LSNAPPSPROFILE-UDP-STUN-1 3748
Done
>bind lsn group LSN-GROUP-1 -applicationprofilename LSNAPPSPROFILE-UDP-STUN-1
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

TCP SYN Idle Timeout

Aug 17, 2015

SYN idle timeout is the timeout for establishing TCP connections that use LSN on the NetScaler appliance. If a TCP session is not established within the configured timeout period, the NetScaler removes the session. SYN idle timeout is useful in providing protection against SYN flood attacks. In an LSN configuration, the LSN group entity includes the SYN idle timeout setting.

Example

In the following sample LSN configuration, SYN idle timeout is set to 30 secs for TCP connections related to subscribers from the 192.0.2.0/24 network.

```
>set lsn parameter -memLimit 4000
Done
>add lsn client LSN-CLIENT-1
Done
>bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done
>add lsn pool LSN-POOL-1
Done
>bind lsn pool LSN-POOL-1 203.0.113.3
Done
>add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 –synidletimeout 30
Done
>bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Overriding LSN configuration with Load balancing Configuration

Jun 20, 2016

An LSN configuration takes precedence over any load balancing configuration by default. For overriding the large scale networking (LSN) configuration with the load balancing configuration for traffic matching both configurations, create a net profile with Override LSN parameter enabled and bind this profile to the virtual server of the load balancing configuration. USNIP or USIP settings of the load balancing configuration are applied to the traffic, instead of applying the LSN IP address of the LSN configuration.

This option is useful in an LSN deployment that includes NetScaler appliances and value added services, such as firewall and optimization devices. In this type of deployment, the ingress traffic on the NetScaler appliance is required to pass through these value-added services before an LSN configuration on the appliance is applied to the traffic. For the NetScaler appliance to send the ingress traffic to a value added service, a load balancing configuration is created and override LSN is enabled on the appliance. The load balancing configuration includes value added services, represented as load balancing services, bound to a virtual server of type ANY. The virtual server is configured with listen policies for identifying the traffic to be sent to the value added service.

To enable override lsn in a net profile by using the NetScaler command line

- To enable override lsn while adding a net profile, at the command prompt, type:
  - **add netProfile** <name> -**overrideLsn** (**ENABLED** | **DISABLED**)
  - **show netprofile** <name>

- To enable override lsn while adding a net profile, at the command prompt, type:
  - **set netProfile** <name> -**overrideLsn** (**ENABLED** | **DISABLED**)
  - **show netprofile** <name>

To enable override lsn in a net profile by using NetScaler GUI

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Set the **Override LSN** parameter while adding or modifying net profiles.

In the following sample configuration, net profile NETPROFILE-OVERRIDELSN-1 has override LSN option enabled and is bound to load balancing virtual server LBVS-1.

Sample configuration COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> add netprofile NETPROFILE-OVERRIDELSN-1 -overrideLsn ENABLED

Done

> set lb vserver LBVS-1 -netprofile NETPROFILE-OVERRIDELSN-1

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Clearing LSN Sessions

Aug 17, 2015

You can remove any unwanted or inefficient LSN sessions from the NetScaler appliance. The appliance immediately releases resources (such as NAT IP address, port, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected LSN sessions from the NetScaler appliance.

To clear all LSN sessions by using the command line interface

At the command prompt, type:

- flush lsn session
- show lsn session

To clear selective LSN sessions by using the command line interface

At the command prompt, type:

- flush lsn session [-clientname <string>] [-network <ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <port>]]
- show lsn session

Example

Clear all LSN sessions existing on a NetScaler ADC
> flush lsn session
Done

Clear all LSN sessions related to LSN client entity LSN-CLIENT-1
> flush lsn session -clientname LSN-CLIENT-1
Done

Clear all LSN sessions related to a subscriber network (192.0.2.0) of LSN client entity LSN-CLIENT-2 belonging to traffic domain 100
> flush lsn session -clientname LSN-CLIENT-2 –network 192.0.2.0 –netmask 255.255.255.0 –td 100
Done

To clear all LSN sessions by using the configuration utility

Navigate to System > Large Scale NAT > Sessions, and click Flush Sessions.

Parameter Descriptions (of commands listed in the CLI procedure)

flush lsn session

clientname

Name of the LSN Client entity. Maximum Length: 127

network

IP address or network address of subscriber(s).

netmask

Subnet mask for the IP address specified by the network parameter.

Default value: 255.255.255.255

td

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Traffic domain ID of the LSN client entity.

Default value: 0

Minimum value: 0

Maximum value: 4094

natIP

Mapped NAT IP address used in LSN sessions.

natPort

Mapped NAT port used in the LSN sessions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing SYSLOG Servers

Jul 07, 2017

Load Balancing SYSLOG Servers

The NetScaler appliance send its SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the NetScaler appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastConnection, LeastPackets, and AuditlogHash.

Load balancing of SYSLOG servers using the command line interface

At the command prompt, type:

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.
   add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP | SYSLOGUDP)> <port>

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.
   add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod <AUDITLOGHASH>]

3. Bing the service to the load balancing virtual server.
   Bind lb vserver <name> <serviceName>

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.
   add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel <logLevel>]

5. Add a SYSLOG policy by specifying the rule and action.
   add syslogpolicy <name> <rule> <action>

6. Bind the SYSLOG policy to the system global for the policy to take effect.
   bind system global <policyName>

Load balancing of SYSLOG servers using the configuration utility

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.
   Navigate to Traffic Management > Services, click Add and select SYLOGTCP or SYSLOGUDP as protocol.

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.
   Navigate to Traffic Management > Virtual Servers, click Add and select SYLOGTCP or SYSLOGUDPas protocol.

3. Bing the service to the load balancing virtual server to the service.
   Bing the service to the load balancing virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Navigate to Traffic Management > Virtual Servers, select a virtual server and then selectAUDITLOGHASH in the Load Balancing Method.

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.
   Navigate to System > Auditing, click Servers and add a server by selecting LB Vserver option inServers.

5. Add a SYSLOG policy by specifying the rule and action.
   Navigate to System > Syslog, click Policies and add a SYSLOG policy.

6. Bind the SYSLOG policy to the system global for the policy to take effect.
   Navigate to System > Syslog, select a SYSLOG policy and click Action, and then click Global Bindings and bind the policy to system global.

Example

The following configuration specifies load balance of SYSLOG messages among the external log servers using the AUDITLOGHASH as load balancing method. The NetScaler appliance generates SYSLOG events and messages that are load balanced amongst the services, service1, service2, and service 3.

```
add service service1 192.0.2.10 SYSLOGUDP 514
add service service2 192.0.2.11 SYSLOGUDP 514
add service service3 192.0.2.11 SYSLOGUDP 514
add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
bind lb vserver lbvserver1 service1
bind lb vserver lbvserver1 service2
bind lb vserver lbvserver1 service3
add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
add syslogpolicy syspol1 ns_true sysaction1
bind system global syspol1
```

Limitations

- The NetScaler appliance does not support an external load balancing virtual server load balancing the SYSLOG messages among the log servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Dual-Stack Lite

Jan 04, 2016

Because of the shortage of IPv4 addresses, and the advantages of IPv6 over IPv4, many ISPs have started transitioning to IPv6 infrastructure. But during the transition, ISPs must continue to support IPv4 along with IPv6, because most of the public Internet still uses only IPv4, and many subscribers do not support IPv6.

Dual Stack Lite (DS-Lite) is an IPv6 transition solution for ISPs with IPv6 infrastructure to connect their IPv4 subscribers to the Internet. DS-Lite uses IPv4-in-IPv6 tunneling to send a subscriber's IPv4 packet through a tunnel on the IPv6 access network to the ISP. The IPv6 packet is decapsulated to recover the subscriber's IPv4 packet and is then sent to the Internet after NAT address and port translation and other LSN related processing. The response packets traverse through the same path to the subscriber.

The NetScaler appliance implements the AFTR component of a DS-Lite deployment and is compliant with RFC 6333.

This document includes the following details:

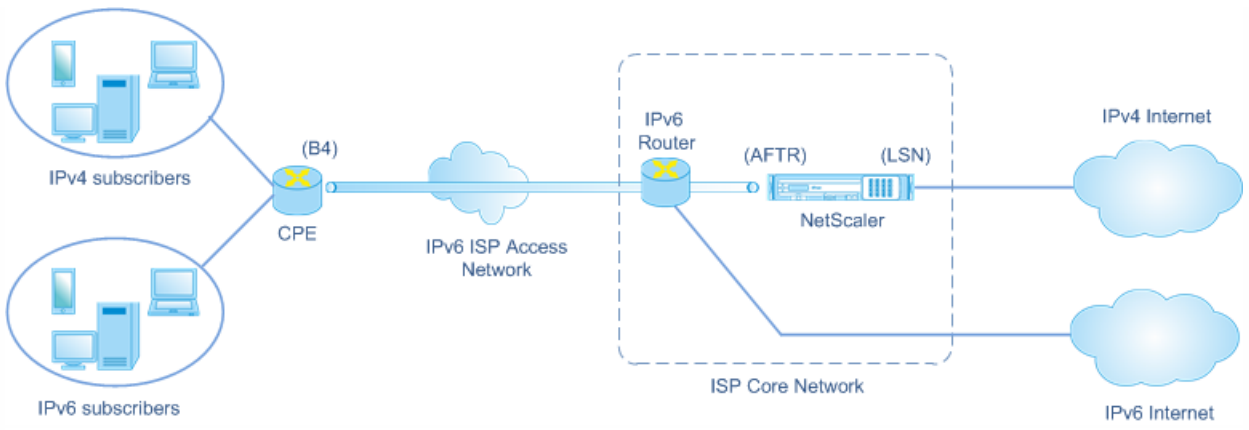
- Architecture
- Example
- Points to Consider before Configuring DS-Lite
- Configuring DS-Lite
- Configuring DS-Lite Static Maps
- Configuring Deterministic NAT Allocation for DS-Lite

Architecture

The Dual-Stack Lite architecture for an ISP consists of the following components:

- **Basic Bridging Broadband (B4).** Basic Bridging broadband, or B4, is a device or component that resides in the subscriber premises. Typically, B4 is a component in the CPE devices in the subscriber premises. IPv4 subscribers are connected to the IPv6-only ISP access network through the CPE device containing the B4 component. The main function of the B4 is to initiate an IPv6 tunnel between B4 and an address family transition router (AFTR) in order to send or receive subscriber IPv4 request or response packets over the tunnel. B4 includes an IPv6 address known as the B4 tunnel endpoint address. B4 uses this address to source IPv6 packets to AFTR and receive packets from AFTR.
- **Address family transition router (AFTR).** AFTR is a device or component residing in the ISP's core network. AFTR terminates the IPv6 tunnel from the B4 device. In other words, the IPv6 tunnel is formed between B4 in the subscriber premise and AFTR in ISP core network. AFTR decapsulates IPv6 packets received from B4 to recover the subscribers' original IPv4 packets. AFTR sends the IPv4 packets to the LSN device or component. LSN routes the IPv4 packets to their destination after performing NAT address and port translation (NAT 44) and other LSN related processing. AFTR includes an IPv6 address known as the AFTR tunnel endpoint address. AFTR uses this address to source IPv6 packets to B4 and receive IPv6 packets from B4. The NetScaler appliance implements the AFTR component.
- **Softwire.** The IPv6 tunnel created between B4 and AFTR is called a softwire.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The DS-Lite architecture of an ISP using a NetScaler appliance consists of subscribers in private address spaces accessing the Internet through a NetScaler appliance deployed in ISP's core network. IPv4 subscribers are connected to a CPE device that includes the DS-Lite B4 functionality. The CPE device is connected to the ISP core network through ISP's IPv6-only access network. The NetScaler appliance contains the DS-Lite AFTR and LSN functionality.

IPv4 subscribers connected to the CPE device are assigned private IPv4 addresses either manually or through DHCP server running on the CPE device. On the CPE device, the AFTR tunnel endpoint address is specified manually or through DHCPv6. Configuration of CPE devices is vendor specific and therefore outside the scope of this documentation.

Upon receiving a request packet that is from an IPv4 subscriber and destined to a location on the Internet, the B4 component of the CPE device encapsulates the IPv4 packet in an IPv6 packet and sends it to the NetScaler appliance in the ISP core network. The NetScaler appliance's AFTR functionality decapsulates the IPv6 packet to recover the subscriber's original IPv4 packet. The LSN functionality of the NetScaler appliance translates the source IP address and port of the IPv4 packet to an NAT IP address and NAT port selected from the configured NAT pool, and then sends the packet to its destination on the Internet.

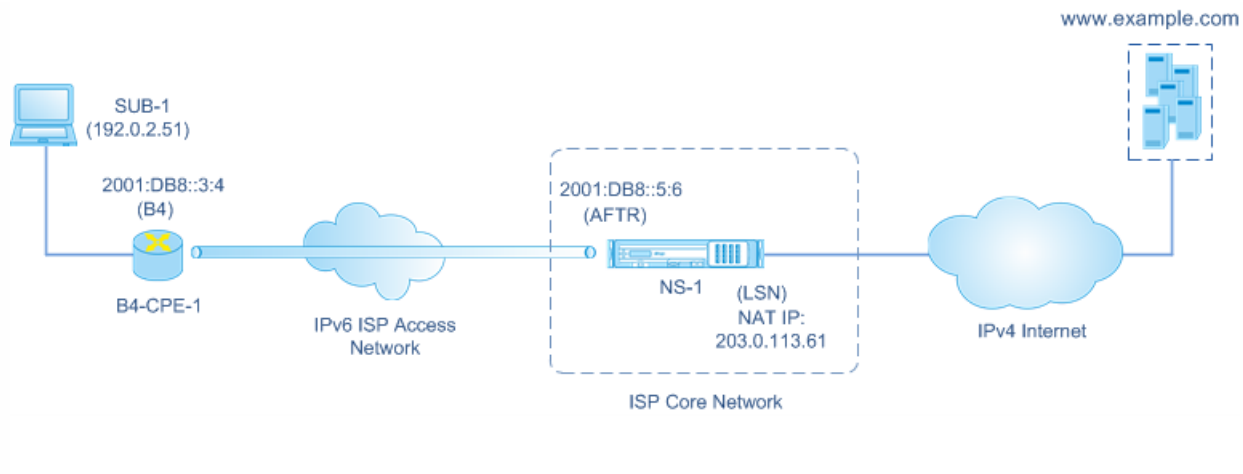
The appliance maintains a record of all active sessions that use the AFTR and LSN functionalities. These sessions are called DS-Lite sessions. The NetScaler appliance also maintains the mappings between B4 IPv6 address, subscriber IPv4 address and port, and NAT IPv4 address and port, for each DS-Lite session. These mappings are called DS-Lite LSN mappings. From DS-Lite session entries and DS-Lite LSN mapping entries, the NetScaler appliance recognizes a response packet (received from the Internet) as belonging to a particular DS-Lite session.

When the NetScaler appliance receives a response packet belonging to a particular DS-Lite session, the appliance's LSN functionality translates the destination IP address and port of the response packet from NAT IP address and port to the subscriber IP address and port, the AFTR functionality encapsulates the resulting packet in an IPv6 packet and sends it to the CPE device. The B4 functionality of the CPE device decapsulates the IPv6 packet to recover the IPv4 response packet, and then sends the IPv4 packet to the subscriber.

Example

Consider an example of a DS-Lite deployment consisting of NetScaler NS-1 in an ISP's core network, CPE device B4-CPE-1 in a subscriber premise, and a single IPv4 subscriber SUB-1. B4-CPE-1 supports the B4 functionality of DS-Lite feature.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The following table lists the settings used in this example.

Entity	Name	Details
IPv4 address of subscriber SUB-1		192.0.2.51
IPv6 address of softwire endpoint on the B4 device (B4-CPE-1)		2001:DB8::3:4
IPv6 address of the softwire endpoint on the AFTR device (NS-1)		2001:DB8::5:6
Settings on NetScaler appliance NS-1		
LSN client	LSN-DSLITE-CLIENT-1	• Network6 (Identifying traffic from B4 devices) = 2001:DB8::3:0/100
LSN pool	LSN-DSLITE-POOL-1	• LSN IPs (NAT IP) = 203.0.113.61 - 203.0.113.70
IPv6 Profile	LSN-DSLITE-PROFILE-1	• Type = DS-LITE • IPv6 address (AFTR IPv6 address) = One of the NetScaler owned IPv6 address of type SNIP6 = 2001:DB8::5:6

© 1999-2017 Citrix Systems, Inc. All rights reserved.

LSN group	LSN-DSLITE-GROUP-1	• LSN client = LSN-DSLITE-CLIENT-1 • LSN pool = LSN-DSLITE-POOL-1 • IPv6 profile = LSN-DSLITE-PROFILE-1

Following is the traffic flow in this example:

1. IPv4 subscriber SUB-1 sends a request to www.example.com. The IPv4 packet has:

- Source IP address = 192.0.2.51
- Source port = 2552
- Destination IP address = 198.51.100.250
- Destination port = 80

2. Upon receiving the IPv4 request packet, B4-CPE-1 encapsulates it in the payload of an IPv6 packet and then sends the IPv6 packet to NS-1. The IPv6 packet has:

- Source IP address = 2001:DB8::3:4
- Destination IP address = 2001:DB8::5:6

3. When NS-1 receives the IPv6 packet, the AFTR module decapsulates the packet by removing the IPv6 headers. The resulting packet is SUB-1's original IPv4 request packet.

4. The LSN module of NS-1 translates the source IP address and port of the packet to an NAT IP address and NAT port selected from the configured NAT pool. The translated IPv4 packet has:

- Source IP address = 203.0.113.61
- Source port = 3002
- Destination IP address = 198.51.100.250
- Destination port = 80

5. The LSN module also creates an LSN mapping and session entry for this DS Lite session. The mapping includes the following information:

- Source IP address of the IPv6 packet (B4-CPE-1's IPv6 address) = 2001:DB8::3:4
- Source IP address of the IPv4 packet (SUB-1's IPv4 address) = 192.0.2.51
- Source port of the IPv4 packet = 2552
- NAT IP address = 203.0.113.61
- NAT port = 3002

6. NS-1 sends the resulting IPv4 packet to its destination on the Internet.

7. The server for www.example.com processes the request packet and sends a response packet. The IPv4 response packet has:

- Source IP address = 198.51.100.250
- Source port = 80
- Destination IP address = 203.0.113.61

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Destination port = 3002

8. Upon receiving the IPv4 packet, NS-1 examines the LSN mapping and session entries and finds that the IPv4 response packet belongs to a DS Lite session. The LSN module of NS-1 translates the destination IP address and port. The IPv4 packet now has:

- Source IP address = 198.51.100.250
- Source port = 80
- Destination IP address = 192.0.2.51
- Destination port = 2552

9. The AFTR module of NS-1 encapsulates the IPv4 packet in an IPv6 packet and then sends the IPv6 packet to B4-CPE-1. The IPv6 packet has:

- Source IP address = 2001:DB8::5:6
- Destination IP address = 2001:DB8::3:4

10. Upon receiving the packet, B4-CPE-1 decapsualtes the IPv6 packet by removing the IPv6 headers, and then sends the resulting IPv4 packet to CL-1.

Points to Consider before Configuring DS-Lite

Consider the following points before configuring DS-Lite on a NetScaler appliance:

1. You must understand the different components of DS-Lite, described in RFC 6333.

2. A DS-lite configuration on a NetScaler appliance uses the LSN commands sets. In a DS-Lite configuration, the LSN client entity specifies the IPv6 address or IPv6 network address or ACL6 rules for identifying the traffic from the B4 device. A DS-Lite configuration also includes an IPv6 profile, which specifies the IPv6 address AFTR component on a NetScaler appliance. For more information on NetScaler's LSN feature, see Large Scale NAT.

3. For a DS-Lite configuration, the NetScaler appliance supports LSN for IPv4 packets that belong to one of the following protocols only. The NetScaler appliance drops IPv4 packets belonging to other protocols:

- TCP
- UDP
- ICMP

4. The NetScaler appliance supports the following ALGs DS-Lite:
- ICMP
- FTP

5. The NetScaler appliance does not support the following features for DS-Lite:
- Deterministic NAT (**Note**: This feature is supported in release 11.0 build 64.x and later.)
- Static LSN maps (**Note**: This feature is supported in release 11.0 build 64.x and later.)
- Hair Pin flow
- Application layer gateways (ALG) for the following protocols:
  - Session Initiation Protocol (SIP)
  - Real Time Streaming Protocol (RTSP)
  - Point-to-Point Protocol (PPTP)
- Subscriber aware traffic steering

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DS-Lite

A DS-lite configuration on a NetScaler appliance uses the LSN commands sets. In a DS-Lite configuration, the LSN client entity specifies the IPv6 address or IPv6 network address or ACL6 rules for identifying the traffic from the B4 device. For more information on the NetScaler LSN feature, see Large Scale NAT. A DS-Lite configuration also includes an IPv6 profile, which specifies the IPv6 address (of type SNIP6) of the DS-Lite AFTR component on a NetScaler appliance.

Configuring DS-Lite on a NetScaler appliance consists of the following tasks:

- **Set the global LSN parameters.** Global parameters include the amount of NetScaler memory reserved for the LSN feature and synchronization of LSN sessions in a high availability setup.
- **Create an LSN client entity for identifying traffic from B4 CPE devices.** The LSN client entity refers to a set of DS-Lite B4 devices. The client entity includes IPv6 addresses or IPv6 network address or ACL6 rules for identifying the traffic from these B4 devices. An LSN client can be bound to only one LSN group. The command line interface has two commands for creating an LSN client entity and binding a subscriber to the LSN client entity. The configuration utility combines these two operations on a single screen.
- **Create an LSN pool and bind NAT IP addresses to it.** An LSN pool defines a pool of NAT IP addresses to be used by the NetScaler appliance to perform LSN. The command line interface has two commands for creating an LSN pool and binding NAT IP addresses to the LSN pool. The configuration utility combines these two operations on a single screen.
- **Create an LSN IP6 profile**. An LSN IP6 profile defines the IPv6 address of the DS-Lite AFTR component on the NetScaler appliance. The IPv6 address must be one of the NetScaler owned IPv6 address of type SNIP6.
- **(Optional) Create an LSN Transport Profile for a specified protocol.** An LSN transport profile defines various timeouts and limits, such as maximum LSN sessions and maximum ports usage that a subscriber can have for a given protocol. You bind an LSN transport profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. A profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN transport profile with default settings for TCP, UDP, and ICMP protocols is bound to an LSN group during its creation. This profile is called the default transport profile. An LSN transport profile that you bind to an LSN group overrides the default LSN transport profile for that protocol.
- **(Optional) Create an LSN Application Profile for a specified protocol and bind a set of destination ports to it.** An LSN application profile defines the LSN mapping and LSN filtering controls of a group for a given protocol and for a set of destination ports. For a set of destination ports, you bind an LSN profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. An LSN application profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN application profile with default settings for TCP, UDP, and ICMP protocols for all destination ports is bound to an LSN group during its creation. This profile is called a default application profile. When you bind an LSN application profile, with a specified set of destination ports, to an LSN group, the bound profile overrides the default LSN application profile for that protocol at that set of destination ports. The command line interface has two commands for creating an LSN application profile and binding a set of destination ports to the LSN application profile. The configuration utility combines these two operations on a single screen.
- **Create an LSN Group and bind LSN pools, LSN IPv6 profile, (optional) LSN transport profiles, and (optional) LSN application profiles to the LSN group.** An LSN group is an entity consisting of an LSN client, an LSN IPv6 profile, LSN pool(s), LSN transport profile(s), and LSN application profiles(s). A group is assigned parameters, such as port block size and logging of LSN sessions. The parameter settings apply to all the subscribers of an LSN client bound to the LSN group. Only one LSN IPv6 profile can be bound to an LSN group, and an LSN IPv6 profile bound to an LSN group cannot be bound to other LSN groups. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group. Only one LSN client entity can be bound to an LSN group, and an LSN client entity bound to an LSN group cannot be bound to other LSN groups. The command line interface has

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

two commands for creating an LSN group and binding LSN pools, LSN transport profiles, and LSN application profiles to the LSN group. The configuration utility combines these two operations in a single screen.

Configuration Using the Command Line

To create an LSN client by using the command line interface

At the command prompt, type:

- **add lsn client** <clientname>
- **show lsn client**

To bind an IPv6 network or an ACL6 rule to an LSN client by using the command line interface

At the command prompt, type:

- **bind lsn client** <clientname> (-**network6** <ipv6_addr|*>| -**acl6name** <string>)
- **show lsn client**

To create an LSN pool by using the command line interface

At the command prompt, type:

- **add lsn pool** <poolname> [-**nattype** (DYNAMIC)] [-**portblockallocation** (ENABLED | DISABLED)] [-**portrealloctimeout** <secs>] [-**maxPortReallocTmq** <positive_integer>]
- **show lsn pool**

To bind an IP address range to an LSN pool by using the command line interface

At the command prompt, type:

- **bind lsn pool** <poolname> <lsnip>
- **show lsn pool**

Note: For removing LSN IP addresses from an LSN pool, use the **unbind lsn pool** command.

To configure an LSN IPv6 profile by using the command line interface

At the command prompt, type:

- **add lsn ip6profile** <name> –**type DS-Lite** –**network6** < ipv6_addr|*s >
- **show lsn ip6profile**

To create an LSN transport profile by using the command line interface

At the command prompt, type:

- **add lsn transportprofile** <transportprofilename> <transportprotocol> [-**sessiontimeout** <secs>] [-**finrsttimeout** <secs>] [-**portquota** <positive_integer>] [-**sessionquota** <positive_integer>] [-**portpreserveparity** (ENABLED | DISABLED)] [-**portpreserverange** (ENABLED | DISABLED)] [-**syncheck** (ENABLED | DISABLED)]
- **show lsn transportprofile**

To create an LSN application profile by using the command line interface

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

- **add lsn appsprofile** <appsprofilename> <transportprotocol> [-**ippooling** (PAIRED | RANDOM)] [-**mapping** <mapping>] [-**filtering** <filtering>][-**tcpproxy** (ENABLED | DISABLED)] [-**td** <positive_integer>]
- **show lsn appsprofile**

To bind an application protocol port range to an LSN application profile by using the command line interface

At the command prompt, type:

- **bind lsn appsprofile** <appsprofilename> <lsnport>
- **show lsn appsprofile**

To create an LSN group by using the command line interface

At the command prompt, type:

- **add lsn group** <groupname> -clientname <string> [-**nattype** (DYNAMIC)] [-**portblocksize** <positive_integer>] [-**logging** (ENABLED | DISABLED)] [-**sessionLogging** (ENABLED | DISABLED)][-**sessionSync** (ENABLED | DISABLED)] [-**snmptraplimit**<positive_integer>] [-**ftp** (ENABLED | DISABLED)] [-**pptp** (ENABLED |DISABLED)] [-**sipalg** (ENABLED | DISABLED)] [-**rtspalg** (ENABLED |DISABLED **)] [-ip6profile** <string>**]**
- **show lsn group**

To bind LSN protocol profiles and LSN pools to an LSN group by using the command line interface

At the command prompt, type:

- **bind lsn group** <groupname> (-**poolname** <string> | -**transportprofilename** <string> | -**httphdrlogprofilename** <string> | -**appsprofilename** <string> | -**sipalgprofilename** <string> | **rtspalgprofilename** <string>)
- **show lsn group**

Configuration Using the Configuration Utility

To configure an LSN client and bind an IPv6 network address or an ACL6 rule by using the configuration utility

Navigate to **System** > **Large Scale NAT** > **Clients**, and add a client and then bind an IPv6 network address or an ACL6 rule to the client.

To configure an LSN pool and bind NAT IP addresses by using the configuration utility

Navigate to **System** > **Large Scale NAT** > **Pools**, and add a pool and then bind an NAT IP address or a range of NAT IP addresses to the pool.

To configure an LSN IPv6 profile by using the configuration utility

Navigate to **System** > **Large Scale NAT** > **Profiles**, click the **IPv6** tab, and assign an IPv6 address for DS-Lite AFTR.

To configure an LSN transport profile by using the configuration utility

1. Navigate to **System** > **Large Scale NAT** > **Profiles**.
2. On the details pane, click **Transport** tab, and then add a transport profile.

To configure an LSN application profile by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to **System** > **Large Scale NAT** > **Profiles**.
2. On the details pane, click **Application** tab, and then add an application profile.

To configure an LSN group and bind an LSN client, an LSN IPv6 profile, pools, transport profiles, and application profiles by using the configuration utility

Navigate to **System** > **Large Scale NAT** > **Groups**, and add a group and then bind an LSN client, an LSN IPv6 profile, pools, transport profiles, and application profiles to the group.

Example COPY

```
> add lsn client LSN-DSLITE-CLIENT-1

Done

> bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100

Done

> add lsn pool LSN-DSLITE-POOL-1

Done

> bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70

Done

> add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:DB8::5:6

Done

> add lsn group LSN-DSLITE-PROFILE-1 -clientname LSN-DSLITE-CLIENT-1 -portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1

Done

> bind lsn group group1 -poolname LSN-DSLITE-POOL-1

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Logging DS-Lite Information

You can log DS-Lite information to diagnose or troubleshoot problems, and to meet legal requirements. The NetScaler appliance supports all LSN logging features for logging DS-Lite information. For configuring DS-Lite logging, use the procedures for configuring LSN logging, described at Logging and Monitoring LSN.

A log message for a DS-Lite LSN mapping entry consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced.
- Time stamp
- Entry type (MAPPING)
- Whether the DS-Lite LSN mapping entry was created or deleted
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
    - Destination IP address and port are not logged for Endpoint-Independent mapping.
    - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
    - Destination IP address and port are logged for Address-Port-Dependent mapping.

A log message for a DS-Lite session consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced.
- Time stamp
- Entry type (SESSION)
- Whether the DS-Lite session is created or removed
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The following table shows sample DS-Lite log entries of each type stored on the configured log servers. These log entries are generated by a NetScaler appliance whose NSIP address is 10.102.37.115.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

LSN Log Entry Type	Sample Log Entry
DS-Lite session creation	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP
DS-Lite session deletion	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP
DS-Lite LSN mapping creation	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP
DS-Lite LSN mapping deletion	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

Displaying Current DS-Lite Sessions

You can display the current DS-Lite sessions for detecting any unwanted or inefficient sessions on the NetScaler appliance. You can display all or some DS-Lite sessions, on the basis of selection parameters.

Configuration Using the Command Line Interface

To display all DS-Lite sessions by using the command line interface

At the command prompt, type:

show lsn session –nattype DS-Lite

To display selected DS-Lite sessions by using the command line interface

At the command prompt, type:

show lsn session –nattype DS-Lite [-**clientname** <string>] [-**network** <ip_addr> [-**netmask** <netmask>] [-**td** <positive_integer>]] [-**natIP** <ip_addr> [-**natPort** <port>]]

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
The following sample ouput displays all DS-Lite sessions existing on a NetScaler appliance:

> show lsn session –nattype DS-Lite

  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP NatPort Proto Dir

1. 2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61 3002 TCP OUT

2. 2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61 52862 TCP OUT

3. 2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61 48116 ICMP OUT

4. 2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69 48305 TCP OUT

Done
```

Configuration Using the Configuration Utility

To display all or selected DS-Lite sessions by using the configuration utility

1. Navigate to **System** > **Large Scale NAT** > **Sessions**, and click the **DS-Lite** tab.
2. For displaying DS-Lite sessions on the basis of selection parameters, click **Search**.

Clearing DS-Lite Sessions

You can remove any unwanted or inefficient DS-Lite sessions from the NetScaler appliance. The appliance immediately releases the resources (such as NAT IP address, port, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected DS-Lite sessions from the NetScaler appliance.

To clear all DS-Lite sessions by using the command line interface

At the command prompt, type:

- **flush lsn session –nattype** DS-Lite
- **show lsn session –nattype** DS-Lite

To clear selected DS-Lite sessions by using the command line interface

At the command prompt, type:

- **flush lsn session –nattype** DS-Lite [-**clientname** <string>] [-**network** <ip_addr> [-**netmask** <netmask>] [-**td** <positive_integer>]] [-**natIP** <ip_addr> [-**natPort** <port>]]
- **show lsn session –nattype** DS-Lite

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To clear all or selected DS-Lite sessions by using the configuration utility

1. Navigate to **System** > **Large Scale NAT** > **Sessions**, and click the **DS-Lite** tab.
2. Click **Flush Sessions**.

Configuring DS-Lite Static LSN Maps

The NetScaler appliance supports manual creation of DS-Lite LSN mappings, which contain the mapping between the following information:

- Subscriber's IP address and port, and IPv6 address of B4 device or component
- NAT IP address and port

Static DS-Lite LSN mappings are useful in cases where you want to ensure that the connections initiated to a NAT IP address and port map to the subscriber IP address and port through the specified B4 device (for example, web servers located in the internal network).

Note: This feature is supported in release 11.0 build 64.x and later.

To create a DS-Lite static LSN mapping by using the command line

At the command prompt, type:

- **add lsn static** <name> <transportprotocol> <subscrIP> <subscrPort> [-**td** <positive_integer>] [-**network6** <B4_ADDR>] [<natIP> [<natPort>]] [-**destIP**<ip_addr> [-**dsttd** <positive_integer>]]
- **show lsn static**

To create a DS-Lite static LSN mapping by using the configuration utility

Navigate to System > Large Scale NAT > Static, and add a new DS-Lite static LSN mapping.

Parameter Descriptions

add lsn static

name

Name for the LSN static mapping entry. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "ds-lite lsn static1" or 'ds-lite lsn static1'). This is a mandatory argument. Maximum Length: 127

transportprotocol

Protocol for the DS-Lite LSN mapping entry.

subscrIP

IPv4 address of a subscriber for the DS-Lite LSN mapping entry.

subscrPort

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Port of the subscriber for the DS-Lite LSN mapping entry.

Network6

IPv6 address of the B4 device or component.

td

ID of the traffic domain to which the B4 device belongs. The IPv6 address of the B4 device is specified in the network6 paramter. If you do not specify an ID, the B4 device is assumed to be a part of the default traffic domain.

natIP

IPv4 address, already existing on the NetScaler appliance as type LSN, to be used as NAT IP address for this mapping entry.

natPort

NAT port for this DS-Lite LSN mapping entry.

destIP

Destination IP address for the DS-Lite LSN mapping entry.

dsttd

ID of the traffic domain through which the destination IP address for this DS-Lite LSN mapping entry is reachable from the NetScaler appliance. If you do not specify an ID, the destination IP address is assumed to be reachable through the default traffic domain, which has an ID of 0.

Configuring Deterministic NAT Allocation for DS-Lite

Deterministic NAT allocation for DS-Lite LSN deployments is a type of NAT resource allocation in which the NetScaler appliance pre-allocates, from the LSN NAT IP pool and on the basis of the specified port block size, an LSN NAT IP address and a block of ports to each subscriber (subscriber behind B4 device).

Note: This feature is supported in release 11.0 build 64.x and later.

The appliance sequentially allocates NAT resources to these subscribers. It assigns the first block of ports on the beginning NAT IP address to the beginning subscriber IP address. The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. At that point, the first port block on the next NAT address is assigned to the subscriber, and so on.

The NetScaler appliance logs the allocated NAT IP address and the port block for a subscriber. For a connection, a subscriber can be identified by just its mapped NAT IP address and port block. For this reason, the NetScaler appliance does not log the creation or deletion of an LSN session.

A DS-Lite subscriber can have only one deterministic port block. If the entire block of ports is being used, the NetScaler appliance drops any new connection from the subscriber.

Example: Deterministic DS-Lite

In this example, a deterministic DS-Lite configuration includes four subscribers with IP addresses 192.0.17.5, 192.0.17.6, 192.0.17.7, and 192.0.17.8. These ipv4 subscribers are behind a B4 device having the IPv6 address 2001:DB8::3:4. In this configuration, the port block size is set to 20480 and LSN NAT IP address pool has IP addresses in the range 203.0.113.41-

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

203.0.113.42.

The NetScaler appliance sequentially pre-allocates, from the LSN NAT IP pool and on the basis of the set port block size, an LSN NAT IP address and a block of ports to each subscriber. It assigns the first block of ports (1024-21503) on the beginning NAT IP address (203.0.113.41) to the beginning subscriber IP address (192.0.17.5). The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. At that point, the first port block on the next NAT IP address is assigned to the subscriber, and so on. The NetScaler logs the NAT IP address and the block of ports allocated for each subscriber.

The NetScaler appliance does not log any LSN session created or deleted for these subscribers.

The following table lists the NAT IP address and blocks of ports allocated to each subscriber in this example:

Subscriber IP address	Allocated NAT IP address	Allocated Block of Ports	IPv6 address of B4
192.0.17.5	203.0.113.41	1024 - 21503	2001:DB8::3:4
192.0.17.6	203.0.113.41	21504 - 41983	2001:DB8::3:4
192.0.17.7	203.0.113.41	41984 - 62463	2001:DB8::3:4
192.0.17.8	203.0.113.42	1024 - 21503	2001:DB8::3:4

Configuration Steps

You need to configure deterministic NAT as part of the DS-Lite configuration. For instructions on configuring DS-Lite, see Configuring DS-Lite.

While configuring DS-Lite, make sure that you:

- Set the NAT Type parameter to Deterministic when adding the LSN pool and the LSN group.
- Set the desired port block size parameter when adding the LSN group, unless you can accept the default value.

Points to Consider before Configuring Deterministic DS-Lite:

Consider the following points before configuring deterministic DS-Lite:

- The complete IP address of each subscriber must be specified in a separate add lsn client command, by setting the Network and Netmask parameters. (Set Netmask to 255.255.255.255.) Also the IPv4 address of the B4 device specified in Network6 parameter must be complete (/128 prefix). In other words, Network and Network6 parameter do not accept addresses other than /32 bit mask and /128 prefix, respectively.
- The NetScaler appliance drops connections from subscribers that are not specified in any deterministic DS-Lite configuration but are behind B4 devices specified in a deterministic DS-lite configuration.
- The NetScaler appliance recognizes subscribers having the same IPv4 address as different subscribers if they are behind different B4 devices. A combination of subscriber IPv4 address and B4 device defines a unique subscriber in the LSN client entity of a DS-Lite configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following configuration uses the settings listed in section Example: Deterministic DS-Lite.

```
> add lsn client LSN-DSLITE-CLIENT-10

Done

> bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.5 -netmask 255.255.255.255 -network6 2001:DB8::3:4/128

Done

> bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.6 -netmask 255.255.255.255 -network6 2001:DB8::3:4/128

Done

> bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.7 -netmask 255.255.255.255 -network6 2001:DB8::3:4/128

Done

> bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.8 -netmask 255.255.255.255 -network6 2001:DB8::3:4/128

Done

> add lsn pool LSN-DSLITE-POOL-10 -nattype DETERMINISTIC

Done

> bind lsn pool LSN-DSLITE-POOL-10  203.0.113.41-203.0.113.42

Done

> add lsn ip6profile LSN-DSLITE-PROFILE-10 -type DS-Lite -network6 2001:DB8::5:6

Done

> add lsn group LSN-DSLITE-GROUP-10 -clientname LSN-DSLITE-CLIENT-10 -nattype DETERMINISTIC -portblocksize 20480 -ip6profil

Done

> bind lsn group LSN-DSLITE-GROUP-10 -poolname LSN-DSLITE-POOL-10

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Telco Subscriber Management

May 31, 2016

The number of subscribers in a telco network is increasing at an unprecedented rate, and managing them is becoming a challenge for service providers. Newer, faster, and smarter devices are placing high demand on the network and the subscriber management systems. It is no longer feasible to provide each subscriber the same standard of service, and the need for traffic processing on a per-subscriber basis is imperative.

The NetScaler appliance provides the intelligence to profile subscribers on the basis of their information stored in the PCRF. When a mobile subscriber connects to the Internet, the packet gateway associates an IP address with the subscriber and forwards the data packet to the appliance. The appliance receives the subscriber information dynamically or you can configure static subscribers. This information enables the NetScaler to apply its rich traffic management capabilities, such as content switching, integrated caching, rewrite, and responder, on a per-subscriber basis to manage the traffic.

Dynamic Subscribers

The NetScaler appliance dynamically receives the subscriber information through any of the following interfaces:

- Gx Interface
- RADIUS Interface
- RADIUS and Gx Interface

> ### Note
>
> High availability (HA) is supported from release 11.0 build 63.x.
>
> In an HA setup, the subscriber sessions are continually synchronized on the secondary node. In the event of a failover, the subscriber information is still available on the secondary node.

Gx Interface

A Gx interface (as specified in 3GPP 29.212) is a standard interface based on the Diameter protocol that allows exchange of policy control and charging rules between a PCRF and a PCEF entity in a Telco network.

As soon as an IP-CAN session is established, the packet gateway forwards the subscriber ID, such as the MSISDN, and Framed-IP address information about the subscriber to the PCRF as a Diameter message. When the data packet arrives at the appliance from PGW, the appliance uses the subscriber IP address to query the PCRF to get the subscriber information. This is also known as secondary PCEF functionality.

The PCC rules received by the appliance over the Gx interface are stored in the appliance for the duration of the subscriber session, that is, until the PCRF sends a RAR message with a Session-Release-Cause AVP or the subscriber session is terminated from the NetScaler command line or the configuration utility. If there are any updates to an existing subscriber, the PCRF sends the updates in an RAR message. A subscriber session is initiated when a subscriber logs on to the network, and terminated when the subscriber logs off.

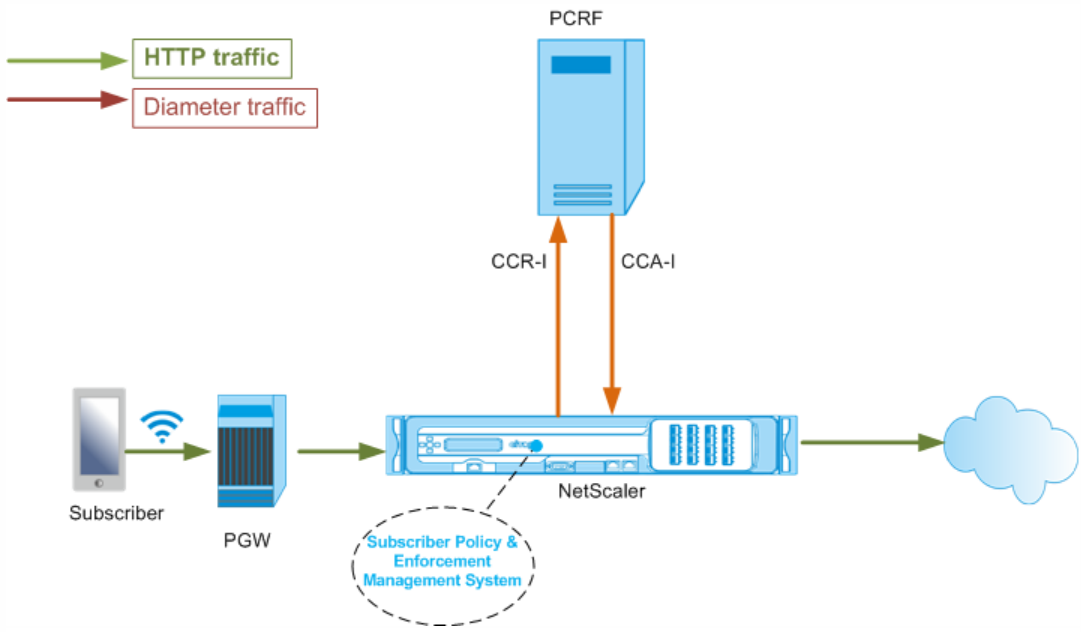
The following illustration shows the high-level traffic flow. It assumes that the data plane traffic is HTTP. The appliance sends a Credit Control Request (CCR) over a Gx interface to the PCRF server and, in the credit control answer (CCA), receives the PCC rules and, optionally, other information, such as the RAT type, that applies to the particular subscriber. PCC rules include one or more policy (rule) names and other parameters. The appliance uses this information to retrieve the predefined rules stored on the appliance, and to direct the flow of traffic. It also stores this information in the subscriber policy and enforcement management system for the duration of the subscriber session. After a subscriber session is terminated, the appliance discards all the information

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

about the subscriber. The following example shows the commands for configuring a Gx interface.

Example COPY

```
set subscriber param -interfaceType GxOnly

add service pcrf-svc 203.0.113.1 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -persistAVPno 263

bind lb vserver vdiam pcrf-svc

set subscriber gxInterface -vServer vdiam -pcrfRealm testrealm1.net -revalidationTimeout 1200 -negativeTTL 120
```



RADIUS Interface

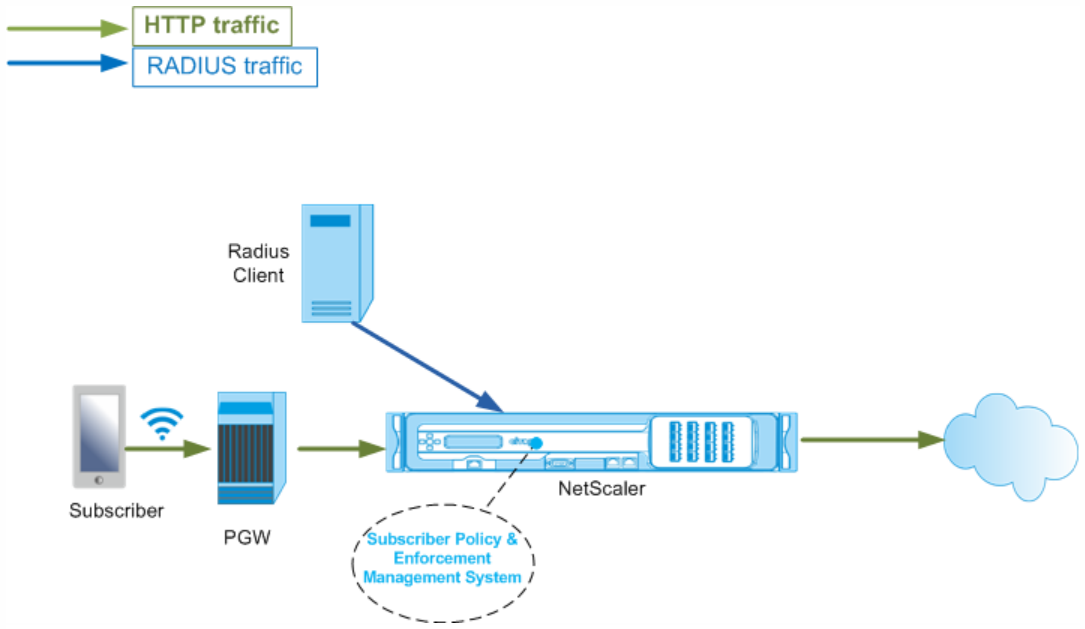
As soon as an IP-CAN session is established, the packet gateway forwards the subscriber information in a RADIUS Accounting Start message to the appliance through the RADIUS interface. A new service type, RADIUSListener, is added to process RADIUS Accounting messages. The following example shows the commands for configuring a RADIUS interface.

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set subscriber param -interfaceType RadiusOnly

add service srad1 192.0.0.206 RADIUSLISTENER 1813

set subscriber radiusInterface -listeningService srad1
```

The following illustration shows the high-level traffic flow.



RADIUS and Gx Interface

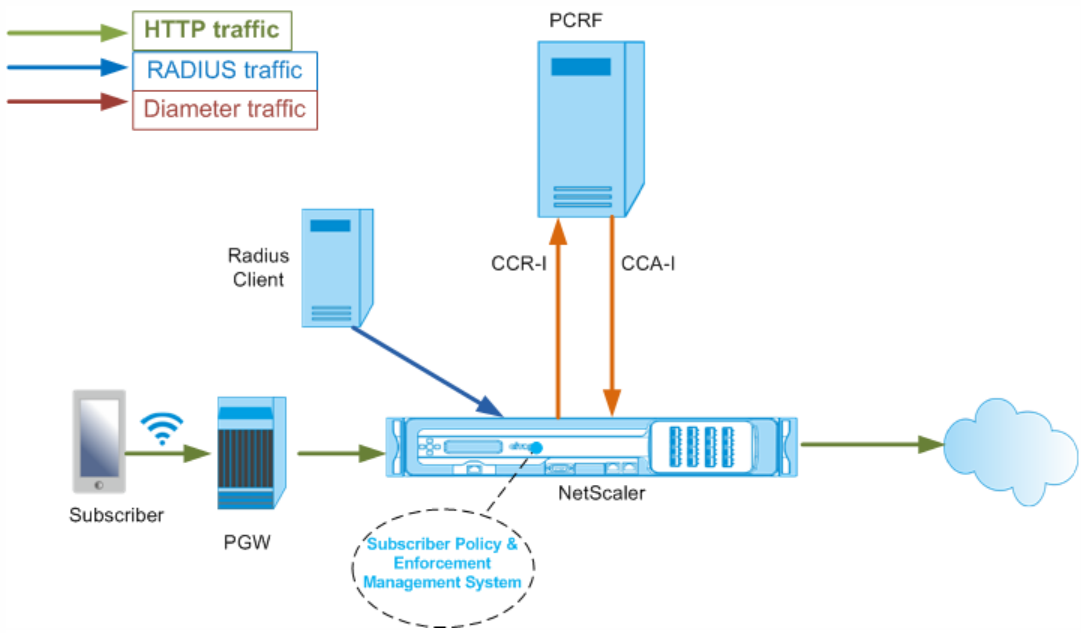
As soon as an IP-CAN session is established, the packet gateway forwards the subscriber ID, such as the MSISDN, and Framed-IP address information about the subscriber to the appliance through the RADIUS interface. The appliance uses this subscriber ID to query the PCRF on the Gx interface to get the subscriber information. This is also known as primary PCEF functionality. The following example shows the commands for configuring a RADIUS and Gx interface.

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set subscriber param -interfaceType RadiusandGx

add service pcrf-svc 203.0.113.1 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -persistAVPno 263

bind lb vserver vdiam pcrf-svc

set subscriber gxInterface -vServer vdiam -pcrfRealm testrealm1.net -holdOnSubscriberAbsence YES -revalidationTimeout 60 -negativeTTL 120

add service srad1 192.0.0.206 RADIUSLISTENER 1813

set subscriber radiusInterface -listeningService srad1
```

The following illustration shows the high-level traffic flow.



Static Subscriber

You can configure the subscribers manually on the NetScaler appliance by using the command line or the configuration utility. You create static subscribers by assigning a unique subscriber ID and optionally associating a policy to each subscriber. The following example shows the command for configuring a static subscriber.

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add subscriber profile 2002::a66:e8d3/64 -subscriberRules policy1 policy3 -subscriptionIdtype E164 –subscriptionIdvalue 98767543212

add subscriber profile * -subscriberRules default_rule



add subscriber profile * -subscriberRules default_rule



add subscriber profile * -subscriberRules default_rule
```

Note: A default profile is used if the appliance cannot find a subscriber session on the appliance.

In the following example, **subscriptionIdvalue** specifies the international telephone number, and **subscriptionIdType** (E164 in this example) specifies the general format for international telephone numbers. Subscriber rules correspond to the Charging-Rule-Install AVP that is used to add a rule that is not already present, or to modify an existing rule as instructed from the PCRF.

Example COPY

```
add subscriber profile 203.0.113.6 -subscriberRules policy1 policy2 -subscriptionIdType E164 –subscriptionIdvalue 98767543211

add subscriber profile 2002::a66:e8d3/64 -subscriberRules policy1 policy3 -subscriptionIdtype E164 –subscriptionIdvalue 98767543212
```

IPv6 Prefix based Subscriber Sessions

A telco user is generally identified by the IPv6 prefix rather than the complete IPv6 address. The NetScaler appliance now uses the prefix instead of the complete IPv6 address (/128) to identify a subscriber in the database (subscriber store). For communicating with the PCRF server (for example, in a CCR-I message), the appliance now uses the framed-IPv6-Prefix AVP instead of the complete IPv6 address. The default prefix length is /64, but you can configure the appliance to use a different value.

To configure the IPv6 prefix by using the command line

set subscriber param [-ipv6PrefixLookupList <positive_integer> ...]

The first example command below sets a single prefix and the second example command sets multiple prefixes.

Examples COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set subscriber param -ipv6PrefixLookupList 64

set subscriber param -ipv6PrefixLookupList 64 72 96
```

To configure the IPv6 prefix by using the configuration utility

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. In the details pane, under **Settings**, click **Configure Subscriber Parameters** and in **IPv6 Prefix Lookup List**, specify one or more prefixes.

Idle Session Management of Subscriber Sessions in a Telco Network

Subscriber session cleanup on a NetScaler appliance is based on control plane events, such as a RADIUS Accounting Stop message, a Diameter RAR (session release) message, or a "clear subscriber session" command. In some deployments, the messages from a RADIUS client or a PCRF server might not reach the appliance. Additionally, during heavy traffic, the messages might be lost. A subscriber session that is idle for a long time continues to consume memory and IP resources on the NetScaler appliance. The idle session management feature provides configurable timers to identify idle sessions, and cleans up these sessions on the basis of the specified action.

A session is considered idle if no traffic from this subscriber is received on the data plane or the control plane. You can specify an update, terminate (inform PCRF and then delete the session), or delete (without informing PCRF) action. The action is taken only after the session is idle for the time specified in the idle timeout parameter.

To configure the idle session timeout and the associated action by using the command line

set subscriber param [-idleTTL <positive_integer>] [-idleAction <idleAction>]

Examples COPY

```
set subscriber param -idleTTL 3600 -idleAction ccrTerminate

set subscriber param -idleTTL 3600 -idleAction ccrUpdate

set subscriber param -idleTTL 3600 -idleAction delete
```

To disable the idle session timeout, set the idle timeout to zero.

set subscriber param – idleTTL 0

To configure the idle session timeout and the associated action by using the configuration utility

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. In the details pane, under **Settings**, click **Configure Subscriber Parameters** and specify an **Idle Time** and **Idle Action**.

Subscriber Policy Enforcement & Management System

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The NetScaler appliance uses the subscriber's IP address as the key to the subscriber policy enforcement and management system.

You can add subscriber expressions to read the subscriber information available in the Subscriber Policy Enforcement & Management System. These expressions can be used with policy rules and actions that are configured for NetScaler features, such as integrated caching, rewrite, responder, and content switching.

The following commands are an example of adding a subscriber-based responder action and policy. The policy evaluates to true if the subscriber rule value is“pol1”.

Example COPY

```
add responder action error_msg respondwith '\"HTTP/1.1 403 OK\r\n\r\n" + \" You are  not authorized to access Internet"'

add responder policy no_internet_access "SUBSCRIBER.RULE_ACTIVE(\"pol1\")" error_msg
```

In the following example, two policies are configured on the appliance. When the appliance checks the subscriber information and the subscriber rule is cache_enable, it performs caching. If the subscriber rule is cache_disable, the appliance does not perform caching.

Example COPY

```
add cache policy nocachepol -rule "SUBSCRIBER.RULE_ACTIVE(\"cache_disable\")" -action NOCACHE

add cache policy cachepol -rule "SUBSCRIBER.RULE_ACTIVE(\"cache_enable\")" -action CACHE -storeInGroup cg1
```

For a complete list of expressions starting with “SUBSCRIBER.” see the *Policy Configuration Guide*.

Subscriber Session Event Logging

The NetScaler appliance currently maintains millions of subscriber sessions in its database (subscriber store) but does not log these messages. Telco administrators need reliable log messages to track the control plane messages specific to a subscriber. They also need historical data to analyze subscriber activities. The appliance now supports logging of RADIUS control plane accounting messages and Gx control plane logging messages. Some of the key attributes are MSISDN and time stamp. By using these logs, you can track a user by using their IP address, and MSISDN if available.

From these logs, you can learn about any activity related to a user, such as the time when a session was updated, deleted, or created (installed). Additionally, error messages are also logged.

Examples

1. The following log entries are examples of GxOnly session update, RADIUSandGX delete, and RADIUSOnly install messages.

09/30/2015:16:38:56 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 159 0 : Session Update, GX MsgType: CCR-U, IP: 100.10.1.1

09/30/2015:17:27:56 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 185 0 : Session Delete, GX MsgType:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

CCR-T, RADIUS MsgType: Stop, IP: 100.10.1.1, ID: E164 - 30000000001

09/30/2015:17:25:05 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 182 0 : Session Install, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001

2. The following log entries are examples of failure messages, such as when a subscriber is not found on the PCRF server and when the appliance cannot connect to the PCRF server.

09/30/2015:16:44:15 GMT Error 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 169 0 : Failure Reason: PCRF failure response, GX MsgType: CCR-I, IP: 100.10.1.1

Sep 30 13:03:01 09/30/2015:16:49:08 GMT 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 176 0 : Failure Reason: Unable to connect to PCRF, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001#000#000#000#000#000#000#000#000#000#000#000#000#000#000#000#000#000#000#000#000#000#000#000

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

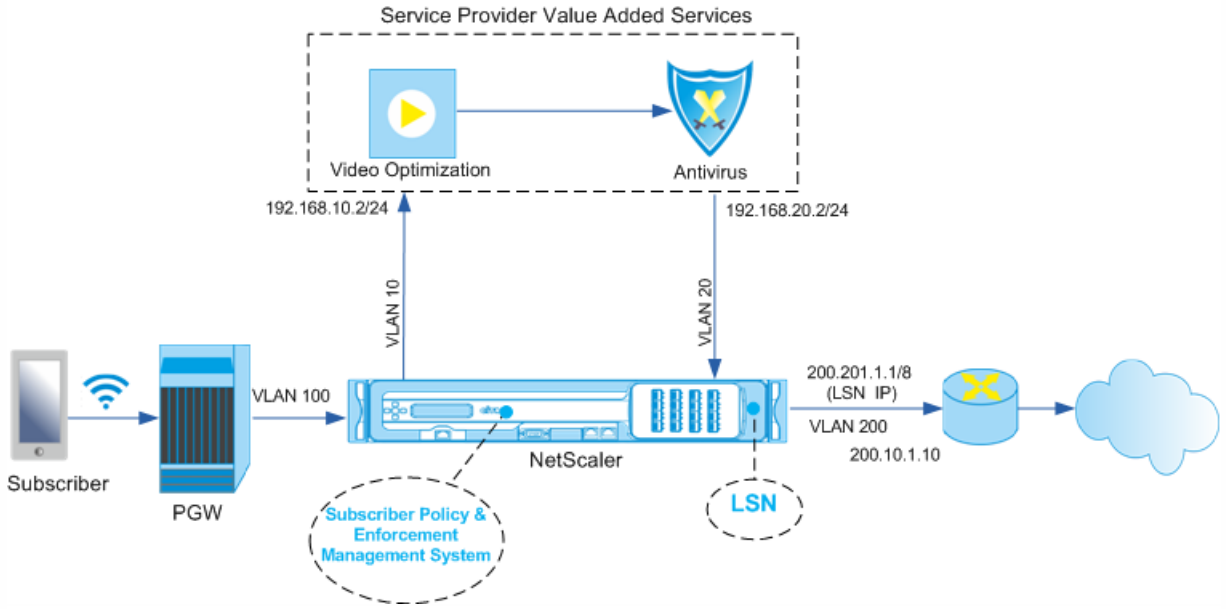
Subscriber-Aware Traffic Steering

Jun 20, 2016

Traffic steering directs subscriber traffic from one point to another. When a subscriber connects to the network, the packet gateway associates an IP address with the subscriber and forwards the data packet to the NetScaler appliance. The appliance communicates with the PCRF server over the Gx interface to get the policy information. Depending on the policy information, the appliance performs one of the following actions:

- Forward the data packet to another set of services (as shown in the following illustration).
- Drop the packet.
- Perform only Large Scale NAT (LSN), if LSN is configured on the appliance.

The values shown in the following figure are configured in the CLI procedure that follows the figure. A content switching virtual server on the NetScaler appliance directs requests to the value added services or skips them, depending on the defined rule, and then sends the packet out to the Internet after performing LSN.



To configure traffic steering for the above deployment by using the NetScaler command line:

1. Add the appliance's subnet IP (SNIP) addresses.

   add ns ip 192.168.10.1 255.255.255.0 –type **snip**

   add ns ip 192.168.20.1 255.255.255.0 –type **snip**

   add ns ip 100.100.100.1 255.0.0.0 –type **snip**

   add ns ip 200.200.200.1 255.0.0.0 –type **snip**

   add ns ip 100.1.1.1 255.0.0.0 –type **snip**

   add ns ip 200.201.1.1 255.0.0.0 –type **snip**

2. Add the VLANs. VLANs help the appliance identify the source of the traffic. Bind the VLANs to the interfaces and subnet

© 1999-2017 Citrix Systems, Inc. All rights reserved.

IP addresses.

add vlan 10

add vlan 20

add vlan 100

add vlan 200

bind vlan 10 -ifnum 1/4 –tagged -IPAddress 192.168.10.1 255.255.255.0

bind vlan 20 -ifnum 1/4 –tagged -IPAddress 192.168.20.1 255.255.255.0

bind vlan 100 -ifnum 1/2 –tagged -IPAddress 100.1.1.1 255.0.0.0

bind vlan 200 -ifnum 1/3 –tagged -IPAddress 200.1.1.1 255.0.0.0

3. Specify the VLAN on which the subscriber traffic arrives on the appliance. Specify the service path AVP that tells the appliance where to look for the service path name within the subscriber session. For primary PCEF functionality, specify the interfaceType as RadiusAndGx.

set ns param -servicePathIngressVLAN 100

set subscriber param -interfaceType **RadiusAndGx**

set subscriber gxinterface -servicepathAVP 1001 1005 -servicepathVendorid 10415

4. Configure a service and virtual server of type Diameter, and bind the service to the virtual server. Then, specify the PCRF realm and subscriber Gx interface parameters. For primary PCEF functionality, configure a RADIUS listener service and RADIUS interface.

add service sd1 10.102.232.200 **DIAMETER** 3868

add lb vserver vdiam **DIAMETER 0.0.0.0 0** -persistenceType **DIAMETER** -persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 **RADIUSListener** 1813

set subscriber radiusInterface -listeningService srad1

5. Add service functions to associate a VAS with an ingress VLAN. Add a service path to define the chain, that is, specify the VAS that the packet must be sent to and the order in which it must go to that VAS. The service path name is usually sent by the PCRF. However, the service path of the default subscriber profile (*) applies if any of the following is true:

- PCRF does not have the subscriber information.
- The subscriber information does not include this AVP.
- The appliance is unable to query the PCRF. For example, the service representing the PCRF is DOWN.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The service path AVP that contains this name must be configured as part of the global configuration earlier. Bind the service function to the service path. The service index specifies the order in which the VAS is added to the chain. The highest number (255) indicates the beginning of the chain.

```
add ns servicefunction SF1 -ingressVLAN 20

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

add subscriber profile * -subscriberrules default_path
```

6. Add the LSN configuration. That is, define the NAT pool and identify the clients for which the appliance must perform LSN.

```
add lsn pool pool1

bind lsn pool pool1 200.201.1.1

add lsn client client1

bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0

add lsn group group1 -clientname client1

bind lsn group group1 -poolname pool1
```

7. The appliance performs LSN by default. To override LSN, you must create a net profile with overrideLsn parameter enabled and bind this profile to all the load balancing virtual servers that are configured for VAS.

```
add netprofile np1

set netprofile np1 -overrideLsn ENABLED

set lb vserver vs1 -netprofile np1
```

8. Configure the value added services (VASs) on the appliance. This includes creating the services and virtual servers and then binding the services to the virtual servers.

```
add service vas1 192.168.10.2 ANY 80 -usip YES

add service sint 200.10.1.10 ANY 80 -usip YES

add lb vserver vs1 ANY -m MAC -l2Conn ON

add lb vserver vint ANY -m MAC -l2Conn ON

bind lb vserver vs1 vas1

bind lb vserver vint sint
```

9. Add the content switching (CS) configuration. This includes virtual servers, policies, and their associated actions. The traffic arrives at the CS virtual server and is then redirected to the appropriate load balancing virtual server. Define expressions that associate a virtual server with a service function.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add cs vserver cs1 ANY * 80 -l2Conn ON

add cs action csact1 -targetLBVserver vs1

add cs action csactint -targetLBVserver vint

add cs policy cspol1 –rule  SUBSCRIBER.SERVICEPATH.IS_NEXT(\"SF1\") &&      SYS.VSERVER(\"vs1\").STATE.EQ(UP)" -action csact1

bind cs vserver cs1 -policyName cspol1 -priority 110

bind cs vserver cs1 -lbvserver vint
```

To configure traffic steering on the appliance by using the configuration utility

1. Navigate to **System** > **Network** > **IPs** and add the subnet IP addresses.
2. Navigate to System > Network > **VLANs** and add VLANs, Bind the VLANs to the interfaces and subnet IP addresses.
3. Navigate to **Traffic Management** > **Service Chaining** > **Configure Service Path Ingress VLAN** and specify an ingress VLAN.
4. Navigate to **Traffic Management** > **Subscriber** > **Parameters** > **Configure Subscriber Parameters** and specify the following:
   - Interface Type: Specify **RadiusAndGx**.
   - Configure a diameter virtual server, PCRF realm, and the subscriber GX interface parameters.
   - Specify the RADIUS interface parameters.
5. Navigate to **Traffic Management** > **Service Chaining** > **Service Function** and add service functions to associate a value-added service with an ingress VLAN.
6. Navigate to **System** > **Network** > **Large Scale NAT**. Click **Pools** and add a pool. Click **Clients** and add a client. Click **Groups** and add a group and specify the client. Edit the group and bind the pool to this group.
7. Navigate to **System** > **Network** > **Net Profiles** and add a net profile. Select **Override LSN**. Optionally, navigate to **System** > **Network** > **Settings** > **Configure Layer 3 Parameters** and verify that **Override LSN** is not selected.
8. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** and configure the virtual servers and value-added services on the appliance. Bind the services and the net profile to the virtual server.
9. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers** and configure a virtual server, policy, and action. Specify the target load balancing virtual server.

To configure service chaining on the appliance by using the configuration utility

1. Navigate to **System** > **Network** > **IPs** and add the subnet IP addresses.
2. Navigate to System > Network > **VLANs** and add VLANs, Bind the VLANs to the interfaces and subnet IP addresses.
3. Navigate to **Traffic Management** > **Service Chaining** > **Configure Service Path Ingress VLAN** and specify an ingress VLAN.
4. Navigate to **Traffic Management** > **Subscriber** > **Parameters** > **Configure Subscriber Parameters** and specify the following:
   - Interface Type: Specify **RadiusAndGx**.
   - Configure a diameter virtual server, PCRF realm, and the subscriber GX interface parameters.
   - Specify the RADIUS interface parameters.
5. Navigate to **Traffic Management** > **Service Chaining** > **Service Function** and add service functions to associate a value-added service with an ingress VLAN.
6. Navigate to **System** > **Network** > **Large Scale NAT**. Click **Pools** and add a pool. Click **Clients** and add a client. Click **Groups** and add a group and specify the client. Edit the group and bind the pool to this group.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

7. Navigate to **System** > **Network** > **Net Profiles** and add a net profile. Select **Override LSN**. Optionally, navigate to **System** > **Network** > **Settings** > **Configure Layer 3 Parameters** and verify that **Override LSN** is not selected.
8. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** and configure the virtual servers and value-added services on the appliance. Bind the services and the net profile to the virtual server.
9. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers** and configure a virtual server, policy, and action. Specify the target load balancing virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Subscriber-Aware Service Chaining

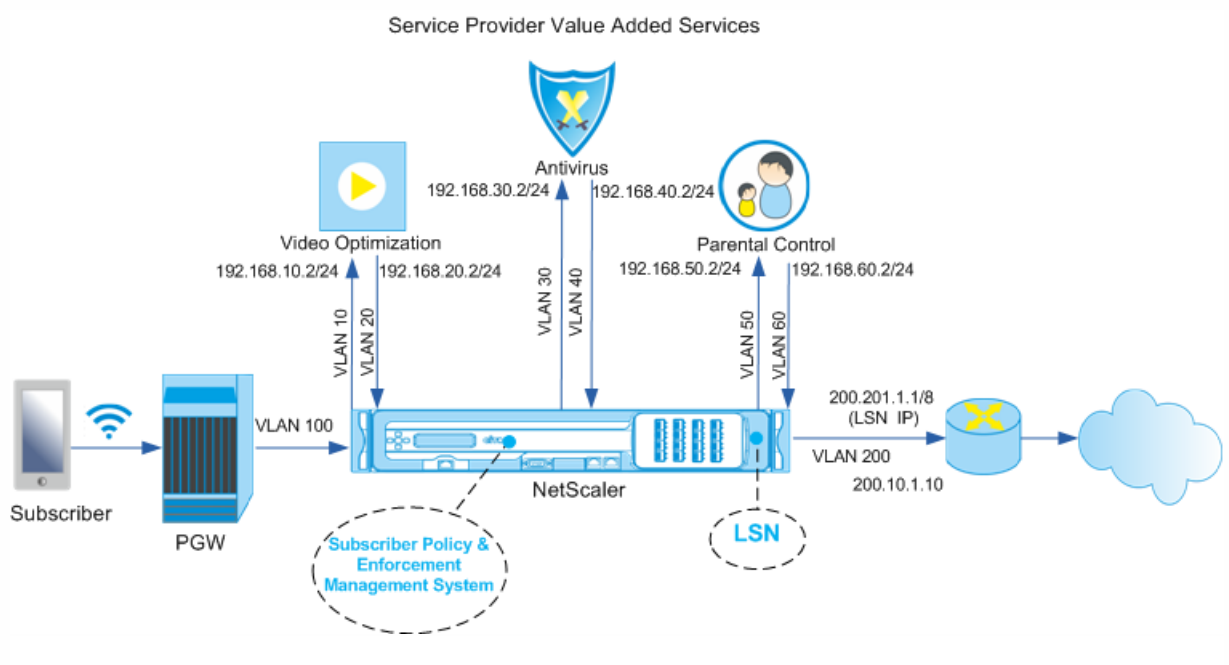
Jun 20, 2016

With the huge increase in the data traffic passing through a telco network, it is no longer feasible for service providers to steer all the traffic through all the value added services (VAS). A service provider should be able to optimize usage of VAS and intelligently steer traffic to improve the user experience. For example, video optimization is not required for traffic that does not include a video. Moreover, if a subscriber is connected to a 4G network, content can be streamed in high definition (HD), and video optimization might not be needed. However, video optimization improves the experience for a user in a 3G network. Similarly, caching provides a faster and better user experience and can be enabled depending on the subscriber plan. Another example of VAS is parental control. If parents provide a mobile handset to a minor child, they would like some kind of control over the websites that their child visits.

To do the above and more, service providers must be able to provide value-added services on a per-subscriber basis. In other words, entities in the service provider network must be capable of extracting the subscriber information and intelligently steering the packet on the basis of this information.

Service chaining determines the set of services through which the traffic from a subscriber must pass before going to the Internet. Instead of sending all the traffic to all the services, the NetScaler intelligently routes all requests from a subscriber to a specific set of services on the basis of the policy defined for that subscriber.

The following figure shows the entities involved in service chaining. The values shown are configured in the procedure that follows the figure. A content switching virtual server on the NetScaler appliance directs requests to the value added services or skips them, depending on the defined rule, and then sends the packet out to the Internet after performing LSN.



To configure service chaining for the above deployment by using the NetScaler command line:

1. Add the appliance's subnet IP (SNIP) addresses.

   add ns ip 192.168.10.1 255.255.255.0 –type **snip**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add ns ip 192.168.20.1 255.255.255.0 –type **snip**

add ns ip 192.168.30.1 255.255.255.0 –type **snip**

add ns ip 192.168.40.1 255.255.255.0 –type **snip**

add ns ip 192.168.50.1 255.255.255.0 –type **snip**

add ns ip 192.168.60.1 255.255.255.0 –type **snip**

add ns ip 100.1.1.1 255.0.0.0 –type **snip**

add ns ip 200.201.1.1 255.0.0.0 –type **snip**

2. Add the VLANs. VLANs help the appliance identify the source of the traffic. Bind the VLANs to the interfaces and subnet IP addresses. Add an ingress and an egress VLAN for each VAS.

add vlan 10

add vlan 20

add vlan 30

add vlan 40

add vlan 50

add vlan 60

add vlan 100

add vlan 200

bind vlan 10 -ifnum 1/4 –tagged -IPAddress 192.168.10.1 255.255.255.0

bind vlan 20 -ifnum 1/4 –tagged -IPAddress 192.168.20.1 255.255.255.0

bind vlan 20 -ifnum 1/4 –tagged -IPAddress 192.168.30.1 255.255.255.0

bind vlan 20 -ifnum 1/4 –tagged -IPAddress 192.168.40.1 255.255.255.0

bind vlan 20 -ifnum 1/4 –tagged -IPAddress 192.168.50.1 255.255.255.0

bind vlan 20 -ifnum 1/4 –tagged -IPAddress 192.168.60.1 255.255.255.0

bind vlan 100 -ifnum 1/2 –tagged -IPAddress 100.1.1.1 255.0.0.0

bind vlan 200 -ifnum 1/3 –tagged -IPAddress 200.201.1.1 255.0.0.0

3. Specify the VLAN on which the subscriber traffic arrives on the appliance. Specify the service path AVP that tells the appliance where to look for the service path name within the subscriber session. For primary PCEF functionality, specify the interfaceType as RadiusAndGx.

set ns param -servicePathIngressVLAN 100

set subscriber param -interfaceType **RadiusAndGx**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

set subscriber gxinterface -servicepathAVP 1001 1005 -servicepathVendorid 10415

4. Configure a service and virtual server of type Diameter, and bind the service to the virtual server. Then, specify the PCRF realm and subscriber Gx interface parameters. For primary PCEF functionality, configure a RADIUS listener service and RADIUS interface.

add service sd1 10.102.232.200 **DIAMETER** 3868

add lb vserver vdiam **DIAMETER 0.0.0.0 0** -persistenceType **DIAMETER** -persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 **RADIUSListener** 1813

set subscriber radiusInterface -listeningService srad1

5. Add service functions to associate a VAS with an ingress VLAN. Add a service path to define the chain, that is, specify the VAS that the packet must be sent to and the order in which it must go to that VAS. The service path name is usually sent by the PCRF. However, the service path of the default subscriber profile (*) applies if any of the following is true:

- PCRF does not have the subscriber information.
- The subscriber information does not include this AVP.
- The appliance is unable to query the PCRF. For example, the service representing the PCRF is DOWN.

The service path AVP that contains this name must be configured as part of the global configuration earlier. Bind the service function to the service path. The service index specifies the order in which the VAS is added to the chain. The highest number (255) indicates the beginning of the chain.

add ns servicefunction SF1 -ingressVLAN 20

add ns servicefunction SF2 -ingressVLAN 40

add ns servicefunction SF3 -ingressVLAN 60

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

bind ns servicepath pol1 -servicefunction SF2 -index 254

bind ns servicepath pol1 -servicefunction SF3 -index 253

add ns servicepath pol2

bind ns servicepath pol2 -servicefunction SF2 -index 255

add ns servicepath pol3

bind ns servicepath pol3 -servicefunction SF1 -index 255

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

add subscriber profile ***** -subscriberrules default_path

6. Add the LSN configuration. That is, define the NAT pool and identify the clients for which the appliance must perform LSN.

add lsn pool pool1

bind lsn pool pool1 200.201.1.1

add lsn client client1

bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0

add lsn group group1 -clientname client1

bind lsn group group1 -poolname pool1

7. The appliance performs LSN by default. To override LSN, you must create a net profile with overrideLsn parameter enabled and bind this profile to all the load balancing virtual servers that are configured for VAS.

add netprofile np1

set netprofile np1 -overrideLsn **ENABLED**

set lb vserver vs1 -netprofile np1

8. Configure the value added services (VASs) on the appliance. This includes creating the services and virtual servers and then binding the services to the virtual servers.

add service vas1 192.168.10.2 **ANY** 80 -usip **YES**

add service vas2 192.168.30.2 **ANY** 80 -usip **YES**

add service vas3 192.168.50.2 **ANY** 80 -usip **YES**

add service sint 200.10.1.10 **ANY** 80 -usip **YES**

add lb vserver vs1 ANY -m **MAC** -l2Conn **ON**

add lb vserver vs2 ANY -m **MAC** -l2Conn **ON**

add lb vserver vs3 ANY -m **MAC** -l2Conn **ON**

add lb vserver vint ANY -m **MAC** -l2Conn **ON**

bind lb vserver vs1 vas1

bind lb vserver vs2 vas2

bind lb vserver vs3 vas3

bind lb vserver vint sint

9. Add the content switching (CS) configuration. This includes virtual servers, policies, and their associated actions. The traffic arrives at the CS virtual server and is then redirected to the appropriate load balancing virtual server. Define

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

expressions that associate a virtual server with a service function.

add cs vserver cs1 **ANY** * 80 -l2Conn **ON**

add cs action csact1 -targetLBVserver vs1

add cs action csact2 -targetLBVserver vs2

add cs action csact3 -targetLBVserver vs3

add cs action csactint -targetLBVserver vint

add cs policy cspol1 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT(\"SF1\") && SYS.VSERVER(\"vs1\").STATE.EQ(UP)" -action csact1

add cs policy cspol2 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT(\"SF2\") && SYS.VSERVER(\"vs2\").STATE.EQ(UP)" -action csact2

add cs policy cspol3 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT(\"SF3\") && SYS.VSERVER(\"vs3\").STATE.EQ(UP)" -action csact3

bind cs vserver cs1 -policyName cspol1 -priority 110

bind cs vserver cs1 -policyName cspol2 -priority 120

bind cs vserver cs1 -policyName cspol3 -priority 130

bind cs vserver cs1 -lbvserver vint

To configure service chaining on the appliance by using the configuration utility

1. Navigate to **System** > **Network** > **IPs** and add the subnet IP addresses.
2. Navigate to System > Network > **VLANs** and add VLANs, Bind the VLANs to the interfaces and subnet IP addresses.
3. Navigate to **Traffic Management** > **Service Chaining** > **Configure Service Path Ingress VLAN** and specify an ingress VLAN.
4. Navigate to **Traffic Management** > **Subscriber** > **Parameters** > **Configure Subscriber Parameters** and specify the following:
   - Interface Type: Specify **RadiusAndGx**.
   - Configure a diameter virtual server, PCRF realm, and the subscriber GX interface parameters.
   - Specify the RADIUS interface parameters.
5. Navigate to **Traffic Management** > **Service Chaining** > **Service Function** and add service functions to associate a value-added service with an ingress VLAN.
6. Navigate to **System** > **Network** > **Large Scale NAT**. Click **Pools** and add a pool. Click **Clients** and add a client. Click **Groups** and add a group and specify the client. Edit the group and bind the pool to this group.
7. Navigate to **System** > **Network** > **Net Profiles** and add a net profile. Select **Override LSN**. Optionally, navigate to **System** > **Network** > **Settings** > **Configure Layer 3 Parameters** and verify that **Override LSN** is not selected.
8. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** and configure the virtual servers and value-added services on the appliance. Bind the services and the net profile to the virtual server.
9. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers** and configure a virtual server, policy, and action. Specify the target load balancing virtual server.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balance Control-Plane Traffic that is based on Diameter, SIP, and SMPP Protocols

May 13, 2016

With the increase in control-plane traffic, the servers can become a bottleneck because the traffic is not optimally distributed among the servers. Therefore, messages must be load balanced. The NetScaler appliance supports Diameter, SIP, and SMPP load balancing.

SIP

NetScaler enables you to load balance SIP messages over UDP or over TCP (including TLS) to a group of proxy servers. NetScaler also provides Call-ID based persistence and Call-ID hash load balancing method using which you direct packets for a particular SIP session to the same load balanced SIP server.
The NetScaler default expressions language contains a number of expressions that operate on Session Initiation Protocol (SIP) connections. These expressions are intended to be used in policies for SIP protocol that operates on a request/response basis. These expressions can be used in content switching, rate limiting, responder, and rewrite policies.

For more information, see Load Balancing a Group of SIP Servers.

SMPP

Millions of short messages are exchanged daily between individuals and value-added service providers, such as banks, advertisers, and directory services, by using the short message peer to peer (SMPP) protocol. Often, message delivery is delayed because servers are overloaded and traffic is not optimally distributed among the servers.
The NetScaler appliance provides optimal distribution of messages across your servers, preventing poor performance and outages. The NetScaler appliance:

- Load balances messages originating from the server and from the client
- Monitors the health of the message centers
- Provides content switching support for the message centers
- Handles concatenated messages

Limitation: Message IDs, from the message center, longer than 59 bytes are not supported. If the message ID length returned by the message center is more than 59 bytes, ancillary operations fail and the NetScaler appliance responds with an error message.

For more information, see SMPP Load Balancing.

Diameter

Diameter is a base protocol with more than 50 protocols (also called applications) built over it. Therefore, the diameter traffic generated in a Telco network is high. To optimally maintain this diameter traffic, the NetScaler appliance performs load balancing, content switching, and acts as a relay agent. Additionally, the appliance offers rewrite and responder functionality. The appliance supports rate limiting of Diameter messages.

For more information, see Configuring Diameter Load Balancing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Provide DNS Infrastructure/Traffic Services, such as, Load Balancing, Caching, and Logging for Telecom Service Providers

May 13, 2016

Telecom service providers can configure the NetScaler appliance to function as a DNS proxy. Caching of DNS records, which is an important function of a DNS proxy, is enabled by default on the NetScaler appliance. This enables the NetScaler to provide quick responses for repeated translations and hence enhances the customer experience and also saves the bandwidth. The caches responses from DNS name servers. When the appliance receives a DNS query, it checks for the queried domain in its cache. If the address for the queried domain is present in its cache, the NetScaler returns the corresponding address to the client. Otherwise, it forwards the query to a DNS name server that checks for the availability of the address and returns it to the NetScaler. The NetScaler then returns the address to the client.

For requests for a domain that has been cached earlier, the NetScaler serves the Address record of the domain from the cache without querying the configured DNS server and hence saves the bandwidth.

From 11.0 release onwards, NetScaler also logs the DNS requests that it receives and also the responses that it sends to the client. Telecom service providers can use this log to:

- Audit the DNS responses to the client
- Audit DNS clients
- Detect and prevent DNS attacks
- Troubleshooting

For more information, see Domain Name System.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Provide Subscriber Load Distribution Using GSLB Across Core-Networks of a Telecom Service Provider

May 13, 2016

Scalability, high availability and performance are critical to service provider deployments. While many service providers deploy there infrastructure at a single location or multiple location, these deployments are subject to a number of inherent limitations, such as:

- If the site loses connectivity to all or part of the public Internet, it will be inaccessible to users and customers, which can have significant impact on the business.
- Users accessing the site from geographically distant locations may experience large and highly variable delays, which are exacerbated by the large number of round trips that HTTP requires to transfer content.

NetScaler appliance's Global Server Load Balancing (GSLB) overcomes these problems by distributing traffic among sites deployed in multiple geographic locations. By serving content from many different points in the Internet, GSLB alleviates the impact of network bandwidth bottlenecks and provides robustness in case of network failures at a particular site. Users can be automatically directed to the nearest or least loaded site at the time of the request, minimizing the likelihood of long download delays and/or service disruptions.
You can use NetScaler appliance's global server load balancing for:

- Disaster recovery or high availability by configuring an Active-standby data center setup that consists of an active and a standby data center. When a failover occurs as a result of a disaster event, the standby data center becomes operational.
- High availability and speed by configuring an active-active data center setup that consists of multiple active data centers. Client requests are load balanced across active data centers.
- Directing client requests to the data center that is closest in geographical distance or network distance by configuring a proximity setup.
- Full-DNS resolutions, GSLB processes DNS queries of the A, AAAA and CNAME types, and the DNS function option can process DNS queries of all other types, such as MX and PTR. Also, if the recursive resolution is enabled, the NetScaler will forward DNS queries for domain names that are not configured on the NetScaler appliance.

For more information, see Global Server Load Balancing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Bandwidth Utilization Using Cache Redirection Functionality

May 13, 2016

The volume of web traffic on the Internet is enormous and a large percentage of that traffic is redundant. Multiple clients ask web servers for the same content repeatedly leading to inefficient use of bandwidth. To relieve the origin web server of processing each request, Internet Service Providers (ISPs) can use the cache redirection feature of NetScaler and serve the content from a cache server instead of from the origin server. The NetScaler analyzes incoming requests, sends requests for cacheable data to cache servers, and sends non-cacheable requests and dynamic HTTP requests to origin servers. Cache redirection feature of NetScaler is policy-based, and by default, requests that match a policy are sent to the origin server, and all other requests are sent to a cache server. You can combine content switching with cache redirection to cache selective content and serve content from specific cache servers for specific types of requested content.

For more information, see Cache Redirection.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Optimizing TCP Performance using TCP NILE

May 13, 2016

The Transmission Control Protocol (TCP) has long been used to establish and manage Internet connections, handle transmission errors, and smoothly connect web applications with client devices. But network traffic has become more difficult to control, because packet loss does not depend only on the congestion in the network, and congestion does not necessarily cause packet loss. Therefore, to measure congestion, a TCP algorithm should focus on both packet loss and bandwidth.

Citrix Systems has developed a new congestion-control algorithm, NILE, a TCP optimization algorithm designed for high-speed networks such as LTE, LTE advanced and 3G. Nile addresses unique challenges caused by fading, random or congestive losses, link layer retransmissions and carrier aggregation.

The NILE algorithm:

- Bases queue-latency estimates on round-trip time measurements.
- Uses a congestion-window-increase function that is inversely proportional to the measured queue latency. This method results in approaching the network congestion point more slowly than does the standard TCP method, and reduces the packet losses during congestion.
- Can distinguish between random loss and congestion based loss on the network by using the estimated queue latency.

The telecom service providers can use the NILE algorithm in their TCP infrastructure to:

- **Optimize mobile and long-distance networks**— The NILE algorithm achieves higher throughput compared to standard TCP. This feature is especially important for mobile and long-distance networks.
- **Decrease application perceived latency and enhance subscriber experience**— The Nile algorithm uses packet loss information to determine whether the transmission-window size should be increased or decreased, and uses queuing delay information to determine the size of the increment or decrement. This dynamic setting of transmission-window size decreases the application latency on the network.

To configure NILE support using the command line interface

At the command prompt, type the following:

set ns tcpProfile <name> [-flavor NILE]

Configuring NILE support using the configuration utility

1. Navigate to **System** > **Profiles** > **TCP Profiles** and click **TCP profiles**.
2. From the **TCP Flavor** drop-down list, select **NILE**.

Example

set ns tcpProfile tcpprofile1 -flavor NILE

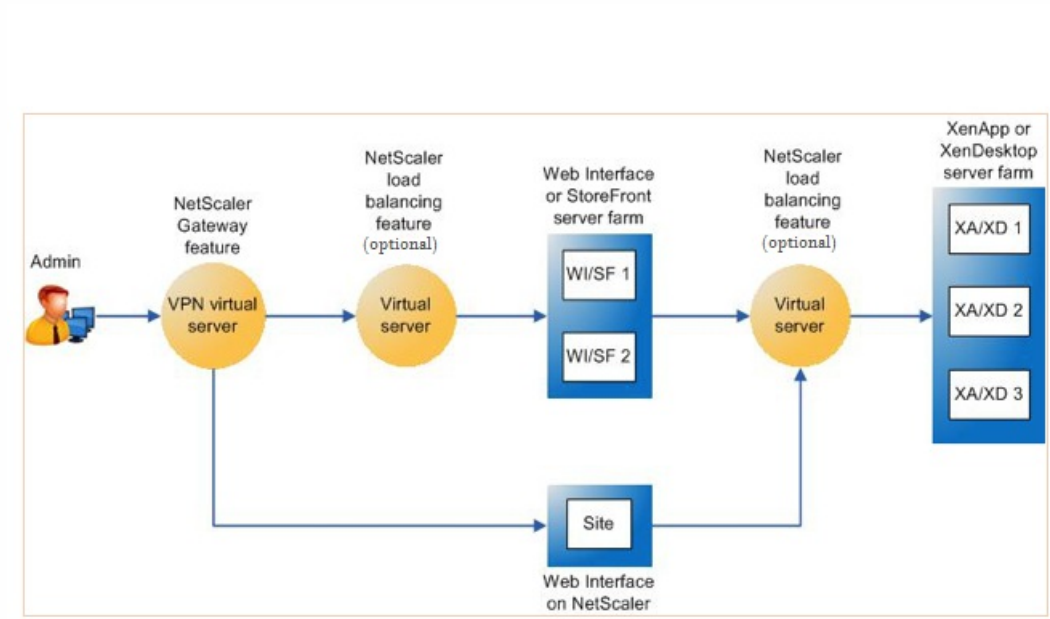
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting Up NetScaler for XenApp/XenDesktop

Dec 27, 2016

A NetScaler appliance can provide load balanced, secure remote access to your XenApp/XenDesktop applications. You can use the NetScaler load balancing feature to distribute traffic across the XenApp/XenDesktop servers, and the NetScaler Gateway feature to provide secure remote access to the servers. NetScaler can also accelerate and optimize the traffic flow and offer visibility features that are useful for XenApp/XenDesktop deployments.

Figure 1. NetScaler Appliance in XenApp/XenDesktop Setup



The above figure shows the components involved in this deployment:

- **NetScaler Gateway**. Provides the URL for user access, and provides security by authenticating the users.
- **NetScaler load balancing virtual server**. Load balances the traffic for the Web Interface or StoreFront servers. You can also deploy a load balancing virtual server in front of the XenApp/XenDesktop servers to load balance key components such as XML Broker and Desktop Delivery Controller (DDC) server.
- **Web Interface or StoreFront or Web Interface on NetScaler.** Provides the interface through which you can access the applications.

  Note: Web Interface on NetScaler (WIonNS) is a customization of the Web Interface product, hosted on the NetScaler appliance.
- **XenApp/XenDesktop**. Provides the applications that your users want to access.

To set up the NetScaler for XenApp/XenDesktop by using the NetScaler GUI

Prerequisites

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- XenApp/XenDesktop servers are configured and available.
- Web Interface, StoreFront, or Web Interface on NetScaler servers are configured and available.
- You have a working knowledge of NetScaler Gateway, NetScaler, XenApp, XenDesktop, and StoreFront/Web Interface/Web Interface on NetScaler. For more information, see "Citrix eDocs."
- You have configured a virtual server and a service and bound the service to the virtual server.

Procedure:

1. Log on to the NetScaler appliance and on the **Configuration** tab click **XenApp and XenDesktop**.
2. On the **Details** pane, click **Get Started**. If the setup exists on the NetScaler, click the **Edit** link corresponding to each of the section that you want to modify.
3. Select the product (StoreFront, Web Interface, or Web Interface on NetScaler) that in your deployment provides the interface for access to the XenApp/XenDesktop applications.
4. Set up secure remote access.
   1. In the **NetScaler Gateway Settings** section, specify the details for the VPN virtual server and click **Continue**.
   2. In the **Server Certificate** section, choose an existing certificate or install a new certificate and click **Continue**.
   3. In the **Authentication** section, configure the primary authentication mechanism to be used and specify the server details or use an existing server and click **Continue**.
   4. In the **StoreFront** section, specify the details of the server that provides the interface for accessing the applications and click **Continue**.
   5. You can use one of the following as your StoreFront server.
      1. LB vserver pointing to multiple SF servers.
      2. Web Interface or StoreFront server directly reachable from the NetScaler appliance.
      3. Web Interface on NetScaler.
5. Click **Done** to complete the configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

RISE Integration: NetScaler ADC and Cisco Nexus 7000 Series Switch

May 13, 2016

Application-delivery solutions that use service modules or switches have required complex configurations and changes in the networking stack. Cisco Remote Integrated Service Engine (RISE) technology can logically integrate a Citrix NetScaler application delivery controller (ADC) with a Cisco Nexus 7000 Series switch as a virtual service module. The integration eliminates the complexities and limitations of traditional inline and one-arm mode configurations. The NetScaler functionality is available as a centralized resource that can be leveraged across the application infrastructure supported by the Cisco Nexus 7000 series switch. The RISE integration provides a service module's streamlined deployment and simplified configuration and operation, and the complete NetScaler application-delivery capabilities that can accelerate application performance for all users.

An ADC deployed in an inline mode can create a bottleneck in the data center. Today, data-center traffic can handle traffic in the terms of terabytes per second, but the ADC capacity can scale up only to gigabytes per second range. The NetScaler ADC, with the new capability to integrate with Cisco Nexus 7000 switch using RISE, can be deployed in a one-arm mode and can provide ADC capabilities to data-center traffic with minimal configuration and maintenance overhead.

A one-arm mode deployment uses Source NAT (SNAT) and Policy Based Routing (PBR) to send only the necessary traffic through the ADC. With SNAT, the servers do not have visibility of client IP addresses. PBR resolves this issue, but it requires complex, manual configuration and is prone to errors. The Automatic Policy Based Routing (APBR) feature in RISE eliminates the need for Source NAT or manual PBR configuration in a one-arm mode design.

For more information about RISE functionality and other features, see Cisco Remote Integrated Service Engine for Citrix NetScaler Appliances and Cisco Nexus 7000 Series Switches Configuration Guide.

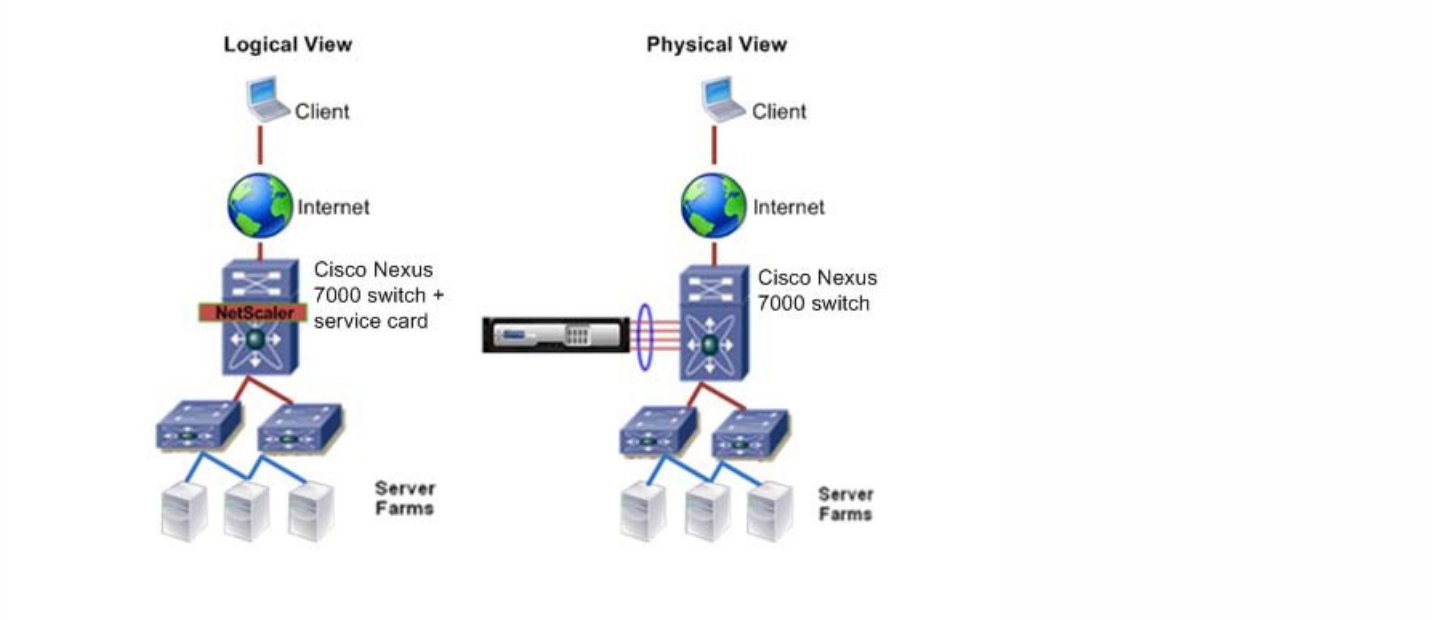
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding RISE

May 29, 2014

Cisco RISE technology logically integrates a Citrix NetScaler ADC with a Cisco Nexus 7000 Series switch as a virtual service module. After you connect the NetScaler ADC and the Cisco Nexus 7000 series switch, an initial handshake is performed and a control channel is established between the two devices to exchange port-channel information. The following figure shows the RISE deployment:

Figure 1. RISE Deployment



Because the NetScaler ADC appears to be a virtual module in the switch, client traffic that reaches the Cisco Nexus 7000 series switch is intelligently routed to the NetScaler ADC and then to the servers. The return traffic flows to the ADC through the Cisco switch, and then back to the client.

The interface or port-channel that connects the NetScaler ADC and Cisco Nexus 7000 series switch is a single trunk carrying both control and data VLANs. The control VLAN is used for all control channel communication, and the data VLAN is used for communicating data traffic.

For more information, see Cisco RISE Integration Overview.

This document includes the following:

- RISE Functionality
- RISE Network Topologies
- RISE Connection Modes

RISE Functionality

Updated: 2014-05-19

The feature integration that RISE enables between the NetScaler ADC and the Cisco Nexus 7000 Series switch provides the following functionalities:

- Plug and play auto-provisioning

© 1999-2017 Citrix Systems, Inc. All rights reserved.

RISE provides a plug and play auto-provisioning feature. You can directly connect the NetScaler ADC to the Cisco Nexus 7000 series switch.

- Discovery and bootstrapping
  The discovery and bootstrap mechanism enables the Cisco Nexus 7000 Series switch to perform the initial setup of NetScaler automatically by exchanging information such as NSIP and VLANs to set up a RISE channel, which transmits control and data packets. For details, see Discovery and Bootstrap.

- Health Monitoring
  The NetScaler ADC uses its health monitoring feature to track and support server health by sending health probes to verify server responses. The Intelligent Services Control Manager (iSCM) on the Cisco Nexus 7000 Series switch and the Intelligent Services Control Client (iSCC) on the NetScaler ADC also periodically send heartbeat packets to each other. If a critical error occurs and health monitoring detects a service instance failure, or if the heartbeat is missed six times successively, the RISE channel becomes nonoperational. For details, see Health Monitoring.

- APBR
  Automatic Policy Based Routing (APBR) automatically routes the return traffic from the servers to the NetScaler ADC, preserving the client IP addresses. The automatic policy based routes are defined on the Cisco Nexus 7000 series switch. When the return traffic from the server reaches the Cisco Nexus 7000 series switch, the APBR policies defined on the switch route the traffic to the NetScaler ADC, which in turn routes the traffic to the client.

  Note:
  - APBR can function only if USIP is enabled on the NetScaler ADC.
  - APBR can be deployed in a VPC mode or a non-VPC mode. For more details on VPC mode, see Cisco VPC.

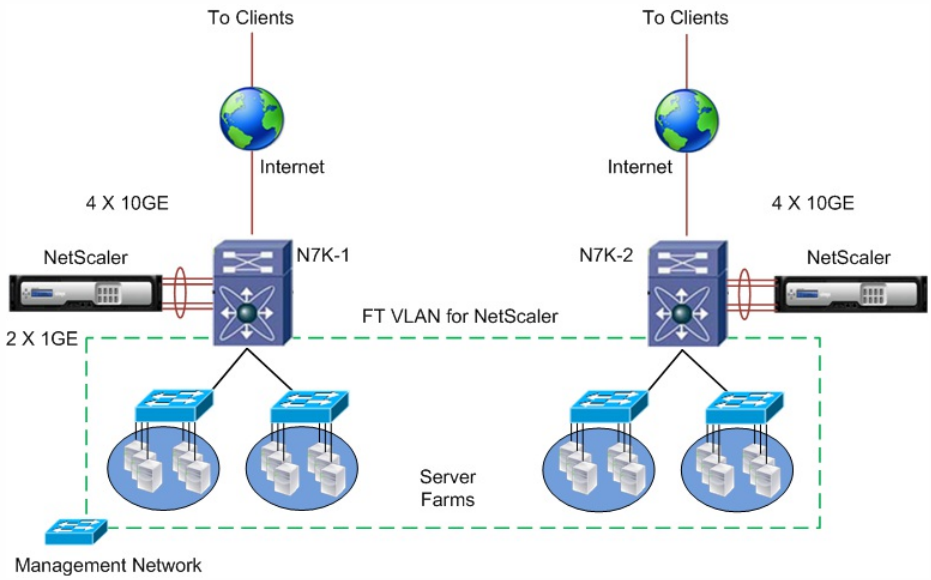
  For details on configuring APBR, see Configuring Auto Policy-Based Routing.

RISE Network Topologies

Updated: 2014-05-19

RISE can be deployed in any of the following modes:

- One-Arm mode— The NetScaler ADC's ports are bundled as a port channel connected to the Cisco Nexus 7000 Series switch. In one-arm mode, the ADC is configured with a VLAN that handles both client and server requests.
  Figure 2. One-Arm Mode

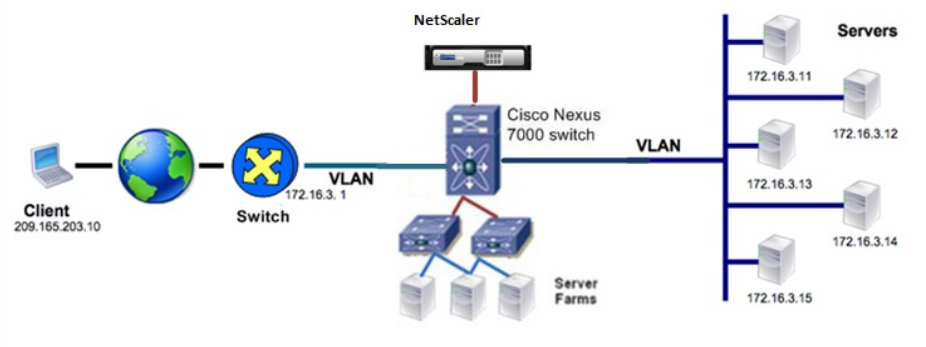


- Bridged mode— In bridged mode, the NetScaler ADC bridges traffic between two VLANs in the same IP subnet. The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

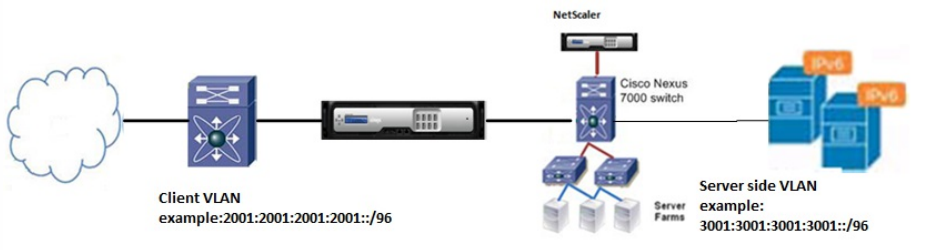
VLAN facing the WAN is the client VLAN. The VLAN facing the data center is the server VLAN. A bridge group virtual interface (BVI) joins the two VLANs into one bridge group.

Figure 3. Bridged Mode



- Routed mode—In routed mode, the NetScaler ADC is the next hop in the network, typically with the client-side VLAN and the server-side VLAN in different IP subnets or in different IP networks.

  Figure 4. Routed Mode



RISE Connection Modes

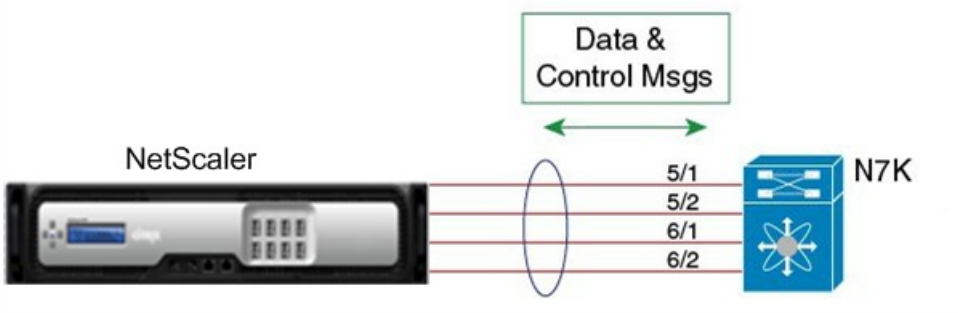
Updated: 2014-05-27

You can connect the Citrix NetScaler appliance to the Cisco Nexus 7000 Series switch in one of the following ways:

Direct Connect Mode for a Standalone Switch

In a direct mode deployment, the NetScaler ADC is attached to a single Nexus 7000 Series switch. The switch can be standalone device or a vPC peer.

Figure 5. Direct Connect Mode

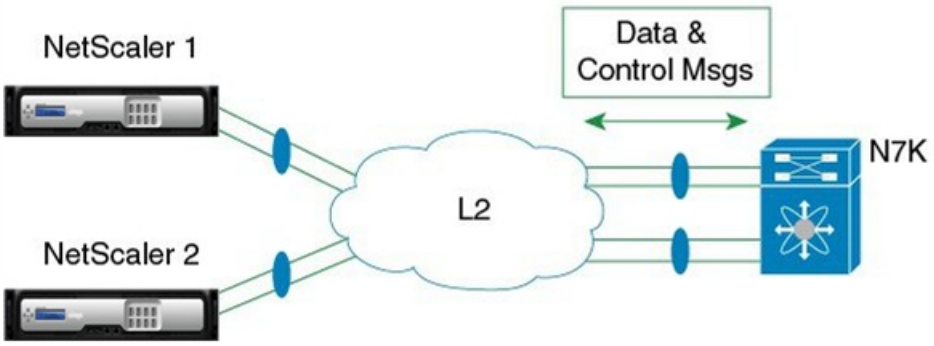


Indirect Connect Mode

In an indirect mode deployment, a virtual NetScaler ADC is connected to a Cisco Nexus 7000 Series switch through a switched layer 2 network.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

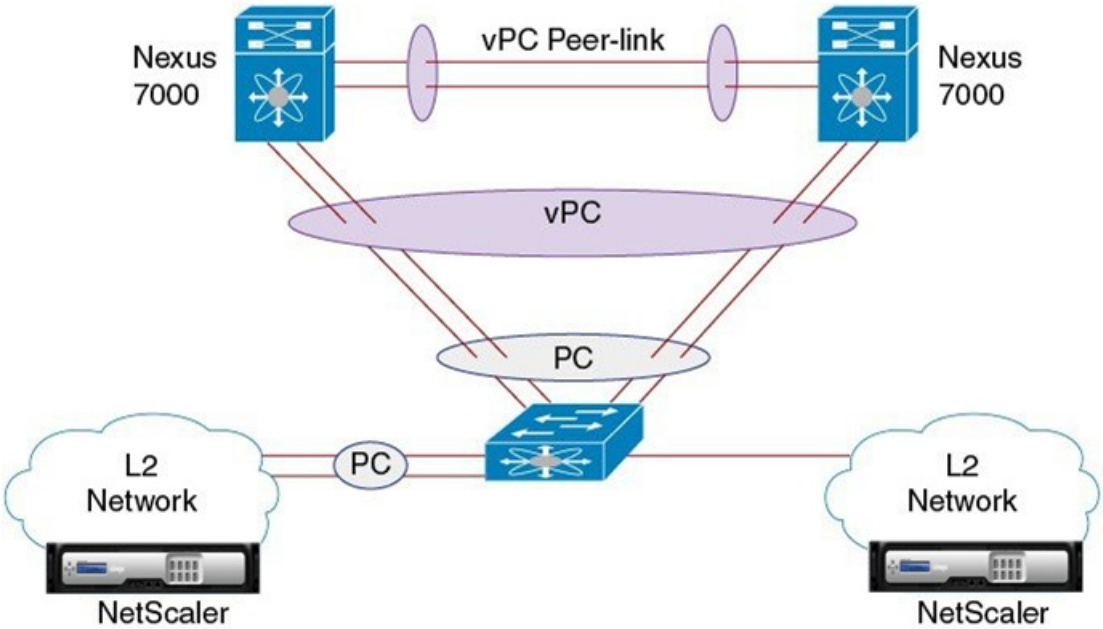
Figure 6. Indirect Connect Mode



Virtual Port Channel (vPC) Connect Mode

In a virtual port channel (vPC) direct mode deployment, the NetScaler ADC is attached to a single Nexus 7000 Series switch that is a vPC peer.

Figure 7. Virtual Port Channel (vPC) Connect Mode

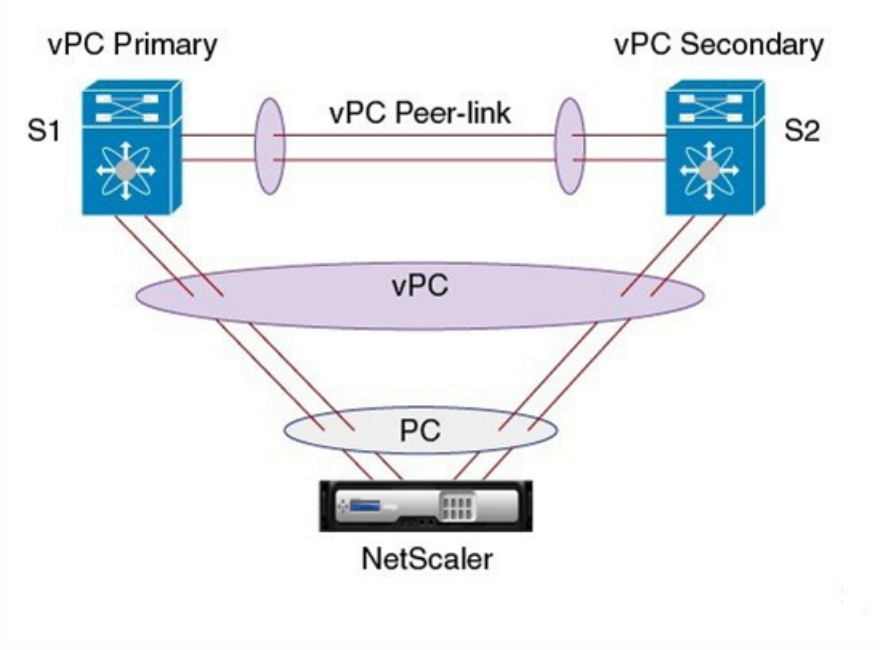


vPC Indirect Connect Mode

In a vPC indirect mode deployment, the NetScaler ADC is indirectly attached to a Cisco Nexus vPC peer through a layer 2 network.

Figure 8. vPC Indirect Connect Mode

© 1999-2017 Citrix Systems, Inc. All rights reserved.



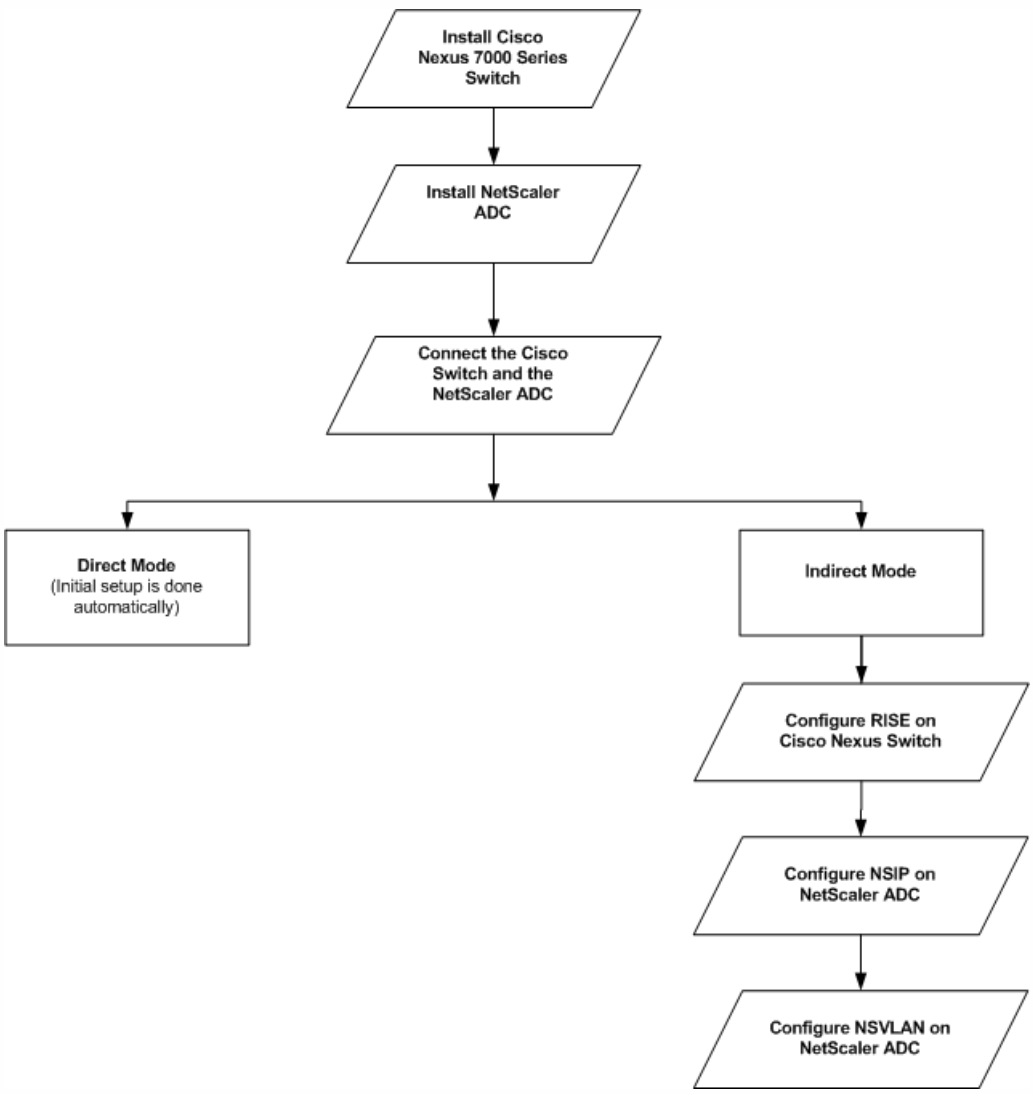
For more information on connection modes, see Connection Modes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Getting Started with RISE

May 19, 2014

To begin using the RISE features, first install the Cisco Nexus 7000 series switch and the NetScaler ADC. Then, connect the NetScaler ADC to the Cisco Nexus 7000 switch either directly or indirectly. If you plan to connect the NetScaler ADC directly to the Cisco Nexus 7000 switch, the initial setup is done automatically by the auto-discovery feature. The following figure explains the flow of steps in the configuration of RISE.



This document includes the following:

- Installing the Cisco Nexus Switch and the NetScaler ADC
- Accessing the Cisco Nexus Switch and the NetScaler ADC
- Configuring RISE
- Configuring High Availability

Installing the Cisco Nexus Switch and the NetScaler ADC

Updated: 2014-05-26

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You need to first install the Cisco Nexus 7000 series switch, and then install the NetScaler ADC.

Installing Cisco Nexus 7000 Series Switch

To install the Cisco Nexus 7000 Series Switch, see Installing Cisco Nexus 7000 Series Switch.

Installing NetScaler ADC

To install the NetScaler ADC, see Installing NetScaler ADC.

To install a virtual NetScaler ADC appliance, see Installing NetScaler Virtual Appliances on XenServer or Installing NetScaler Virtual Appliances on VMware ESX.

Accessing the Cisco Nexus Switch and the NetScaler ADC

Updated: 2014-05-26

After you have completed the installation process, you can access the Cisco Nexus Series 7000 switch through the command-line interface (CLI) and the Citrix NetScaler appliance through the graphical user interface (GUI) or the CLI. To perform administrative tasks and also, to configure RISE, you need to access the Cisco switch and the NetScaler ADC.

Accessing the Cisco Nexus Series 7000 Switch

After installing the Cisco Nexus 7000 series switch, you can access it using the command line interface. To access the Cisco Nexus 7000 Series Switch, see Accessing the Cisco Nexus 7000 series switch.

Accessing the NetScaler ADC

A NetScaler appliance has both a command line interface (CLI) and a graphical user interface (GUI). The GUI includes a configuration utility for configuring the appliance and a statistical utility, called Dashboard. For initial access, all appliances ship with the default NetScaler IP address (NSIP) of 192.168.100.1 and default subnet mask of 255.255.0.0. You can assign a new NSIP and an associated subnet mask during initial configuration.
Note: If you are using the direct connect mode to connect the appliance to the Cisco Nexus switch, you are not required to access the Citrix Netscaler appliance to configure RISE. For direct connect mode, the IP address and VLAN for management are pushed from the Cisco Nexus switch as part of RISE simplified provisioning.
For information about the procedures to access through CLI and GUI, see Using the Command Line Interface and Using Graphical User Interface.

Configuring RISE

Updated: 2015-03-18

After you install the Cisco Nexus 7000 series switch and the NetScaler ADC, configure the appliance to work in direct mode, indirect mode, VPC direct mode, or VPC indirect mode.

Direct Mode

The NetScaler ADC which is directly connected to the switch is automatically configured for RISE mode and all of its ports are in operation mode. No configuration is required on the appliance in a direct mode deployment. For details on configuring RISE in direct mode, see Configuring RISE in Direct Mode.

Indirect Mode

For indirect mode deployment, configure RISE on Cisco Nexus switch, and then configure NSIP and NSVLAN on NetScaler

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

ADC.

- Configuring RISE on Cisco Nexus Switch
- Configuring NSIP on NetScaler ADC
- Configuring NSVLAN on NetScaler ADC

If the direct mode is disabled. all Layer 2 (L2) discovery messages are dropped and L2 RISE discovery does not take place. While, if the direct mode is enabled, the L2 discovery messages are parsed. If L2 RISE discovery messages are successfully parsed, the Indirect Mode is automatically enabled.

If indirect mode is disabled, NetScaler RISE daemon does not listen on TCP ports 8000 and 8001. As a result, RISE heartbeats stop and RISE profiles go down. While, if indirect mode is enabled, NetScaler RISE daemon starts listening on TCP ports 8000 and 8001. As a result, RISE heartbeats may be exchanged and RISE profiles may come up.

The default setting for RISE is direct attach mode as enabled and indirect mode as disabled. With these settings, the direct attach mode works. However, indirect mode will not work. If you want to deploy NetScaler in an indirect mode setting with N7K then use the following setting:

set rise param -indirectMode ENABLED

With active RISE profiles in direct mode, you must consider the followings points:

1. The set rise param -directMode DISABLED command displays a warning message; Warning: Disabling direct mode with one or more active RISE profile(s) on DIRECT ATTACH mode may cause corresponding RISE profile(s) to not come UP on reboot
2. The set rise param -indirectMode DISABLED displays a warning message; Warning: Disabling indirect mode will cause any active RISE profile(s) to go down

However, if the direct mode is enabled, and NetScaler starts receiving L2 RISE discovery messages, it automatically turns the indirect mode as enabled. Otherwise, RISE direct mode does not work.

With active RISE profiles in Indirect Attach mode, the set rise param -indirectMode DISABLED command displays a warning message; Warning: Disabling indirect mode will cause any active RISE profile(s) to go down leading to RISE profiles on Indirect Attach going down.

vPC Direct Mode

The NetScaler ADC which is directly connected to the switch is automatically configured for RISE mode and all of its ports are in operation mode. No configuration is required on the appliance in a direct mode deployment. For details on configuring RISE in direct mode, see Configuring RISE in vPC Direct Mode.

vPC Indirect Mode

In a vPC indirect mode deployment, the NetScaler ADC is indirectly attached to a Cisco Nexus vPC peer through a Layer 2 network.

- Configuring RISE on Cisco Nexus Switch
- Configuring NSIP on NetScaler ADC
- Configuring NSVLAN on NetScaler ADC

For details on configuring RISE in direct mode, see Configuring RISE in vPC Direct Mode.
For information on configuring RISE, see Configuring RISE.

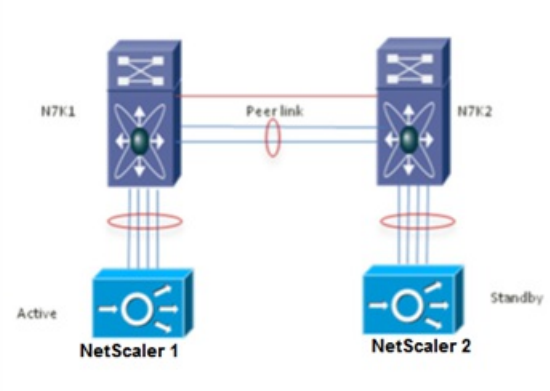
Configuring High Availability

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2014-05-26

In a High Availability setup, the RISE deployment uses a maximum of two NetScaler ADCs. If one ADC becomes unavailable, the traffic flow is seamlessly switched to the other ADC.

Figure 1. High Availability



For information about high availability configuration, see High Availability.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

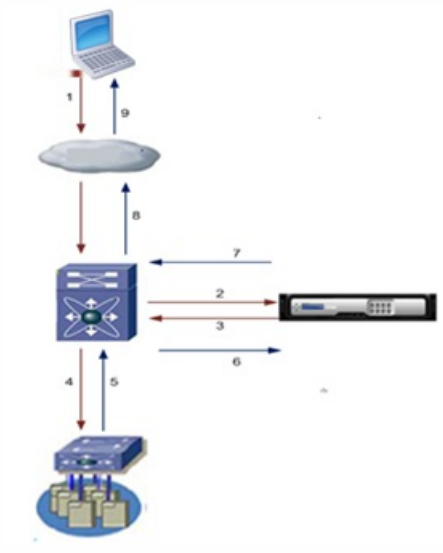
Use Case: Configuring Auto Policy-Based Routing

May 27, 2014

Auto Policy-Based Routing (APBR) automatically routes the return traffic from the servers to the NetScaler ADC, preserving the client IP addresses. The automatic policy based routes are defined on the Cisco Nexus 7000 series switch. When the return traffic from the server reaches the Cisco Nexus 7000 series switch, the APBR policies defined on the switch route the traffic to the NetScaler ADC, which in turn routes the traffic to the client.

To understand the need for APBR, first consider a NAT based scenario in which a packet flows from the client to the server and from the server back to the client.

Figure 1. Packet Flow



1. Client initiates the traffic to the virtual IP (VIP) address.
   SRC_IP= Client IP; DST_IP= VIP
2. The Cisco Nexus switch forwards the packet to the NetScaler ADC.
   SRC_IP= Client IP; DST_IP= VIP
3. The ADC performs source NAT and destination NAT (Network Address Translation), changes the source IP and destination IP addresses, and sends the packet to the Cisco Nexus switch.
   SRC_IP= NAT_IP; DST_IP= RS_IP
4. The Cisco Nexus switch receives the packet and forwards it to a server.
   SRC_IP= NAT_IP; DST_IP= RS_IP
5. The server processes the packet and forwards it to the Cisco Nexus 7000 series switch.
   SRC_IP= RS_IP IP; DST_IP= NAT_IP
6. The Cisco Nexus switch forwards the packet to the NetScaler ADC.
   SRC_IP= RS_IP IP; DST_IP= NAT_IP
7. The NetScaler ADC changes the source IP address and forwards the packet to the Cisco Nexus 7000 series switch.
   SRC_IP= VIP; DST_IP= Client_IP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

8. The Cisco Nexus 7000 series switch forwards the packet to the client.
   SRC_IP= VIP; DST_IP= Client_IP

   The client receives the packet. However, the client IP address is not visible to the server.

Now, consider a scenario in which policy based routing (PBR) directs packet flow.

1. Client initiates the traffic to the virtual IP (VIP) address.
   SRC_IP= Client IP; DST_IP= VIP

2. The Cisco Nexus switch forwards the packet to the NetScaler ADC.
   SRC_IP= Client IP; DST_IP= VIP

3. The ADC performs destination NAT (Network Address Translation), changes the destination IP, and then sends the packet to the Cisco Nexus switch.
   SRC_IP= Client IP; DST_IP= RS_IP

4. The Cisco Nexus switch receives the packet and forwards it to a server.
   SRC_IP= Client IP; DST_IP= RS_IP

5. The server processes the packet and forwards it to the Cisco Nexus 7000 series switch.
   SRC_IP= RS_IP IP; DST_IP= Client IP

6. The Cisco Nexus switch forwards the packet to the NetScaler ADC.
   SRC_IP= RS_IP IP; DST_IP= Client IP

7. The NetScaler ADC changes the source IP address and forwards the packet to the Cisco Nexus 7000 series switch.
   SRC_IP= VIP; DST_IP= Client_IP

8. The Cisco Nexus 7000 series switch forwards the packet to the client.
   SRC_IP= VIP; DST_IP= Client_IP

9. The client receives the packet. The client IP address is visible to the server. However, PBR requires manual and complex configurations and is prone to errors.

To overcome these drawbacks, configure APBR rules on the RISE appliance. When APBR is configured, the packets flow as described in the following procedure:

1. Client initiates the traffic to the virtual IP (VIP) address.
   SRC_IP= Client IP; DST_IP= VIP

2. The Cisco Nexus switch forwards the packet to the NetScaler ADC.
   SRC_IP= Client IP; DST_IP= VIP

3. The ADC performs load balancing and changes the destination IP address to the appropriate server IP address and forwards the packet to the Cisco Nexus switch in an APBR message.
   SRC_IP= Client IP; DST_IP= RS_IP

4. The Cisco Nexus switch receives the packet and forwards it to a server by using a route map.
   SRC_IP= Client IP; DST_IP= RS_IP

5. The server processes the packet and forwards it to the Cisco Nexus 7000 series switch.
   SRC_IP= RS_IP IP; DST_IP= Client_IP

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

6. When the packet reaches the Nexus switch, the switch applies the APBR rules, sets the next hop IP address to that of the NetScaler ADC, and forwards the packet to the NetScaler ADC.
   SRC_IP= RS_IP IP; DST_IP= Client_IP

7. The NetScaler ADC changes the source IP address and forwards the packet to the Cisco Nexus 7000 series switch.
   SRC_IP= VIP; DST_IP= Client_IP

8. The Cisco Nexus 7000 series switch forwards the packet to the client.
   SRC_IP= VIP; DST_IP= Client_IP

9. The client receives the packet successfully.

Note: APBR rules are configured on the Cisco Nexus switch by the Citrix Netscaler appliance only if the Use Source IP (USIP) option is enabled in the services or service groups on the Citrix Netscaler appliance.
The APBR message control flow is explained below

1. After USIP is enabled in the services on Netscaler ADC, it publishes the IP address, port number and protocol details of the server to the Cisco Nexus 7000 series switch over the RISE control channel.
2. Using the IP address, port number and protocol details of the server, the Cisco Nexus 7000 series switch creates an APBR rule which consists of ACLs and route maps.
   Note:
   - For local servers, the switch creates ACLs and route maps.
   - For remote servers , the switch forwards the APBR messages to other Cisco Nexus 7000 series switches.
3. The RISE appliance then applies the APBR rules to the switch virtual interface on the Cisco Nexus 7000 series switch connected to server.

To configure the APBR functionality:

- Enable the feature on the Cisco Nexus switch
- Configure APBR on NetScaler ADC
  - Configure NSIP
  - Configure NSVLAN
  - Enable USIP option

For more information, see Configuring Auto Policy-Based Routing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Global Server Load Balancing (GSLB) Powered Zone Preference

May 11, 2016

In a distributed XenApp/XenDesktop deployment, StoreFront might not select an optimal datacenter when multiple equivalent resources are available from multiple datacenters. In such cases, StoreFront randomly selects a datacenter. It can send the request to any of the XenApp/XenDesktop servers in any datacenter, regardless of proximity to the client making the request.

With this enhancement, the client IP address is examined when an HTTP request arrives at the NetScaler Gateway appliance, and the real client IP address is used to create the datacenter preference list that is forwarded to StoreFront. If the NetScaler appliance is configured to insert the zone preference header, StoreFront 3.5 or later can use the information provided by the appliance to reorder the list of delivery controllers and connect to an optimal delivery controller in the same zone as the client. StoreFront selects the optimal gateway VPN virtual server for the selected datacenter zone, adds this information to the ICA file with appropriate IP addresses, and sends it to the client. Storefront then tries to launch applications hosted on the preferred datacenter's delivery controllers before trying to contact equivalent controllers in other datacenters.

For more information about configuring this solution, click here.

For a video overview about GSLB powered zone preference solution, click https://www.youtube.com/watch?v=Y8DELum0Xp0.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Getting Started with Citrix NetScaler

Jul 05, 2017

Intended for system and network administrators who install and configure complex networking equipment, this section of the library describes initial set-up and basic configuration of the NetScaler, including the following topics.

- Understanding the NetScaler
- Processing Order of Features
- Where Does a NetScaler Appliance Fit in the Network?
- How a NetScaler Communicates with Clients and Servers
- Introduction to the Citrix NetScaler Product Line
- Installing the NetScaler Hardware
- Accessing a Citrix NetScaler
- Configuring a NetScaler for the First Time
- Configuring a High Availability Pair for the First Time
- Configuring a FIPS Appliance for the First Time
- Understanding Common Network Topologies
- Configuring System Management Settings
- Load Balancing Traffic on a NetScaler Appliance
- Accelerating Load Balanced Traffic by Using Compression
- Securing Load Balanced Traffic by Using SSL
- Features at a Glance

Understanding the NetScaler

The Citrix NetScaler product is an application switch that performs application-specific traffic analysis to intelligently distribute, optimize, and secure Layer 4-Layer 7 (L4–L7) network traffic for web applications. For example, a NetScaler bases load balancing decisions on individual HTTP requests instead of on long-lived TCP connections, so that the failure or slowdown of a server is managed much more quickly and with less disruption to clients. The NetScaler feature set can be broadly categorized as consisting of switching features, security and protection features, and server-farm optimization features.

Switching Features

When deployed in front of application servers, a NetScaler ensures optimal distribution of traffic by the way in which it directs client requests. Administrators can segment application traffic according to information in the body of an HTTP or TCP request, and on the basis of L4–L7 header information such as URL, application data type, or cookie. Numerous load balancing algorithms and extensive server health checks improve application availability by ensuring that client requests are directed to the appropriate servers.

Security and Protection Features

NetScaler security and protection features protect web applications from Application Layer attacks. A NetScaler allows legitimate client requests and can block malicious requests. It provides built-in defenses against denial-of-service (DoS) attacks and supports features that protect against legitimate surges in application traffic that would otherwise overwhelm the servers. An available built-in firewall protects web applications from Application Layer attacks, including buffer overflow exploits, SQL injection attempts, cross-site scripting attacks, and more. In addition, the firewall provides identity theft

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

protection by securing confidential corporate information and sensitive customer data.

Optimization Features

Optimization features offload resource-intensive operations, such as Secure Sockets Layer (SSL) processing, data compression, client keep-alive, TCP buffering, and the caching of static and dynamic content from servers. This improves the performance of the servers in the server farm and therefore speeds up applications. A NetScaler supports several transparent TCP optimizations, which mitigate problems caused by high latency and congested network links, accelerating the delivery of applications while requiring no configuration changes to clients or servers.

Understanding Policies and Expressions

A policy defines specific details of traffic filtering and management on a NetScaler. It consists of two parts: the expression and the action. The expression defines the types of requests that the policy matches. The action tells the NetScaler what to do when a request matches the expression. As an example, the expression might be to match a specific URL pattern to a type of security attack, with the action being to drop or reset the connection. Each policy has a priority, and the priorities determine the order in which the policies are evaluated.

When a NetScaler receives traffic, the appropriate policy list determines how to process the traffic. Each policy on the list contains one or more expressions, which together define the criteria that a connection must meet to match the policy.

For all policy types except Rewrite policies, a NetScaler implements only the first policy that a request matches, not any additional policies that it might also match. For Rewrite policies, the NetScaler evaluates the policies in order and, in the case of multiple matches, performs the associated actions in that order. Policy priority is important for getting the results you want.

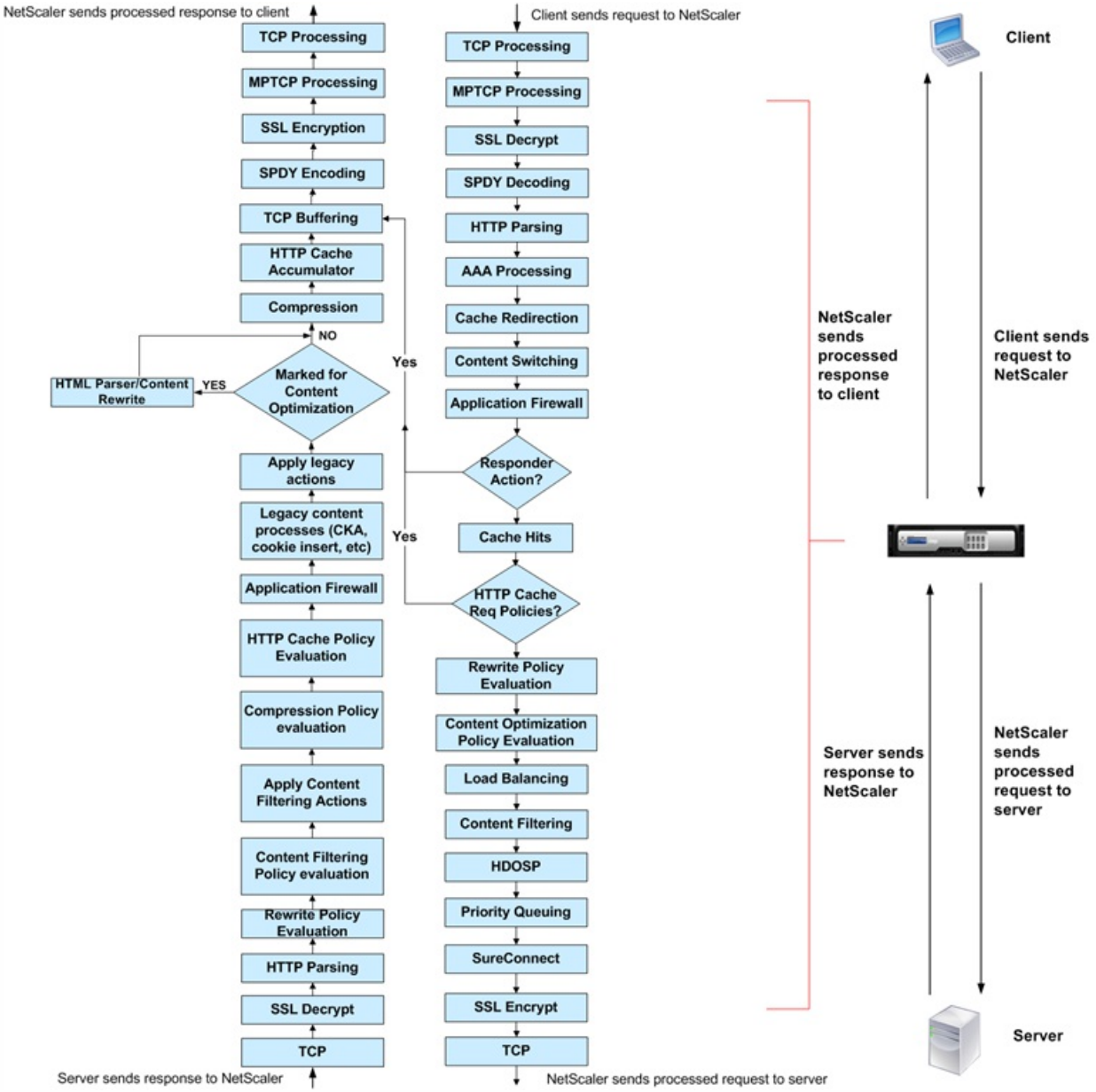
Processing Order of Features

Depending on requirements, you can choose to configure multiple features. For example, you might choose to configure both compression and SSL offload. As a result, an outgoing packet might be compressed and then encrypted before being sent to the client.

The following figure shows the L7 packet flow in the NetScaler.

Figure 1. L7 Packet Flow Diagram

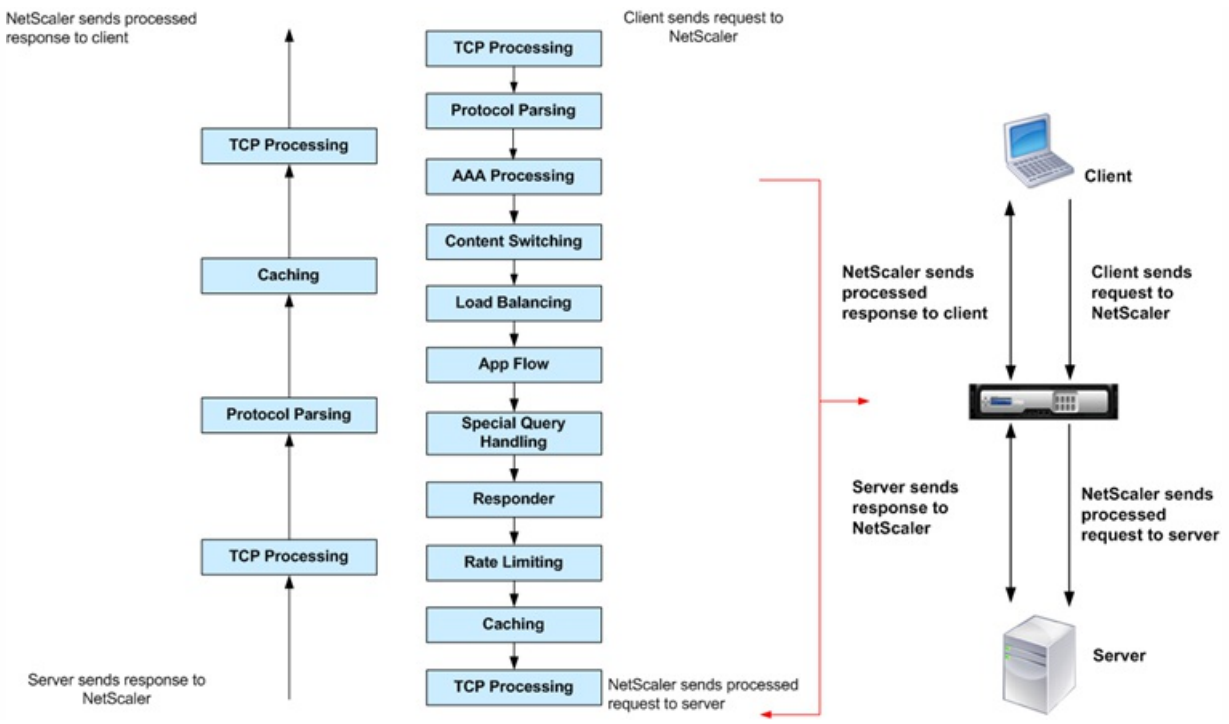
© 1999-2017 Citrix Systems, Inc. All rights reserved.



The following figure shows the DataStream packet flow in the NetScaler. DataStream is supported for MySQL and MS SQL databases. For information about the DataStream feature, see [DataStream].

Figure 2. DataStream Packet Flow Diagram

© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Where Does a NetScaler Appliance Fit in the Network?

Jun 24, 2013

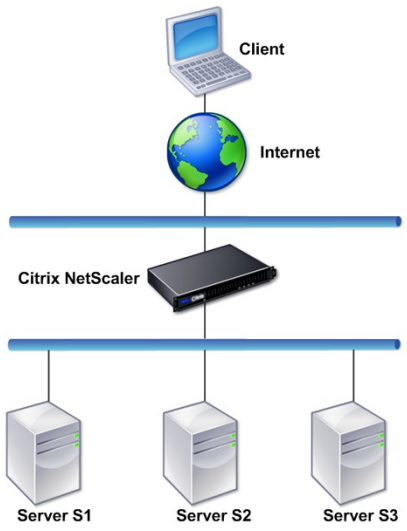
A NetScaler appliance resides between the clients and the servers, so that client requests and server responses pass through it. In a typical installation, virtual servers configured on the appliance provide connection points that clients use to access the applications behind the appliance. In this case, the appliance owns public IP addresses that are associated with its virtual servers, while the real servers are isolated in a private network. It is also possible to operate the appliance in a transparent mode as an L2 bridge or L3 router, or even to combine aspects of these and other modes.

Physical Deployment Modes

Updated: 2013-09-04

A NetScaler appliance logically residing between clients and servers can be deployed in either of two physical modes: inline and one-arm. In inline mode, multiple network interfaces are connected to different Ethernet segments, and the appliance is placed between the clients and the servers. The appliance has a separate network interface to each client network and a separate network interface to each server network. The appliance and the servers can exist on different subnets in this configuration. It is possible for the servers to be in a public network and the clients to directly access the servers through the appliance, with the appliance transparently applying the L4-L7 features. Usually, virtual servers (described later) are configured to provide an abstraction of the real servers. The following figure shows a typical inline deployment.

Figure 1. Inline Deployment



In one-arm mode, only one network interface of the appliance is connected to an Ethernet segment. The appliance in this case does not isolate the client and server sides of the network, but provides access to applications through configured virtual servers. One-arm mode can simplify network changes needed for NetScaler installation in some environments.

For examples of inline (two-arm) and one-arm deployment, see "Understanding Common Network Topologies."

Citrix NetScaler as an L2 Device

Updated: 2013-09-04

© 1999-2017 Citrix Systems, Inc. All rights reserved.

A NetScaler functioning as an L2 device is said to operate in L2 mode. In L2 mode, the NetScaler forwards packets between network interfaces when all of the following conditions are met:

- The packets are destined to another device's media access control (MAC) address.
- The destination MAC address is on a different network interface.
- The network interface is a member of the same virtual LAN (VLAN).

By default, all network interfaces are members of a pre-defined VLAN, VLAN 1. Address Resolution Protocol (ARP) requests and responses are forwarded to all network interfaces that are members of the same VLAN. To avoid bridging loops, L2 mode must be disabled if another L2 device is working in parallel with the NetScaler.

For information about how the L2 and L3 modes interact, see "Configuring Modes of Packet Forwarding."

For information about configuring L2 mode, see "Enabling and Disabling Layer 2 Mode."

Citrix NetScaler as a Packet Forwarding Device

Updated: 2014-03-14

A NetScaler appliance can function as a packet forwarding device, and this mode of operation is called L3 mode. With L3 mode enabled, the appliance forwards any received unicast packets that are destined for an IP address that does not belong to the appliance, if there is a route to the destination. The appliance can also route packets between VLANs.

In both modes of operation, L2 and L3, the appliance generally drops packets that are in:

- Multicast frames
- Unknown protocol frames destined for an appliance's MAC address (non-IP and non-ARP)
- Spanning Tree protocol (unless BridgeBPDUs is ON)

For information about how the L2 and L3 modes interact, see "Configuring Modes of Packet Forwarding."

For information about configuring the L3 mode, see "Enabling and Disabling Layer 3 Mode."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How a NetScaler Communicates with Clients and Servers

Nov 10, 2016

A NetScaler appliance is usually deployed in front of a server farm and functions as a transparent TCP proxy between clients and servers, without requiring any client-side configuration. This basic mode of operation is called Request Switching technology and is the core of NetScaler functionality. Request Switching enables an appliance to multiplex and offload the TCP connections, maintain persistent connections, and manage traffic at the request (application layer) level. This is possible because the appliance can separate the HTTP request from the TCP connection on which the request is delivered.

Depending on the configuration, an appliance might process the traffic before forwarding the request to a server. For example, if the client attempts to access a secure application on the server, the appliance might perform the necessary SSL processing before sending traffic to the server.

To facilitate efficient and secure access to server resources, an appliance uses a set of IP addresses collectively known as NetScaler-owned IP addresses. To manage your network traffic, you assign NetScaler-owned IP addresses to virtual entities that become the building blocks of your configuration. For example, to configure load balancing, you create virtual servers to receive client requests and distribute them to services, which are entities representing the applications on your servers.

Understanding NetScaler-Owned IP Addresses

To function as a proxy, a NetScaler appliance uses a variety of IP addresses. The key NetScaler-owned IP addresses are:

NetScaler IP (NSIP) address

The NSIP address is the IP address for management and general system access to the appliance itself, and for communication between appliances in a high availability configuration.

Virtual server IP (VIP) address

A VIP address is the IP address associated with a virtual server. It is the public IP address to which clients connect. An appliance managing a wide range of traffic may have many VIPs configured.

Subnet IP (SNIP) address

A SNIP address is used in connection management and server monitoring. You can specify multiple SNIP addresses for each subnet. SNIP addresses can be bound to a VLAN.

IP Set

An IP set is a set of IP addresses, which are configured on the appliance as SNIP . An IP set is identified with a meaningful name that helps in identifying the usage of the IP addresses contained in it.

Net Profile

A net profile (or network profile) contains an IP address or an IP set. A net profile can be bound to load balancing or content switching virtual servers, services, service groups, or monitors. During communication with physical servers or peers, the appliance uses the addresses specified in the profile as source IP addresses.

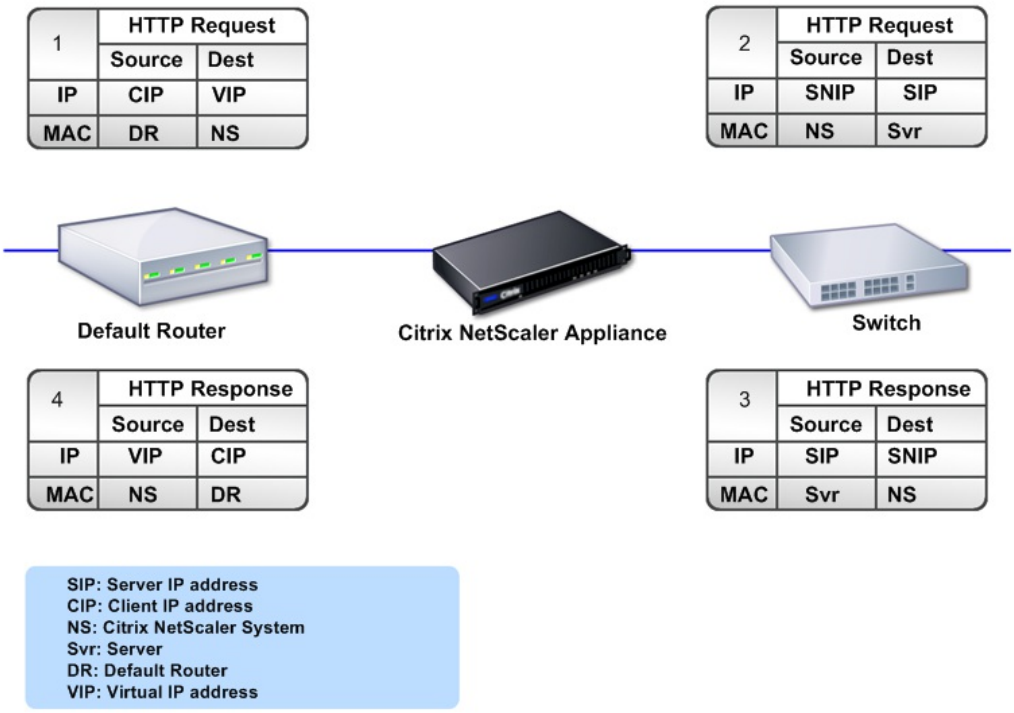
How Traffic Flows Are Managed

Because a NetScaler appliance functions as a TCP proxy, it translates IP addresses before sending packets to a server. When you configure a virtual server, clients connect to a VIP address on the NetScaler instead of directly connecting to a

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

server. As determined by the settings on the virtual server, the appliance selects an appropriate server and sends the client's request to that server. By default, the appliance uses a SNIP address to establish connections with the server, as shown in the following figure.

Figure 1. Virtual Server Based Connections



In the absence of a virtual server, when an appliance receives a request, it transparently forwards the request to the server. This is called the transparent mode of operation. When operating in transparent mode, an appliance translates the source IP addresses of incoming client requests to the SNIP address but does not change the destination IP address. For this mode to work, L2 or L3 mode has to be configured appropriately.

For cases in which the servers need the actual client IP address, the appliance can be configured to modify the HTTP header by inserting the client IP address as an additional field, or configured to use the client IP address instead of a SNIP address for connections to the servers.

Traffic Management Building Blocks

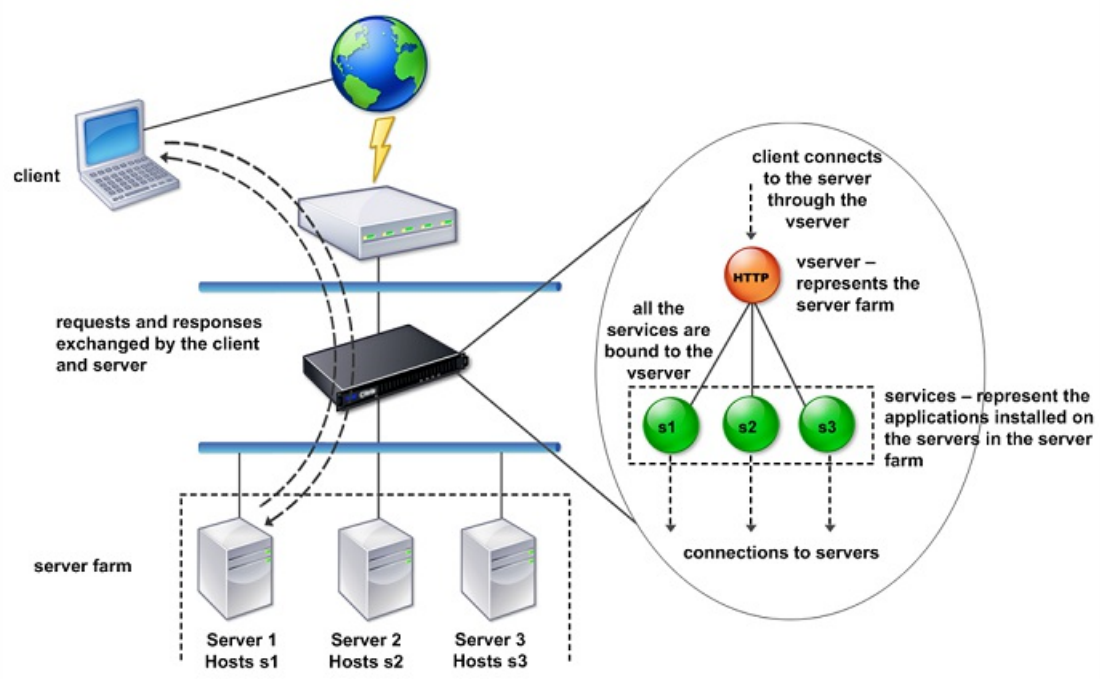
The configuration of a NetScaler appliance is typically built up with a series of virtual entities that serve as building blocks for traffic management. The building block approach helps separate traffic flows. Virtual entities are abstractions, typically representing IP addresses, ports, and protocol handlers for processing traffic. Clients access applications and resources through these virtual entities. The most commonly used entities are virtual servers and services. Virtual servers represent

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

groups of servers in a server farm or remote network, and services represent specific applications on each server.

Most features and traffic settings are enabled through virtual entities. For example, you can configure an appliance to compress all server responses to a client that is connected to the server farm through a particular virtual server. To configure the appliance for a particular environment, you need to identify the appropriate features and then choose the right mix of virtual entities to deliver them. Most features are delivered through a cascade of virtual entities that are bound to each other. In this case, the virtual entities are like blocks being assembled into the final structure of a delivered application. You can add, remove, modify, bind, enable, and disable the virtual entities to configure the features. The following figure shows the concepts covered in this section.

Figure 2. How Traffic Management Building Blocks Work



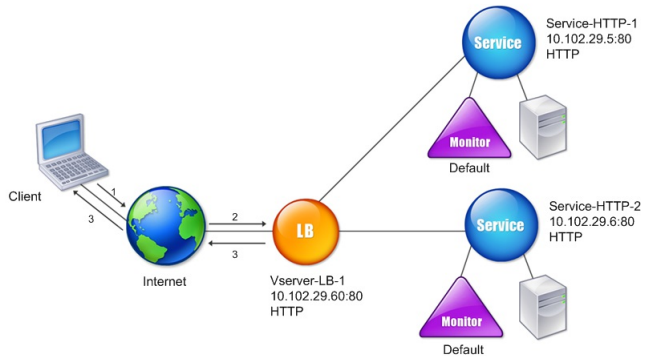
A Simple Load Balancing Configuration

In the example shown in the following figure, the NetScaler appliance is configured to function as a load balancer. For this configuration, you need to configure virtual entities specific to load balancing and bind them in a specific order. As a load balancer, an appliance distributes client requests across several servers and thus optimizes the utilization of resources.

The basic building blocks of a typical load balancing configuration are services and load balancing virtual servers. The services represent the applications on the servers. The virtual servers abstract the servers by providing a single IP address to which the clients connect. To ensure that client requests are sent to a server, you need to bind each service to a virtual server. That is, you must create services for every server and bind the services to a virtual server. Clients use the VIP address to connect to a NetScaler appliance. When the appliance receives client requests sent to the VIP address, it sends them to a server determined by the load balancing algorithm. Load balancing uses a virtual entity called a monitor to track whether a specific configured service (server plus application) is available to receive requests.

Figure 3. Load Balancing Virtual Server, Services, and Monitors

© 1999-2017 Citrix Systems, Inc. All rights reserved.



In addition to configuring the load balancing algorithm, you can configure several parameters that affect the behavior and performance of the load balancing configuration. For example, you can configure the virtual server to maintain persistence based on source IP address. The appliance then directs all requests from any specific IP address to the same server.

Understanding Virtual Servers

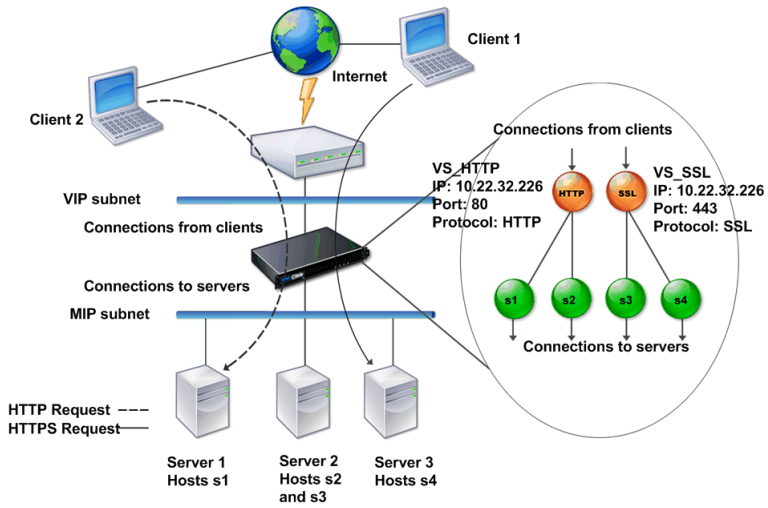
A virtual server is a named NetScaler entity that external clients can use to access applications hosted on the servers. It is represented by an alphanumeric name, virtual IP (VIP) address, port, and protocol. The name of the virtual server is of only local significance and is designed to make the virtual server easier to identify. When a client attempts to access applications on a server, it sends a request to the VIP instead of the IP address of the physical server. When the appliance receives a request at the VIP address, it terminates the connection at the virtual server and uses its own connection with the server on behalf of the client. The port and protocol settings of the virtual server determine the applications that the virtual server represents. For example, a web server can be represented by a virtual server and a service whose port and protocol are set to 80 and HTTP, respectively. Multiple virtual servers can use the same VIP address but different protocols and ports.

Virtual servers are points for delivering features. Most features, like compression, caching, and SSL offload, are normally enabled on a virtual server. When the appliance receives a request at a VIP address, it chooses the appropriate virtual server by the port on which the request was received and its protocol. The appliance then processes the request as appropriate for the features configured on the virtual server.

In most cases, virtual servers work in tandem with services. You can bind multiple services to a virtual server. These services represent the applications running on physical servers in a server farm. After the appliance processes requests received at a VIP address, it forwards them to the servers as determined by the load balancing algorithm configured on the virtual server. The following figure illustrates these concepts.

Figure 4. Multiple Virtual Servers with a Single VIP Address

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The preceding figure shows a configuration consisting of two virtual servers with a common VIP address but different ports and protocols. Each of the virtual servers has two services bound to it. The services s1 and s2 are bound to VS_HTTP and represent the HTTP applications on Server 1 and Server 2. The services s3 and s4 are bound to VS_SSL and represent the SSL applications on Server 2 and Server 3 (Server 2 provides both HTTP and SSL applications). When the appliance receives an HTTP request at the VIP address, it processes the request as specified by the settings of VS_HTTP and sends it to either Server 1 or Server 2. Similarly, when the appliance receives an HTTPS request at the VIP address, it processes it as specified by the settings of VS_SSL and it sends it to either Server 2 or Server 3.

Virtual servers are not always represented by specific IP addresses, port numbers, or protocols. They can be represented by wildcards, in which case they are known as wildcard virtual servers. For example, when you configure a virtual server with a wildcard instead of a VIP, but with a specific port number, the appliance intercepts and processes all traffic conforming to that protocol and destined for the predefined port. For virtual servers with wildcards instead of VIPs and port numbers, the appliance intercepts and processes all traffic conforming to the protocol.

Virtual servers can be grouped into the following categories:

Load balancing virtual server

Receives and redirects requests to an appropriate server. Choice of the appropriate server is based on which of the various load balancing methods the user configures.

Cache redirection virtual server

Redirects client requests for dynamic content to origin servers, and requests for static content to cache servers. Cache redirection virtual servers often work in conjunction with load balancing virtual servers.

Content switching virtual server

Directs traffic to a server on the basis of the content that the client has requested. For example, you can create a content switching virtual server that directs all client requests for images to a server that serves images only. Content switching virtual servers often work in conjunction with load balancing virtual servers.

Virtual private network (VPN) virtual server

Decrypts tunneled traffic and sends it to intranet applications.

SSL virtual server

Receives and decrypts SSL traffic, and then redirects to an appropriate server. Choosing the appropriate server is similar to

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

choosing a load balancing virtual server.

Understanding Services

Services represent applications on a server. While services are normally combined with virtual servers, in the absence of a virtual server, a service can still manage application-specific traffic. For example, you can create an HTTP service on a NetScaler appliance to represent a web server application. When the client attempts to access a web site hosted on the web server, the appliance intercepts the HTTP requests and creates a transparent connection with the web server.

In service-only mode, an appliance functions as a proxy. It terminates client connections, uses a SNIP address to establish a connection to the server, and translates the source IP addresses of incoming client requests to a SNIP address. Although the clients send requests directly to the IP address of the server, the server sees them as coming from the SNIP address. The appliance translates the IP addresses, port numbers, and sequence numbers.

A service is also a point for applying features. Consider the example of SSL acceleration. To use this feature, you must create an SSL service and bind an SSL certificate to the service. When the appliance receives an HTTPS request, it decrypts the traffic and sends it, in clear text, to the server. Only a limited set of features can be configured in the service-only case.

Services use entities called monitors to track the health of applications. Every service has a default monitor, which is based on the service type, bound to it. As specified by the settings configured on the monitor, the appliance sends probes to the application at regular intervals to determine its state. If the probes fail, the appliance marks the service as down. In such cases, the appliance responds to client requests with an appropriate error message or re-routes the request as determined by the configured load balancing policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Introduction to the Citrix NetScaler Product Line

Feb 12, 2018

The Citrix NetScaler product line optimizes delivery of applications over the internet and private networks, combining application-level security, optimization, and traffic management into a single, integrated appliance. You can install a NetScaler appliance in your server room and route all connections to your managed servers through it. The NetScaler features that you enable and the policies you set are then applied to incoming and outgoing traffic.

A NetScaler appliance can be integrated into any network as a complement to existing load balancers, servers, caches, and firewalls. It requires no additional client or server side software, and can be configured using the NetScaler web-based GUI and CLI configuration utilities.

This topic includes the following sections:

- Citrix NetScaler Hardware Platforms
- Citrix NetScaler Editions
- Supported Releases on NetScaler Hardware
- Supported Browsers

Citrix NetScaler Hardware Platforms

NetScaler hardware is available in a variety of platforms that have a range of hardware specifications:

Citrix NetScaler MPX harware platform

Citrix NetScaler SDX hardware platform

Citrix NetScaler Editions

The NetScaler operating system is available in three editions:

- Standard
- Enterprise
- Platinum

The Enterprise and Standard editions have limited features available. Feature licenses are required for all editions.

For more information about NetScaler software editions, see the NetScaler Editions datasheet.

For information about how to obtain and install licenses, see Licensing.

Supported Releases on NetScaler Hardware

See the following compatibility matrix tables for all NetScaler hardware platforms and the software releases supported on these platforms:

NetScaler MPX Hardware-Software Compatibility Matrix

NetScaler SDX Hardware-Software Compatibility Matrix

Supported Browsers

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To access the NetScaler GUI, your workstation must have a supported web browser.

Operating System	Browser	Versions
Windows 7	Internet Explorer	9, 10, and 11
	Mozilla Firefox	3.6.25 and above
	Google Chrome	Latest
Windows 64 bit	Internet Explorer	8, 9, 10, and 11
	Google Chrome	Latest
MAC	Mozilla Firefox	3.6.25 and above
	Safari	5.1.3 and above
	Google Chrome	Latest

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing the NetScaler Hardware

Jul 27, 2016

Before installing a NetScaler appliance, review the pre-installation checklist.

To use the SDX appliance, you must complete the following tasks by following the instructions given in the resources provided in table. Complete the tasks in the sequence given.

Task	Description
1. Read safety, cautions, warnings, and other information	Read the caution and danger information you need to know, before installing the product.
2. Prepare for installation	Unpack your appliance and ensure all parts were delivered, prepare the site and rack, and follow basic electrical safety precautions before you install your new appliance.
3. Install the hardware	Rack mount the appliance, install transceivers (if available), and connect the appliance to the network and a power source.
4. Configure the appliance	Configure the initial settings of the SDX appliance by using the SDX GUI or the serial console.

Follow the steps given in the following documentations to complete these tasks:

- MPX hardware documentation
- SWG hardware documentation
- SDX hardware documentation

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Accessing a Citrix NetScaler

Dec 28, 2016

A NetScaler appliance has both a command line interface (CLI) and a graphical user interface (GUI). The GUI includes a configuration utility for configuring the appliance and a statistical utility, called Dashboard. For initial access, all appliances ship with the default NetScaler IP address (NSIP) of 192.168.100.1 and default subnet mask of 255.255.0.0. You can assign a new NSIP and an associated subnet mask during initial configuration.

If you encounter an IP address conflict when deploying multiple NetScaler units, check for the following possible causes:

- Did you select an NSIP that is an IP address already assigned to another device on your network?
- Did you assign the same NSIP to multiple NetScaler appliances?
- The NSIP is reachable on all physical ports. The ports on a NetScaler are host ports, not switch ports.

The following table summarizes the available access methods.

Table 1. Methods for Accessing a NetScaler appliance

Access Method	Port	Default IP Address Required? (Y/N)
CLI	Console	N
CLI and GUI	Ethernet	Y

Using the Command Line Interface

Updated: 2013-09-04

You can access the CLI either locally, by connecting a workstation to the console port, or remotely, by connecting through secure shell (SSH) from any workstation on the same network.

Logging on to the Command Line Interface through the Console Port

The appliance has a console port for connecting to a computer workstation. To log on to the appliance, you need a serial crossover cable and a workstation with a terminal emulation program.
To log on to the CLI through the console port

1. Connect the console port to a serial port on the workstation, as described in .
2. On the workstation, start HyperTerminal or any other terminal emulation program. If the logon prompt does not appear, you may need to press ENTER one or more times to display it.
3. Log on by using the administrator credentials. The command prompt (>) appears on the workstation monitor.

Logging on to the Command Line Interface by using SSH

The SSH protocol is the preferred remote access method for accessing an appliance remotely from any workstation on the same network. You can use either SSH version 1 (SSH1) or SSH version 2 (SSH2.)

If you do not have a working SSH client, you can download and install any of the following SSH client programs:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- PuTTY
  Open Source software supported on multiple platforms. Available at:

  "http://www.chiark.greenend.org.uk/~sgtatham/putty/"

- Vandyke Software SecureCRT
  Commercial software supported on the Windows platform. Available at:

  "http://www.vandyke.com/products/securecrt/"

These programs have been tested by the Citrix NetScaler team, which has verified that they work correctly with a NetScaler appliance. Other programs may also work correctly, but have not been tested.

To verify that the SSH client is installed properly, use it to connect to any device on your network that accepts SSH connections.

To log on to a NetScaler by using an SSH client

1. On your workstation, start the SSH client.
2. For initial configuration, use the default NetScaler IP address (NSIP), which is 192.168.100.1. For subsequent access, use the NSIP that was assigned during initial configuration. Select either SSH1 or SSH2 as the protocol.
3. Log on by using the administrator credentials. For example:

```
login as: nsroot
Using keyboard-interactive authentication.
Password:
Last login: Tue Jun 16 10:37:28 2009 from 10.102.29.9

 Done
>
```

Using the Graphical User Interface

Updated: 2014-06-30

Important: A certificate-key pair is required for HTTPS access to the NetScaler configuration utility. On a NetScaler ADC, a certificate-key pair is automatically bound to the internal services. On an MPX or SDX appliance, the default key size is 1024 bytes, and on a VPX instance, the default key size is 512 bytes. However, most browsers today do not accept a key that is less than 1024 bytes. As a result, HTTPS access to the VPX configuration utility is blocked.
Additionally, if a license is not present on an MPX appliance when it starts, and you add a license later and restart the appliance, you might lose the certificate binding.

Citrix recommends that you install a certificate-key pair of at least 1024 bytes on a NetScaler ADC for HTTPS access to the configuration utility, and that you install an appropriate license before starting the ADC.

The graphical user interface includes a configuration utility and a statistical utility, called Dashboard, either of which you access through a workstation connected to an Ethernet port on the appliance.

The system requirements for the workstation running the GUI are as follows:

- For Windows-based workstations, a Pentium 166 MHz or faster processor.
- For Linux-based workstations, a Pentium platform running Linux kernel v2.2.12 or above, and glibc version 2.12-11 or later.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

A minimum of 32 MB RAM is required, and 48 MB RAM is recommended. The workstation should support 16-bit color mode, KDE and KWM window managers used in conjunction, with displays set to local hosts.

- For Solaris-based workstations, a Sun running either Solaris 2.6, Solaris 7, or Solaris 8.

Your workstation must have a supported web browser to access the configuration utility and Dashboard.

The following browsers are supported.

Operating System	Browser	Versions
Windows 7	Internet Explorer	9, 10, and 11
	Mozilla Firefox	3.6.25 and above
	Google Chrome	Latest
Windows 64 bit	Internet Explorer	8, 9, 10, and 11
	Google Chrome	Latest
MAC	Mozilla Firefox	3.6.25 and above
	Safari	5.1.3 and above
	Google Chrome	Latest

Using the Configuration Utility

Once you log on to the configuration utility, you can configure the appliance through a graphical interface that includes context-sensitive help.

To log on to the configuration utility

1. Open your web browser and enter the NetScaler IP (NSIP) as an HTTP address. If you have not yet set up the initial configuration, enter the default NSIP (http://192.168.100.1). The Citrix Logon page appears.
   Note: If you have two NetScaler appliances in a high availability setup, make sure that you do not access the GUI by entering the IP address of the secondary NetScaler. If you do so and use the GUI to configure the secondary NetScaler, your configuration changes will not be applied to the primary NetScaler.
2. In the User Name text box, type nsroot.
3. In the Password text box, type the administrative password you assigned to the nsroot account during initial configuration and click Login. The Configuration Utility page appears.
   If you need to access the online help, select Help from the Help menu at the top right corner.

Using the Statistical Utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Dashboard, the statistical utility, is a browser-based application that displays charts and tables on which you can monitor the performance of a NetScaler.

To log on to Dashboard

1. Open your web browser and enter the NSIP as an HTTP address (http://<NSIP>). The Citrix Logon page appears.
2. In the User Name text box, type nsroot.
3. In the Password text box, type the administrative password you assigned to the nsroot account during initial configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a NetScaler for the First Time

Aug 22, 2013

Initial configuration is the same for the multifunction Citrix NetScaler, the dedicated NetScaler Gateway Enterprise Edition, and the dedicated Citrix NetScaler Application Firewall appliances. You can use any of the following interfaces for initial configuration of your appliance:

- First-time use wizard—If you use a web browser to connect to the appliance, you are prompted to enter the network configuration and licensing information, if it is not already specified.
- LCD keypad—You can specify the network settings, but you must use a different interface to upload your licenses.
- Serial console—After connecting to the serial console, you can use the command line interface to specify the network settings and upload your licenses.
- NITRO API—You can use the NITRO API suite to configure the NetScaler appliance.

For initial configuration, use nsroot as both the administrative user name and the password. For subsequent access, use the password assigned during initial configuration.

If you are setting up two NetScaler appliances as a high availability pair, you configure one as primary and the other as secondary.

The configuration procedure for a FIPS appliance is slightly different from the procedure for a NetScaler MPX appliance or a NetScaler virtual appliance.

Using the First-time Setup Wizard

To configure a NetScaler appliance (or NetScaler virtual appliance) for the first time, you need an administrative computer configured on the same network as the appliance.

You must assign a NetScaler IP (NSIP) address as the management IP address of your NetScaler appliance. This is the address at which you access the NetScaler for configuration, monitoring, and other management tasks. Assign a subnet IP (SNIP) address for your NetScaler to communicate with the backend servers. Specify a host name to identify your NetScaler, an IP address for a DNS server to resolve domain names, and the time zone in which your NetScaler is located.

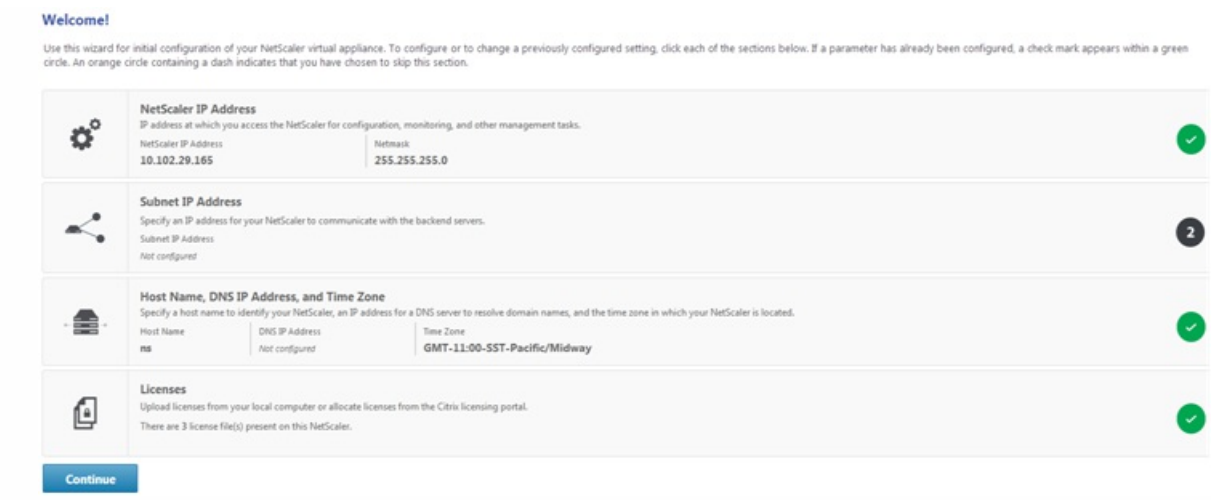
The wizard automatically appears if any of the following conditions are met:

- The appliance is configured with the default IP address (192.168.100.1).
- A subnet IP address is not configured.
- Licenses are not present on the appliance.

To perform first-time configuration of your appliance

1. In a web browser, type: http://192.168.100.1
   Note: The NetScaler software is preconfigured with a default IP address. If you have already assigned as NSIP address, type that address in a web browser.
2. In User Name and Password, type the administrator credentials. The following screen appears.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



3. To configure or to change a previously configured setting, click inside each section. When done, click Continue.
4. When prompted, select Reboot.

Using the LCD Keypad

When you first install the appliance, you can configure the initial settings by using the LCD keypad on the front panel of the appliance. The keypad interacts with the LCD display module, which is also on the front panel of these appliances.

Note: You can use the LCD keypad for initial configuration on a new appliance with the default configuration. The configuration file (ns.conf) should contain the following command and default values.
set ns config -IPAddress 192.168.100.1 -netmask 255.255.0.0

The functions of the different keys are explained in the following table.

Table 1. LCD Key Functions

Key	Function
<	Moves the cursor one digit to the left.
>	Moves the cursor one digit to the right.
^	Increments the digit under the cursor.
v	Decrements the digit under the cursor.
.	Processes the information, or terminates the configuration, if none of the values are changed. This key is also known as the ENTER key.

To perform the initial configuration by using the LCD keypad press the "<" key.

You are prompted to enter the subnet mask, NetScaler IP address (NSIP), and gateway in that order respectively. The subnet mask is associated with both the NSIP and default gateway IP address. The NSIP is the IPv4 address of the NetScaler appliance. The default gateway is the IPv4 address for the router, which will handle external IP traffic that the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler cannot otherwise route. The NSIP and the default gateway should be on the same subnet.

If you enter a valid value for the subnet mask, such as 255.255.255.224, you are prompted to enter the IP address. Similarly, if you enter a valid value for the IP address, you are prompted to enter the gateway address. If the value you entered is invalid, the following error message appears for three seconds, where xxx.xxx.xxx.xxx is the IP address you entered, followed by a request to re-enter the value.

Invalid addr!
xxx.xxx.xxx.xxx
If you press the ENTER (.) key without changing any of the digits, the software interprets this as a user exit request. The following message will be displayed for three seconds.

Exiting menu...
xxx.xxx.xxx.xxx
If all the values entered are valid, when you press the ENTER key, the following message appears.

Values accepted,
Rebooting...
The subnet mask, NSIP, and gateway values are saved in the configuration file.

Note: For information about deploying a high availability (HA) pair, see "High Availability."

Using the NetScaler Serial Console

When you first install the appliance, you can configure the initial settings by using the serial console. With the serial console, you can change the system IP address, create a subnet or mapped IP address, configure advanced network settings, and change the time zone.

Note: To locate the serial console port on your appliance, see "RS232 Serial Console Port" in "Ports."

To configure initial settings by using a serial console

1. Connect the console cable into your appliance. For more information, see "Connecting the Console Cable" in "Connecting the Cables."
2. Run the vt100 terminal emulation program of your choice on your computer to connect to the appliance and configure the following settings: 9600 baud, 8 data bits, 1 stop bit, parity, and flow control set to NONE.
3. Press ENTER. The terminal screen displays the Logon prompt.
   Note: You might have to press ENTER two or three times, depending on which terminal program you are using.
4. Log on to the appliance with the administrator credentials. Your sales representative or Citrix Customer Service can provide you with the administrator credentials.
5. At the prompt, type config ns to run the NetScaler configuration script.
6. To complete the initial configuration of your appliance, follow the prompts.
   Note: To prevent an attacker from breaching your ability to send packets to the appliance, choose a non-routable IP address on your organization's LAN as your appliance IP address.
   You can replace steps 5 and 6 with the following NetScaler commands. At the NetScaler command prompt, type:

   set ns config -ipaddress<IPAddress> -netmask<subnetMask>

   add ns ip<IPAddress> <subnetMask> -type<type>

   add route<network> <netmask> <gateway>

   set system user <userName> -password

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

save ns config

reboot

Example

```
set ns config -ipaddress 10.102.29.60 -netmask 255.255.255.0
add ns ip 10.102.29.61 255.255.255.0 -type snip
add route 0.0.0.0 0.0.0.0 10.102.29.1
set system user nsroot -password
Enter password: *****
Confirm password: *****
save ns config
reboot
```

You have now completed initial configuration of your appliance. To continue configuring the appliance, choose one of the following options:

Citrix NetScaler.

If you are configuring your appliance as a standard NetScaler with other licensed features, see "Load Balancing."

Citrix NetScaler Application Firewall.

If you are configuring your appliance as a standalone application firewall, see "Application Firewall."

NetScaler Gateway.

If you are configuring your appliance as an NetScaler Gateway, see "NetScaler Gateway 10.5."

Note: For information about deploying a high availability (HA) pair, see "Configuring High Availability."

Configuring a NetScaler by Using the NITRO API

You can use the NITRO API to configure the NetScaler appliance. NITRO exposes its functionality through Representational State Transfer (REST) interfaces. Therefore, NITRO applications can be developed in any programming language. Additionally, for applications that must be developed in Java or .NET or Python, NITRO APIs are exposed through relevant libraries that are packaged as separate Software Development Kits (SDKs). For more information, see NITRO API.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a High Availability Pair for the First Time

Sep 07, 2016

You can deploy two NetScaler appliances in a high availability configuration, where one unit actively accepts connections and manages servers while the secondary unit monitors the first. The NetScaler that is actively accepting connections and managing the servers is called a primary unit and the other one is called a secondary unit in a high availability configuration. If there is a failure in the primary unit, the secondary unit becomes the primary and begins actively accepting connections.

Each NetScaler in a high availability pair monitors the other by sending periodic messages, called heartbeat messages or health checks, to determine the health or state of the peer node. If a health check for a primary unit fails, the secondary unit retries the connection for a specific time period. For more information about high availability, see "High Availability." If a retry does not succeed by the end of the specified time period, the secondary unit takes over for the primary unit in a process called failover. The following figure shows two high availability configurations, one in one-arm mode and the other in two-arm mode.

Figure 1. High availability in one-arm mode

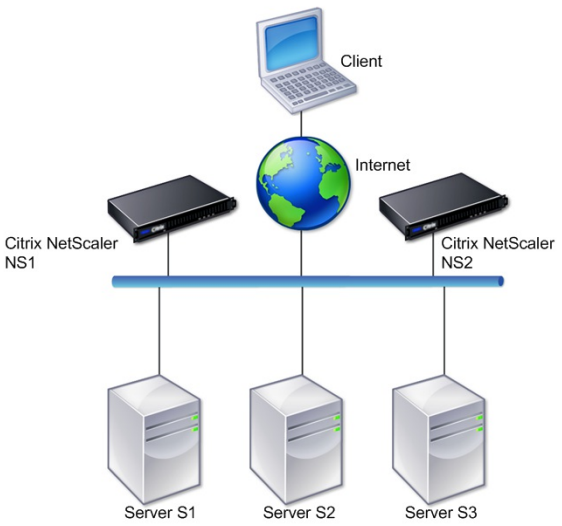
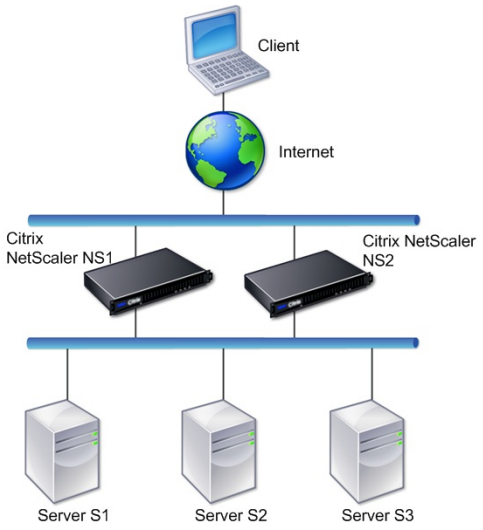


Figure 2. High availability in two-arm mode

© 1999-2017 Citrix Systems, Inc. All rights reserved.



In one-arm configuration, both NS1 and NS2 and servers S1, S2, and S3 are connected to the switch.

In two-arm configuration, both NS1 and NS2 are connected to two switches. The servers S1, S2, and S3 are connected to the second switch. The traffic between client and the servers passes through either NS1 or NS2.

To set up a high availability environment, configure one NetScaler as primary and another as secondary. Perform the following tasks on each of the NetScalers:

- Add a node.
- Disable high availability monitoring for unused interfaces.

Adding a Node

Updated: 2013-06-24

A node is a logical representation of a peer NetScaler appliance. It identifies the peer unit by ID and NSIP. An appliance uses these parameters to communicate with the peer and track its state. When you add a node, the primary and secondary units exchange heartbeat messages asynchronously. The node ID is an integer that must not be greater than 64.

To add a node by using the command line interface

At the command prompt, type the following commands to add a node and verify that the node has been added:

- add HA node <id> <IPAddress>
- show HA node <id>

  Example

  ```
  add HA node 0 10.102.29.170
   Done
  > show HA node 0
  1)    Node ID:    0
        IP:  10.102.29.200 (NS200)
  ```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Node State: UP
    Master State: Primary
    SSL Card Status: UP
    Hello Interval: 200 msecs
    Dead Interval: 3 secs
    Node in this Master State for: 1:0:41:50 (days:hrs:min:sec)
```

To add a node by using the configuration utility

1. Navigate to **System** > **High Availability**.
2. Click **Add** on the **Nodes** tab.
3. On the **Create HA Node** page, in the **Remote Node IP Address** text box, type the NSIP Address (for example, 10.102.29.170) of the remote node.
4. Ensure that the **Configure remote system to participate in High Availability setup** check box is selected. Provide the login credentials of the remote node in the text boxes under **Remote System Login Credentials**.
5. Select the **Turn off HA monitor on interfaces/channels that are down** check box to disable the HA monitor on interfaces that are down.

Verify that the node you added appears in the list of nodes in the **Nodes** tab.

Disabling High Availability Monitoring for Unused Interfaces

Updated: 2013-06-24

The high availability monitor is a virtual entity that monitors an interface. You must disable the monitor for interfaces that are not connected or being used for traffic. When the monitor is enabled on an interface whose status is DOWN, the state of the node becomes NOT UP. In a high availability configuration, a primary node entering a NOT UP state might cause a high availability failover. An interface is marked DOWN under the following conditions:

- The interface is not connected
- The interface is not working properly
- The cable connecting the interface is not working properly

To disable the high availability monitor for an unused interface by using the command line interface

At the command prompt, type the following commands to disable the high availability monitor for an unused interface and verify that it is disabled:

- set interface <id> -haMonitor OFF
- show interface <id>

  Example

```
> set interface 1/8 -haMonitor OFF
 Done
> show interface 1/8
    Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
    flags=0x4000 <ENABLED, DOWN, down, autoneg, 802.1q>
    MTU=1514, native vlan=1, MAC=00:d0:68:15:fd:3d, downtime 238h55m44s
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Requested: media AUTO, speed AUTO, duplex AUTO, fctl OFF,
    throughput 0

RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
Bandwidth thresholds are not set.
```

When the high availability monitor is disabled for an unused interface, the output of the show interface command for that interface does not include "HAMON."

To disable the high availability monitor for unused interfaces by using the configuration utility

1. Navigate to System > Network > Interfaces.
2. Select the interface for which the monitor must be disabled.
3. Click Open. The Modify Interface dialog box appears.
4. In HA Monitoring, select the OFF option.
5. Click OK.
6. Verify that, when the interface is selected, "HA Monitoring: OFF" appears in the details at the bottom of the page.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a FIPS Appliance for the First Time

Sep 16, 2013

A certificate-key pair is required for HTTPS access to the configuration utility and for secure remote procedure calls. RPC nodes are internal system entities used for system-to-system communication of configuration and session information. One RPC node exists on each appliance. This node stores the password, which is checked against the one provided by the contacting appliance. To communicate with other NetScaler appliances, each appliance requires knowledge of the other appliances, including how to authenticate on the other appliance. RPC nodes maintain this information, which includes the IP addresses of the other NetScaler appliances and the passwords used to authenticate on each.

On a NetScaler MPX appliance virtual appliance, a certificate-key pair is automatically bound to the internal services. On a FIPS appliance, a certificate-key pair must be imported into the hardware security module (HSM) of a FIPS card. To do so, you must configure the FIPS card, create a certificate-key pair, and bind it to the internal services.

To configure secure HTTPS by using the command line interface

1. Initialize the hardware security module (HSM) on the FIPS card of the appliance. For information about initializing the HSM, see "Configuring the HSM."
2. If the appliance is part of a high availability setup, enable the SIM. For information about enabling the SIM on the primary and secondary appliances, see "Configuring FIPS Appliances in a High Availability Setup."
3. Import the FIPS key into the HSM of the FIPS card of the appliance. At the command prompt, type:
   import ssl fipskey serverkey -key ns-server.key -inform PEM
4. Add a certificate-key pair. At the command prompt, type:
   add certkey server -cert ns-server.cert -fipskey serverkey
5. Bind the certificate-key created in the previous step to the following internal services. At the command prompt, type:
   bind ssl service nshttps-127.0.0.1-443 -certkeyname server

   bind ssl service nshttps-::11-443 -certkeyname server

To configure secure HTTPS by using the configuration utility

1. Initialize the hardware security module (HSM) on the FIPS card of the appliance. For information about initializing the HSM, see "Configuring the HSM."
2. If the appliance is part of a high availability setup, enable the secure information system (SIM). For information about enabling the SIM on the primary and secondary appliances, see " Configuring FIPS Appliances in a High Availability Setup."
3. Import the FIPS key into the HSM of the FIPS card of the appliance. For more information about importing a FIPS key, see "Importing an Existing FIPS Key."
4. Navigate to Traffic Management > SSL > Certificates.
5. In the details pane, click Install.
6. In the Install Certificate dialog box, type the certificate details.
7. Click Create, and then click Close.
8. Navigate to Traffic Management > Load Balancing > Services.
9. In the details pane, on the Action tab, click Internal Services.
10. Select nshttps-127.0.0.1-443 from the list, and then click Open.
11. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
12. Select nshttps-::11-443 from the list, and then click Open.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

13. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
14. Click OK.

To configure secure RPC by using the command line interface

1. Initialize the hardware security module (HSM) on the FIPS card of the appliance. For information about initializing the HSM, see "Configuring the HSM."
2. Enable the secure information system (SIM). For information about enabling the SIM on the primary and secondary appliances, see " Configuring FIPS Appliances in a High Availability Setup."
3. Import the FIPS key into the HSM of the FIPS card of the appliance. At the command prompt, type:
   import ssl fipskey serverkey -key ns-server.key -inform PEM
4. Add a certificate-key pair. At the command prompt, type:
   add certkey server -cert ns-server.cert -fipskey serverkey
5. Bind the certificate-key pair to the following internal services. At the command prompt, type:
   bind ssl service nsrpcs-127.0.0.1-3008 -certkeyname server

   bind ssl service nskrpcs-127.0.0.1-3009 -certkeyname server

   bind ssl service nsrpcs-::1l-3008 -certkeyname server
6. Enable secure RPC mode. At the command prompt, type:
   set ns rpcnode <IP address> -secure YES

To configure secure RPC by using the configuration utility

1. Initialize the hardware security module (HSM) on the FIPS card of the appliance. For information about initializing the HSM, see "Configuring the HSM."
2. Enable the secure information system (SIM). For information about enabling the SIM on the primary and secondary appliances, see " Configuring FIPS Appliances in a High Availability Setup."
3. Import the FIPS key into the HSM of the FIPS card of the appliance. For more information about importing a FIPS key, see "Importing an Existing FIPS Key."
4. Navigate to Traffic Management > SSL > Certificates.
5. In the details pane, click Install.
6. In the Install Certificate dialog box, type the certificate details.
7. Click Create, and then click Close.
8. Navigate to Traffic Management > Load Balancing > Services.
9. In the details pane, on the Action tab, click Internal Services.
10. Select nsrpcs-127.0.0.1-3008 from the list, and then click Open.
11. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
12. Select nskrpcs-127.0.0.1-3009 from the list, and then click Open.
13. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
14. Select nsrpcs-::11-3008 from the list, and then click Open.
15. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
16. Click OK.
17. Navigate to System > Network > RPC
18. In the details pane, select the IP address, and click Open.
19. In the Configure RPC Node dialog box, select Secure.
20. Click OK.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding Common Network Topologies

Oct 27, 2015

As described in "Physical Deployment Modes," you can deploy the Citrix NetScaler appliance either inline between the clients and servers or in one-arm mode. Inline mode uses a two-arm topology, which is the most common type of deployment.

This document includes the following:

- Setting Up Common Two-Arm Topologies
- Setting Up Common One-Arm Topologies

Setting Up Common Two-Arm Topologies

In a two-arm topology, one network interface is connected to the client network and another network interface is connected to the server network, ensuring that all traffic flows through the appliance. This topology might require you to reconnect your hardware and also might result in a momentary downtime. The basic variations of two-arm topology are multiple subnets, typically with the appliance on a public subnet and the servers on a private subnet, and transparent mode, with both the appliance and the servers on the public network.

Setting Up a Simple Two-Arm Multiple Subnet Topology

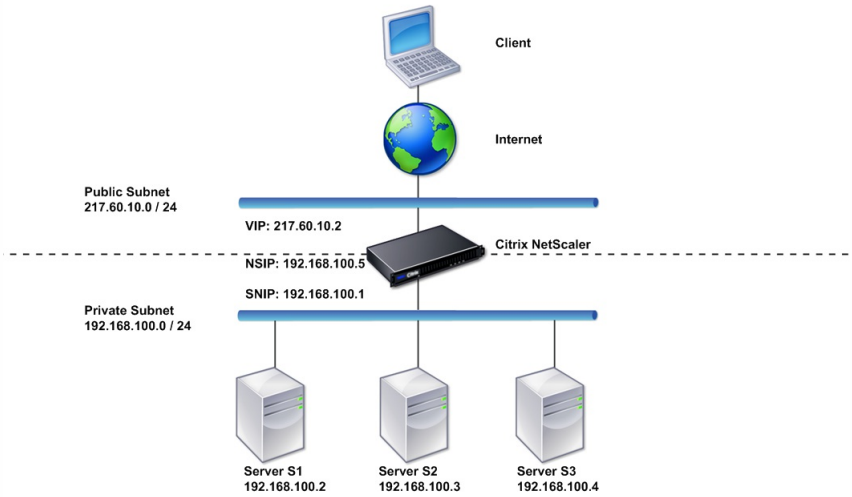
One of the most commonly used topologies has the NetScaler appliance inline between the clients and the servers, with a virtual server configured to handle the client requests. This configuration is used when the clients and servers reside on different subnets. In most cases, the clients and servers reside on public and private subnets, respectively.

For example, consider an appliance deployed in two-arm mode for managing servers S1, S2, and S3, with a virtual server of type HTTP configured on the appliance, and with HTTP services running on the servers. The servers are on a private subnet and a SNIP is configured on the appliance to communicate with the servers. The Use SNIP (USNIP) option must be enabled on the appliance so that it uses the SNIP instead of the MIP.

As shown in the following figure, the VIP is on public subnet 217.60.10.0, and the NSIP, the servers, and the SNIP are on private subnet 192.168.100.0/24.

Figure 1. Topology Diagram for Two-Arm Mode, Multiple Subnets

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Task overview: To deploy a NetScaler appliance in two-arm mode with multiple subnets

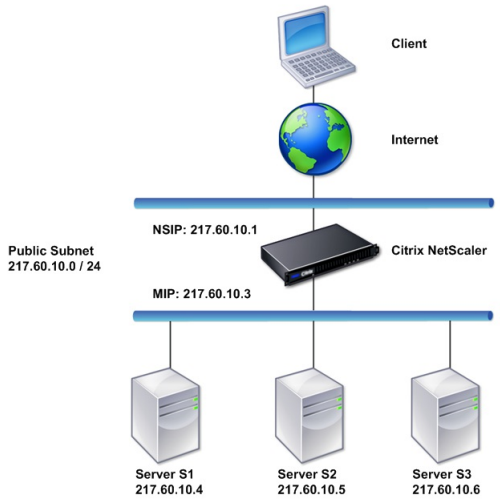
1. Configure the NSIP and default gateway, as described in "Configuring the NetScaler IP Address (NSIP)."
2. Configure the SNIP, as described in "Configuring Subnet IP Addresses."
3. Enable the USNIP option, as described in "To enable or disable USNIP mode."
4. Configure the virtual server and the services, as described in "Creating a Virtual Server" and "Configuring Services."
5. Connect one of the network interfaces to a private subnet and the other interface to a public subnet.

Setting Up a Simple Two-Arm Transparent Topology

Use transparent mode if the clients need to access the servers directly, with no intervening virtual server. The server IP addresses must be public because the clients need to be able to access them. In the example shown in the following figure, a NetScaler appliance is placed between the client and the server, so the traffic must pass through the appliance. You must enable L2 mode for bridging the packets. The NSIP and MIP are on the same public subnet, 217.60.10.0/24.

Figure 2. Topology Diagram for Two-Arm, Transparent Mode

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Task overview: To deploy a NetScaler in two-arm, transparent mode

1. Configure the NSIP, MIP, and default gateway, as described in "Configuring a NetScaler by Using the Command Line Interface."
2. Enable L2 mode, as described in "Enabling and Disabling Layer 2 Mode."
3. Configure the default gateway of the managed servers as the MIP.
4. Connect the network interfaces to the appropriate ports on the switch.

Setting Up Common One-Arm Topologies

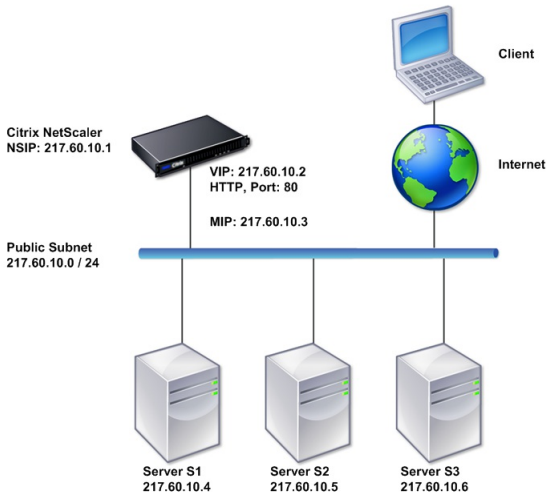
The two basic variations of one-arm topology are with a single subnet and with multiple subnets.

Setting Up a Simple One-Arm Single Subnet Topology

You can use a one-arm topology with a single subnet when the clients and servers reside on the same subnet. For example, consider a NetScaler deployed in one-arm mode for managing servers S1, S2, and S3. A virtual server of type HTTP is configured on a NetScaler, and HTTP services are running on the servers. As shown in the following figure, the NetScaler IP address (NSIP), the Mapped IP address (MIP), and the server IP addresses are on the same public subnet, 217.60.10.0/24.

Figure 3. Topology Diagram for One-Arm Mode, Single Subnet

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Task overview: To deploy a NetScaler in one-arm mode with a single subnet

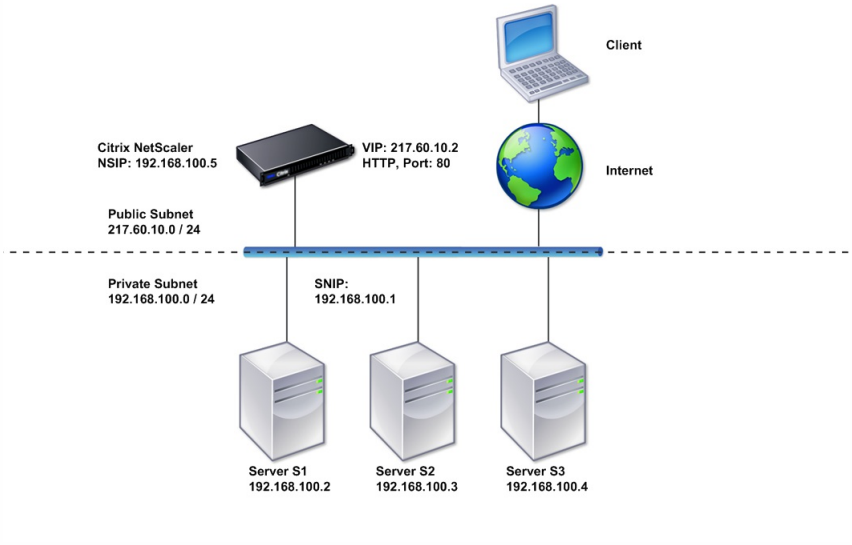
1. Configure the NSIP, MIP, and the default gateway, as described in "Configuring the NetScaler IP Address (NSIP)".
2. Configure the virtual server and the services, as described in "Creating a Virtual Server" and "Configuring Services".
3. Connect one of the network interfaces to the switch.

Setting Up a Simple One-Arm Multiple Subnet Topology

You can use a one-arm topology with multiple subnets when the clients and servers reside on the different subnets. For example, consider a NetScaler appliance deployed in one-arm mode for managing servers S1, S2, and S3, with the servers connected to switch SW1 on the network. A virtual server of type HTTP is configured on the appliance, and HTTP services are running on the servers. These three servers are on the private subnet, so a subnet IP address (SNIP) is configured to communicate with them. The Use Subnet IP address (USNIP) option must be enabled so that the appliance uses the SNIP instead of a MIP. As shown in the following figure, the virtual IP address (VIP) is on public subnet 217.60.10.0/24; the NSIP, SNIP, and the server IP addresses are on private subnet 192.168.100.0/24.

Figure 4. Topology Diagram for One-Arm Mode, Multiple Subnets

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Task overview: To deploy a NetScaler appliance in one-arm mode with multiple subnets

1. Configure the NSIP and the default gateway, as described in "Configuring the NetScaler IP Address (NSIP)".
2. Configure the SNIP and enable the USNIP option, as described in "Configuring Subnet IP Addresses".
3. Configure the virtual server and the services, as described in "Creating a Virtual Server" and "Configuring Services".
4. Connect one of the network interfaces to the switch.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring System Management Settings

Sep 04, 2013

Once your initial configuration is in place, you can configure settings to define the behavior of the Citrix NetScaler appliance and facilitate connection management. You have a number of options for handling HTTP requests and responses. Routing, bridging, and MAC based forwarding modes are available for handling packets not addressed to the NetScaler. You can define the characteristics of your network interfaces and can aggregate the interfaces. To prevent timing problems, you can synchronize the NetScaler clock with a Network Time Protocol (NTP) server. The NetScaler can operate in various DNS modes, including as an authoritative domain name server (ADNS). You can set up SNMP for system management and customize syslog logging of system events. Before deployment, verify that your configuration is complete and correct.

This document includes the following:

- Configuring System Settings
- Configuring Modes of Packet Forwarding
- Configuring Network Interfaces
- Configuring Clock Synchronization
- Configuring DNS
- Configuring SNMP
- Verifying the Configuration

Note: In addition to the tasks listed above, you can configure Syslog logging. For instructions, see "Audit Logging."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring System Settings

Sep 04, 2013

Configuration of system settings includes basic tasks such as configuring HTTP ports to enable connection keep-alive and server offload, setting the maximum number of connections for each server, and setting the maximum number of requests per connection. You can enable client IP address insertion for situations in which a proxy IP address is not suitable, and you can change the HTTP cookie version.

You can also configure a NetScaler appliance to open FTP connections on a controlled range of ports instead of ephemeral ports for data connections. This improves security, because opening all ports on the firewall is insecure. You can set the range anywhere from 1,024 to 64,000.

Before deployment, go through the verification checklists to verify your configuration. To configure HTTP parameters and the FTP port range, use the NetScaler configuration utility.

You can modify the types of HTTP parameters described in the following table.

Table 1. HTTP Parameters

Parameter Type	Specifies
HTTP Port Information	The web server HTTP ports used by your managed servers. If you specify the ports, the appliance performs request switching for any client request that has a destination port matching a specified port. Note: If an incoming client request is not destined for a service or a virtual server that is specifically configured on the appliance, the destination port in the request must match one of the globally configured HTTP ports. This allows the appliance to perform connection keep-alive and server off-load.
Limits	The maximum number of connections to each managed server, and the maximum number of requests sent over each connection. For example, if you set Max Connections to 500, and the appliance is managing three servers, it can open a maximum of 500 connections to each of the three servers. By default, the appliance can create an unlimited number of connections to any of the servers it manages. To specify an unlimited number of requests per connection, set Max Requests to 0. Note: If you are using the Apache HTTP server, you must set Max Connections equal to the value of the MaxClients parameter in the Apache httpd.conf file. Setting this parameter is optional for other web servers.
Client IP Insertion	Enable/disable insertion of the client's IP address into the HTTP request header. You can specify a name for the header field in the adjacent text box. When a web server managed by an appliance receives a mapped IP address or a subnet IP address, the server identifies it as the client’s IP address. Some applications need the client’s IP address for logging purposes or to dynamically determine the content to be served by the web server. You can enable insertion of the actual client IP address into the HTTP header request sent from the client to one, some, or all servers managed by the appliance. You can then access the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter Type	Specifies
	inserted address through a minor modification to the server (using an Apache module, ISAPI interface, or NSAPI interface).
Cookie Version	The HTTP cookie version to use when COOKIEINSERT persistence is configured on a virtual server. The default, version 0, is the most common type on the Internet. Alternatively, you can specify version 1.
Requests/Responses	Options for handling certain types of requests, and enable/disable logging of HTTP error responses.
Server Header Insertion	Insert a server header in NetScaler-generated HTTP responses.

To configure HTTP parameters by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Settings, click Change HTTP parameters.
3. In the Configure HTTP parameters dialog box, specify values for some or all of the parameters that appear under the headings listed in the table above.
4. Click OK.

To set the FTP port range by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Settings, click Change global system settings.
3. Under FTP Port Range, in the Start Port and End Port text boxes, type the lowest and highest port numbers, respectively, for the range you want to specify (for example, 5000 and 6000).
4. Click OK.

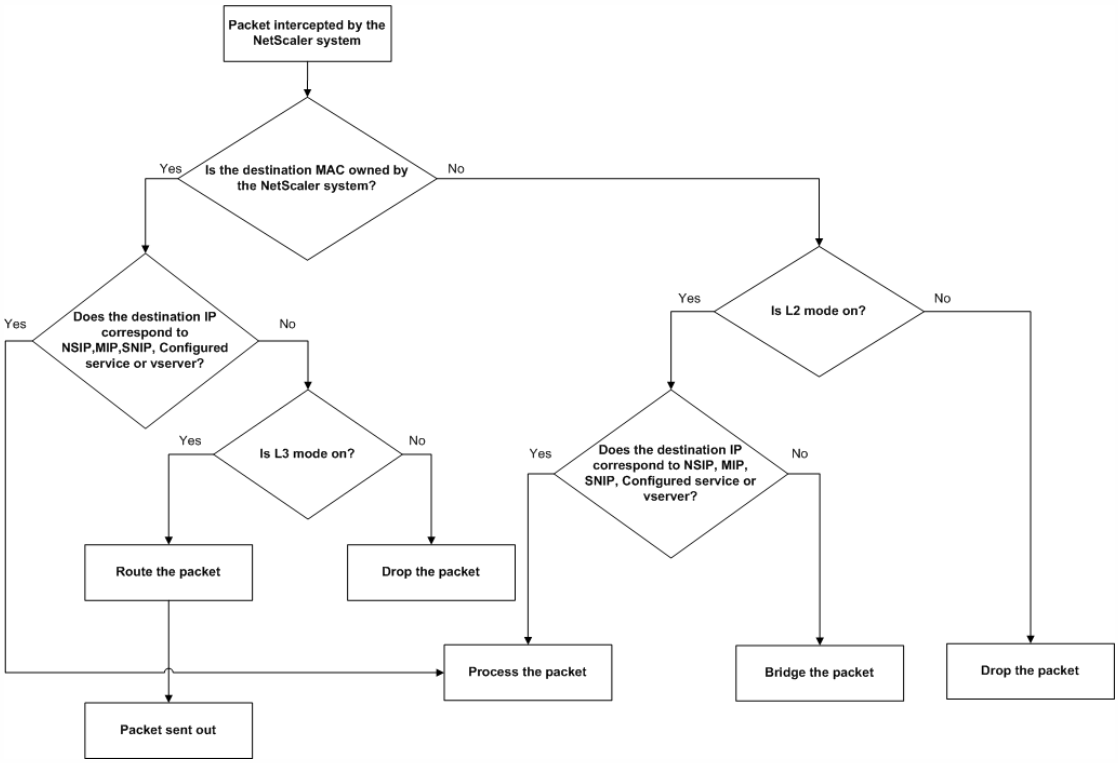
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Modes of Packet Forwarding

Jun 24, 2013

The NetScaler appliance can either route or bridge packets that are not destined for an IP address owned by the appliance (that is, the IP address is not the NSIP, a MIP, a SNIP, a configured service, or a configured virtual server). By default, L3 mode (routing) is enabled and L2 mode (bridging) is disabled, but you can change the configuration. The following flow chart shows how the appliance evaluates packets and either processes, routes, bridges, or drops them.

Figure 1. Interaction between Layer 2 and Layer 3 Modes



An appliance can use the following modes to forward the packets it receives:

- Layer 2 (L2) Mode
- Layer 3 (L3) Mode
- MAC-Based Forwarding Mode

Enabling and Disabling Layer 2 Mode

Updated: 2013-09-13

Layer 2 mode controls the Layer 2 forwarding (bridging) function. You can use this mode to configure a NetScaler appliance to behave as a Layer 2 device and bridge the packets that are not destined for it. When this mode is enabled, packets are not forwarded to any of the MAC addresses, because the packets can arrive on any interface of the appliance and each interface has its own MAC address.

With Layer 2 mode disabled (which is the default), the appliance drops packets that are not destined for one of its MAC address. If another Layer 2 device is installed in parallel with the appliance, Layer 2 mode must be disabled to prevent

© 1999-2017 Citrix Systems, Inc. All rights reserved.

bridging (Layer 2) loops. You can use the configuration utility or the command line to enable Layer 2 mode.

Note: The appliance does not support spanning tree protocol. To avoid loops, if you enable L2 mode, do not connect two interfaces on the appliance to the same broadcast domain.

To enable or disable Layer 2 mode by using the command line interface

At the command prompt, type the following commands to enable/disable Layer 2 mode and verify that it has been enabled/disabled:

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

Examples

```
> enable ns mode l2
 Done
> show ns mode

    Mode              Acronym        Status
    -------           -------        ------
 1)  Fast Ramp          FR            ON
 2)  Layer 2 mode       L2            ON
 .
 .
 .
 Done
>

> disable ns mode l2
 Done
> show ns mode

    Mode              Acronym        Status
    -------           -------        ------
 1)  Fast Ramp          FR            ON
 2)  Layer 2 mode       L2            OFF
 .
 .
 .
 Done
>
```

To enable or disable Layer 2 mode by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure modes.
3. In the Configure Modes dialog box, to enable Layer 2 mode, select the Layer 2 Mode check box. To disable Layer 2 mode, clear the check box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Click OK. The Enable/Disable Mode(s)? message appears in the details pane.
5. Click Yes.

Enabling and Disabling Layer 3 Mode

Updated: 2013-09-13

Layer 3 mode controls the Layer 3 forwarding function. You can use this mode to configure a NetScaler appliance to look at its routing table and forward packets that are not destined for it. With Layer 3 mode enabled (which is the default), the appliance performs route table lookups and forwards all packets that are not destined for any appliance-owned IP address. If you disable Layer 3 mode, the appliance drops these packets.

To enable or disable Layer 3 mode by using the command line interface

At the command prompt, type the following commands to enable/disable Layer 3 mode and verify that it has been enabled/disabled:

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

Examples

```
> enable ns mode l3
 Done
> show ns mode

   Mode                    Acronym        Status
   -------                 -------        ------
 1)  Fast Ramp                FR            ON
 2)  Layer 2 mode              L2            OFF
 .
 .
 .
 9)  Layer 3 mode (ip forwarding)  L3            ON
 .
 .
 .
 Done
>

> disable ns mode l3
 Done
> show ns mode

   Mode                    Acronym        Status
   -------                 -------        ------
 1)  Fast Ramp                FR            ON
 2)  Layer 2 mode              L2            OFF
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
.
.
.
9)  Layer 3 mode (ip forwarding)  L3            OFF
.
.
.
Done
>
```

To enable or disable Layer 3 mode by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure modes.
3. In the Configure Modes dialog box, to enable Layer 3 mode, select the Layer 3 Mode (IP Forwarding) check box. To disable Layer 3 mode, clear the check box.
4. Click OK. The Enable/Disable Mode(s)? message appears in the details pane.
5. Click Yes.

Enabling and Disabling MAC-Based Forwarding Mode

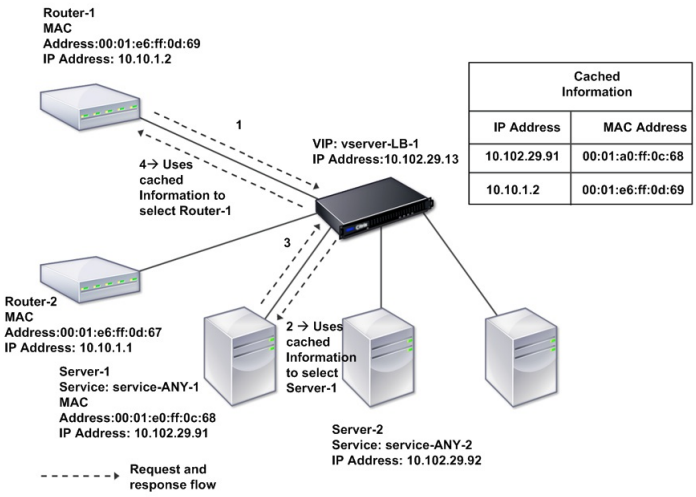
Updated: 2013-09-13

You can use MAC-based forwarding to process traffic more efficiently and avoid multiple-route or ARP lookups when forwarding packets, because the NetScaler appliance remembers the MAC address of the source. To avoid multiple lookups, the appliance caches the source MAC address of every connection for which it performs an ARP lookup, and it returns the data to the same MAC address.

MAC-based forwarding is useful when you use VPN devices because the appliance ensures that all traffic flowing through a particular VPN passes through the same VPN device.

The following figure shows the process of MAC-based forwarding.

Figure 2. MAC-Based Forwarding Process



 © 1999-2017 Citrix Systems, Inc. All rights reserved.

When MAC-based forwarding is enabled, the appliance caches the MAC address of:

- The source (a transmitting device such as router, firewall, or VPN device) of the inbound connection.
- The server that responds to the requests.

When a server responds through an appliance, the appliance sets the destination MAC address of the response packet to the cached address, ensuring that the traffic flows in a symmetric manner, and then forwards the response to the client. The process bypasses the route table lookup and ARP lookup functions. However, when an appliance initiates a connection, it uses the route and ARP tables for the lookup function. To enable MAC-based forwarding, use the configuration utility or the command line.

Some deployments require the incoming and outgoing paths to flow through different routers. In these situations, MAC-based forwarding breaks the topology design. For a global server load balancing (GSLB) site that requires the incoming and outgoing paths to flow through different routers, you must disable MAC-based forwarding and use the appliance's default router as the outgoing router.

With MAC-based forwarding disabled and Layer 2 or Layer 3 connectivity enabled, a route table can specify separate routers for outgoing and incoming connections. To disable MAC-based forwarding, use the configuration utility or the command line.

To enable or disable MAC-based forwarding by using the command line interface

At the command prompt, type the following commands to enable/disable MAC-based forwarding mode and verify that it has been enabled/disabled:

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

  Example

```
> enable ns mode mbf
 Done
> show ns mode

   Mode                    Acronym        Status
   -------                 -------        ------
 1)  Fast Ramp               FR             ON
 2)  Layer 2 mode             L2            OFF
 .
 .
 .
 6)  MAC-based forwarding      MBF           ON
 .
 .
 .
 Done
>

> disable ns mode mbf
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done
> show ns mode

   Mode                    Acronym        Status
   -------                 -------        ------
1)  Fast Ramp              FR             ON
2)  Layer 2 mode           L2             OFF
.
.
.
6)  MAC-based forwarding   MBF            OFF
.
.
.
Done
>
```

To enable or disable MAC-based forwarding by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features group, click Configure modes.
3. In the Configure Modes dialog box, to enable MAC-based forwarding mode, select the MAC Based Forwarding check box. To disable MAC-based forwarding mode, clear the check box.
4. Click OK. The Enable/Disable Mode(s)? message appears in the details pane.
5. Click Yes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Network Interfaces

Aug 24, 2016

NetScaler interfaces are numbered in slot/port notation. In addition to modifying the characteristics of individual interfaces, you can configure virtual LANs to restrict traffic to specific groups of hosts. You can also aggregate links into high-speed channels.

Virtual LANs

The NetScaler supports (Layer 2) port and IEEE802.1Q tagged virtual LANs (VLANs). VLAN configurations are useful when you need to restrict traffic to certain groups of stations. You can configure a network interface to belong to multiple VLANs by using IEEE 802.1q tagging.

You can bind your configured VLANs to IP subnets. The NetScaler (if it is configured as the default router for the hosts on the subnets) then performs IP forwarding between these VLANs. A NetScaler supports the following types of VLANs.

Default VLAN

By default, the network interfaces on a NetScaler are included in a single, port-based VLAN as untagged network interfaces. This default VLAN has a VID of 1 and exists permanently. It cannot be deleted, and its VID cannot be changed.

Port-Based VLANs

A set of network interfaces that share a common, exclusive, Layer 2 broadcast domain define the membership of a port-based VLAN. You can configure multiple port-based VLANs. When you add an interface to a new VLAN as an untagged member, it is automatically removed from the default VLAN.

Tagged VLAN

A network interface can be a tagged or untagged member of a VLAN. Each network interface is an untagged member of only one VLAN (its native VLAN). The untagged network interface forwards the frames for the native VLAN as untagged frames. A tagged network interface can be a part of more than one VLAN. When you configure tagging, be sure that both ends of the link have matching VLAN settings. You can use the configuration utility to define a tagged VLAN (nsvlan) that can have any ports bound as tagged members of the VLAN. Configuring this VLAN requires a reboot of the NetScaler and therefore must be done during initial network configuration.

Link Aggregate Channels

Link aggregation combines incoming data from multiple ports into a single high speed link. Configuring the link aggregate channel increases the capacity and availability of the communication channel between a NetScaler and other connected devices. An aggregated link is also referred to as a channel.

When a network interface is bound to a channel, the channel parameters have precedence over the network interface parameters. A network interface can be bound to only one channel. Binding a network interface to a link aggregate channel changes the VLAN configuration. That is, binding network interfaces to a channel removes them from the VLANs that they originally belonged to and adds them to the default VLAN. However, you can bind the channel back to the old VLAN, or to a new one. For example, if you have bound network interfaces 1/2 and 1/3 to a VLAN with ID 2, and then you bind them to link aggregate channel LA/1, the network interfaces are moved to the default VLAN, but you can bind them to VLAN 2.

Note: You can also use Link Aggregation Control Protocol (LACP) to configure link aggregation. For more information, see Configuring Link Aggregation by Using the Link Aggregation Control Protocol.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Clock Synchronization

Aug 23, 2013

You can configure your NetScaler appliance to synchronize its local clock with a Network Time Protocol (NTP) server. This ensures that its clock has the same date and time settings as the other servers on your network. NTP uses User Datagram Protocol (UDP) port 123 as its transport layer. You have to add NTP servers in the NTP configuration file so that the appliance periodically gets updates from these servers.

If you do not have a local NTP server, you can find a list of public, open access, NTP servers at the official NTP site at http://www.ntp.org.

To configure clock synchronization on your appliance

1. Log on to the command line and enter the shell command.
2. At the shell prompt, copy the ntp.conf file from the /etc directory to the /nsconfig directory. If the file already exists in the /nsconfig directory, make sure that you remove the following entries from the ntp.conf file:
   restrict localhost

   restrict 127.0.0.2

   These entries are required only if you want to run the device as a time server. However, this feature is not supported on the NetScaler.

3. Edit /nsconfig/ntp.conf by typing the IP address for the desired NTP server under the file's server and restrict entries.
4. Create a file named rc.netscaler in the /nsconfig directory, if the file does not already exist in the directory.
5. Edit /nsconfig/rc.netscaler by adding the following entry: /usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ntpd.log &
   This entry starts the ntpd service, checks the ntp.conf file, and logs messages in the /var/log directory.

   Note: If the time difference between the NetScaler and the time server is more than 1000 sec, the ntpd service terminates with a message to the NetScaler log. To avoid this, you need to start ntpd with the -g option, which forcibly syncs the time. Add the following entry in /nsconfig/rc.netscaler:
   /usr/sbin/ntpd -g -c /nsconfig/ntp.conf -l /var/log/ntpd.log &

   If you do not want to forcibly sync the time when there is a large difference, you can set the date manually and then start ntpd again. You can check the time difference between the appliance and the time server by running the following command in the shell:

   ntpdate -q <IP address or domain name of the NTP server>
6. Reboot the appliance to enable clock synchronization.
   Note: If you want to start time synchronization before you restart the appliance, enter the following command (which you added to the rc.netscaler file in step 5) at the shell prompt:
   /usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ ntpd.log &

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DNS

Sep 04, 2013

You can configure a NetScaler appliance to function as an Authoritative Domain Name Server (ADNS), DNS proxy server, End Resolver, or Forwarder. You can add DNS resource records such as SRV Records, AAAA Records, A Records, MX Records, NS Records, CNAME Records, PTR Records, and SOA Records. Also, the appliance can balance the load on external DNS servers.

A common practice is to configure an appliance as a forwarder. For this configuration, you need to add external name servers. After you have added the external servers, you should verify that your configuration is correct.

You can add, remove, enable, and disable external name servers. You can create a name server by specifying its IP address, or you can configure an existing virtual server as the name server.

When adding name servers, you can specify IP addresses or virtual IP addresses (VIPs). If you use IP addresses, the appliance load balances requests to the configured name servers in a round robin manner. If you use VIPs, you can specify any load balancing method.

To add a name server by using the command line interface

At the command prompt, type the following commands to add a name server and verify the configuration:

- add dns nameServer <IP>
- show dns nameServer <IP>

  Example

  ```
  > add dns nameServer 10.102.29.10
   Done
  > show dns nameServer 10.102.29.10
  1)      10.102.29.10  -  State: DOWN
   Done
  >
  ```

To add a name server by using the configuration utility

1. Navigate to Traffic Management > DNS > Name Servers.
2. In the details pane, click Add.
3. In the Create Name Server dialog box, select IP Address.
4. In the IP Address text box, type the IP address of the name server (for example, 10.102.29.10). If you are adding an external name server, clear the Local check box.
5. Click Create, and then click Close.
6. Verify that the name server you added appears in the Name Servers pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

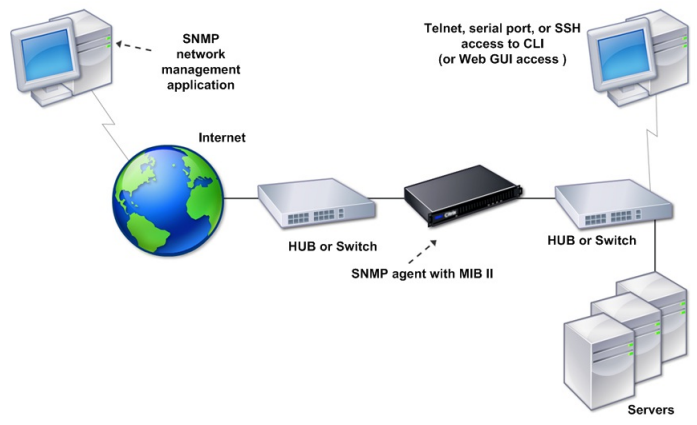
Configuring SNMP

May 14, 2012

The Simple Network Management Protocol (SNMP) network management application, running on an external computer, queries the SNMP agent on the NetScaler. The agent searches the management information base (MIB) for data requested by the network management application and sends the data to the application.

SNMP monitoring uses traps messages and alarms. SNMP traps messages are asynchronous events that the agent generates to signal abnormal conditions, which are indicated by alarms. For example, if you want to be informed when CPU utilization is above 90 percent, you can set up an alarm for that condition. The following figure shows a network with a NetScaler that has SNMP enabled and configured.

Figure 1. SNMP on the NetScaler



The SNMP agent on a NetScaler supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3). Because it operates in bilingual mode, the agent can handle SNMPv2 queries, such as Get-Bulk, and SNMPv1 queries. The SNMP agent also sends traps compliant with SNMPv2 and supports SNMPv2 data types, such as counter64. SNMPv1 managers (programs on other servers that request SNMP information from the NetScaler) use the NS-MIB-smiv1.mib file when processing SNMP queries. SNMPv2 managers use the NS-MIB-smiv2.mib file.

The NetScaler supports the following enterprise-specific MIBs:

A subset of standard MIB-2 groups

Provides MIB-2 groups SYSTEM, IF, ICMP, UDP, and SNMP.

A system enterprise MIB

Provides system-specific configuration and statistics.

To configure SNMP, you specify which managers can query the SNMP agent, add SNMP trap listeners that will receive the SNMP trap messages, and configure SNMP Alarms.

Adding SNMP Managers

Updated: 2013-06-05

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You can configure a workstation running a management application that complies with SNMP version 1, 2, or 3 to access an appliance. Such a workstation is called an SNMP manager. If you do not specify an SNMP manager on the appliance, the appliance accepts and responds to SNMP queries from all IP addresses on the network. If you configure one or more SNMP managers, the appliance accepts and responds to SNMP queries from only those specific IP addresses. When specifying the IP address of an SNMP manager, you can use the netmask parameter to grant access from entire subnets. You can add a maximum of 100 SNMP managers or networks.

To add an SNMP manager by using the command line interface

At the command prompt, type the following commands to add an SNMP manager and verify the configuration:

- add snmp manager <IPAddress> ... [-netmask <netmask>]
- show snmp manager <IPAddress>

  Example

```
> add snmp manager 10.102.29.5 -netmask 255.255.255.255
 Done
> show snmp manager 10.102.29.5
1)     10.102.29.5        255.255.255.255
 Done
>
```

To add an SNMP manager by using the configuration utility

1. In the navigation pane, expand System, expand SNMP, and then click Managers.
2. In the details pane, click Add.
3. In the Add SNMP Manager dialog box, in the IP Address text box, type the IP address of the workstation running the management application (for example, 10.102.29.5).
4. Click Create, and then click Close.
5. Verify that the SNMP manager you added appears in the Details section at the bottom of the pane.

Adding SNMP Traps Listeners

Updated: 2013-09-13

After configuring the alarms, you need to specify the trap listener to which the appliance will send the trap messages. Apart from specifying parameters like IP address and the destination port of the trap listener, you can specify the type of trap (either generic or specific) and the SNMP version.

You can configure a maximum of 20 trap listeners for receiving either generic or specific traps.

To add an SNMP trap listener by using the command line interface

At the command prompt, type the following command to add an SNMP trap and verify that it has been added:

- add snmp trap specific <IP>
- show snmp trap

  Example

```
> add snmp trap specific 10.102.29.3
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 Done
> show snmp trap
Type     DestinationIP   DestinationPort  Version  SourceIP      Min-Severity  Community
----     -------------   ---------------  -------  --------      ------------  ---------
generic  10.102.29.9     162              V2       NetScaler IP  N/A           public
generic  10.102.29.5     162              V2       NetScaler IP  N/A           public
generic  10.102.120.101  162              V2       NetScaler IP  N/A           public
.
.
.
specific 10.102.29.3     162              V2       NetScaler IP  -             public
 Done
>
```

To add an SNMP trap listener by using the configuration utility

1. In the navigation pane, expand System, expand SNMP, and then click Traps.
2. In the details pane, click Add.
3. In the Create SNMP Trap Destination dialog box, in the Destination IP Address text box, type the IP address (for example, 10.102.29.3).
4. Click Create and then click Close.
5. Verify that the SNMP trap you added appears in the Details section at the bottom of the pane.

Configuring SNMP Alarms

Updated: 2013-09-13

You configure alarms so that the appliance generates a trap message when an event corresponding to one of the alarms occurs. Configuring an alarm consists of enabling the alarm and setting the severity level at which a trap is generated. There are five severity levels: Critical, Major, Minor, Warning, and Informational. A trap is sent only when the severity of the alarm matches the severity specified for the trap.

Some alarms are enabled by default. If you disable an SNMP alarm, the appliance will not generate trap messages when corresponding events occur. For example, if you disable the Login-Failure SNMP alarm, the appliance will not generate a trap message when a login failure occurs.

To enable or disable an alarm by using the command line interface

At the command prompt, type the following commands to enable or disable an alarm and verify that it has been enabled or disabled:

- set snmp alarm <trapName> [-state ENABLED | DISABLED]
- show snmp alarm <trapName>

Example

```
> set snmp alarm LOGIN-FAILURE -state ENABLED
 Done
> show snmp alarm LOGIN-FAILURE
  Alarm          Alarm Threshold  Normal Threshold  Time  State    Severity   Logging
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
  -----        ---------------  ----------------  ----  --------  ---------  --------
 1) LOGIN-FAILURE  N/A          N/A            N/A   ENABLED   -       ENABLED
 Done
>
```

To set the severity of the alarm by using the command line interface

At the command prompt, type the following commands to set the severity of the alarm and verify that the severity has been set correctly:

- set snmp alarm <trapName> [-severity <severity>]
- show snmp alarm <trapName>

Example

```
> set snmp alarm LOGIN-FAILURE -severity Major
 Done
> show snmp alarm LOGIN-FAILURE
  Alarm        Alarm Threshold  Normal Threshold  Time  State    Severity   Logging
  -----        ---------------  ----------------  ----  --------  ---------  --------
 1) LOGIN-FAILURE  N/A          N/A            N/A   ENABLED   Major     ENABLED
 Done
>
```

To configure alarms by using the configuration utility

1. In the navigation pane, expand System, expand SNMP, and then click Alarms.
2. In the details pane, select an alarm (for example, LOGIN-FAILURE), and then click Open.
3. In the Configure SNMP Alarm dialog box, to enable the alarm, select Enabled in the State drop-down list. To disable the alarm, select Disabled.
4. In the Severity drop-down list, select a severity option (for example, Major).
5. Click OK, and then click Close.
6. Verify that the parameters for the SNMP alarm you configured are correctly configured by viewing the Details section at the bottom of the pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Verifying the Configuration

Nov 08, 2013

After you finish configuring your system, complete the following checklists to verify your configuration.

Configuration Checklist

- The build running is:
- There are no incompatibility issues. (Incompatibility issues are documented in the build's release notes.)
- The port settings (speed, duplex, flow control, monitoring) are the same as the switch's port.
- Enough mapped IP addresses have been configured to support all server-side connections during peak times.
  - The number of configured mapped IP addresses is: _____
  - The expected number of simultaneous server connections is:
    [ ] 62,000 [ ] 124,000 [ ] Other_____

Topology Configuration Checklist

- The routes have been used to resolve servers on other subnets.

  The routes entered are:

  _____

- If the NetScaler is in a public-private topology, reverse NAT has been configured.
- The failover (high availability) settings configured on the NetScaler resolve in a one arm or two-arm configuration. All unused network interfaces have been disabled:

  _____

- If the NetScaler is placed behind an external load balancer, then the load balancing policy on the external load balancer is not "least connection."
  The load balancing policy configured on the external load balancer is:

  _____

- If the NetScaler is placed in front of a firewall, the session time-out on the firewall is set to a value greater than or equal to 300 seconds.
  Note: The TCP idle connection timeout on a NetScaler appliance is 360 seconds. If the timeout on the firewall is also set to 300 seconds or more, then the appliance can perform TCP connection multiplexing effectively because connections will not be closed earlier.
  The value configured for the session time-out is: _____

Server Configuration Checklist

- "Keep-alive" has been enabled on all the servers.
  The value configured for the keep-alive time-out is: _____

- The default gateway has been set to the correct value. (The default gateway should either be a NetScaler or upstream router.) The default gateway is:

  _____

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- The server port settings (speed, duplex, flow control, monitoring) are the same as the switch port settings.

  _____

- If the Microsoft® Internet Information Server is used, buffering is enabled on the server.
- If an Apache Server is used, the MaxConn (maximum number of connections) parameter is configured on the server and on the NetScaler.
  The MaxConn (maximum number of connections) value that has been set is:

  _____

- If a Netscape® Enterprise Server™ is used, the maximum requests per connection parameter is set on the NetScaler. The maximum requests per connection value that has been set is:

  _____

Software Features Configuration Checklist

- Does the Layer 2 mode feature need to be disabled? (Disable if another Layer 2 device is working in parallel with a NetScaler.)
  Reason for enabling or disabling:

  _____

- Does the MAC-based forwarding feature need to be disabled? (If the MAC address used by return traffic is different, it should be disabled.)
  Reason for enabling or disabling:

  _____

- Does host-based reuse need to be disabled? (Is there virtual hosting on the servers?)
  Reason for enabling or disabling:

  _____

- Do the default settings of the surge protection feature need to be changed?
  Reason for changing or not changing:

  _____

Access Checklist

- The system IPs can be pinged from the client-side network.
- The system IPs can be pinged from the server-side network.
- The managed server(s) can be pinged through the NetScaler.
- Internet hosts can be pinged from the managed servers.
- The managed server(s) can be accessed through the browser.
- The Internet can be accessed from managed server(s) using the browser.
- The system can be accessed using SSH.
- Admin access to all managed server(s) is working.

Note: When you are using the ping utility, ensure that the pinged server has ICMP ECHO enabled, or your ping will not succeed.

Firewall Checklist

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following firewall requirements have been met:

- UDP 161 (SNMP)
- UDP 162 (SNMP trap)
- TCP/UDP 3010 (GUI)
- HTTP 80 (GUI)
- TCP 22 (SSH)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing Traffic on a NetScaler Appliance

Jun 24, 2013

The load balancing feature distributes client requests across multiple servers to optimize resource utilization. In a real-world scenario with a limited number of servers providing service to a large number of clients, a server can become overloaded and degrade the performance of the server farm. A Citrix NetScaler appliance uses load balancing criteria to prevent bottlenecks by forwarding each client request to the server best suited to handle the request when it arrives.

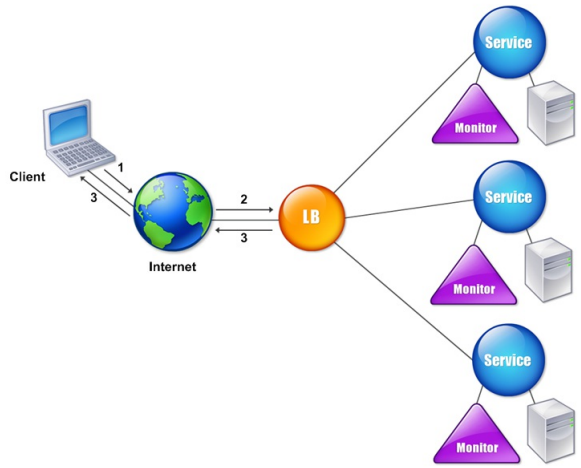
To configure load balancing, you define a virtual server to proxy multiple servers in a server farm and balance the load among them.

When a client initiates a connection to the server, a virtual server terminates the client connection and initiates a new connection with the selected server, or reuses an existing connection with the server, to perform load balancing. The load balancing feature provides traffic management from Layer 4 (TCP and UDP) through Layer 7 (FTP, HTTP, and HTTPS).

The NetScaler appliance uses a number of algorithms, called load balancing methods, to determine how to distribute the load among the servers. The default load balancing method is the Least Connections method.

A typical load balancing deployment consists of the entities described in the following figure.

Figure 1. Load Balancing Architecture



The entities function as follows:

- **Virtual server**. An entity that is represented by an IP address, a port, and a protocol. The virtual server IP address (VIP) is usually a public IP address. The client sends connection requests to this IP address. The virtual server represents a bank of servers.
- **Service**. A logical representation of a server or an application running on a server. Identifies the server's IP address, a port, and a protocol. The services are bound to the virtual servers.
- **Server object**. An entity that is represented by an IP address. The server object is created when you create a service. The IP address of the service is taken as the name of the server object. You can also create a server object and then create

© 1999-2017 Citrix Systems, Inc. All rights reserved.

services by using the server object.

- **Monitor**. An entity that tracks the health of the services. The appliance periodically probes the servers using the monitor bound to each service. If a server does not respond within a specified response timeout, and the specified number of probes fails, the service is marked DOWN. The appliance then performs load balancing among the remaining services.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Load Balancing

Jun 24, 2013

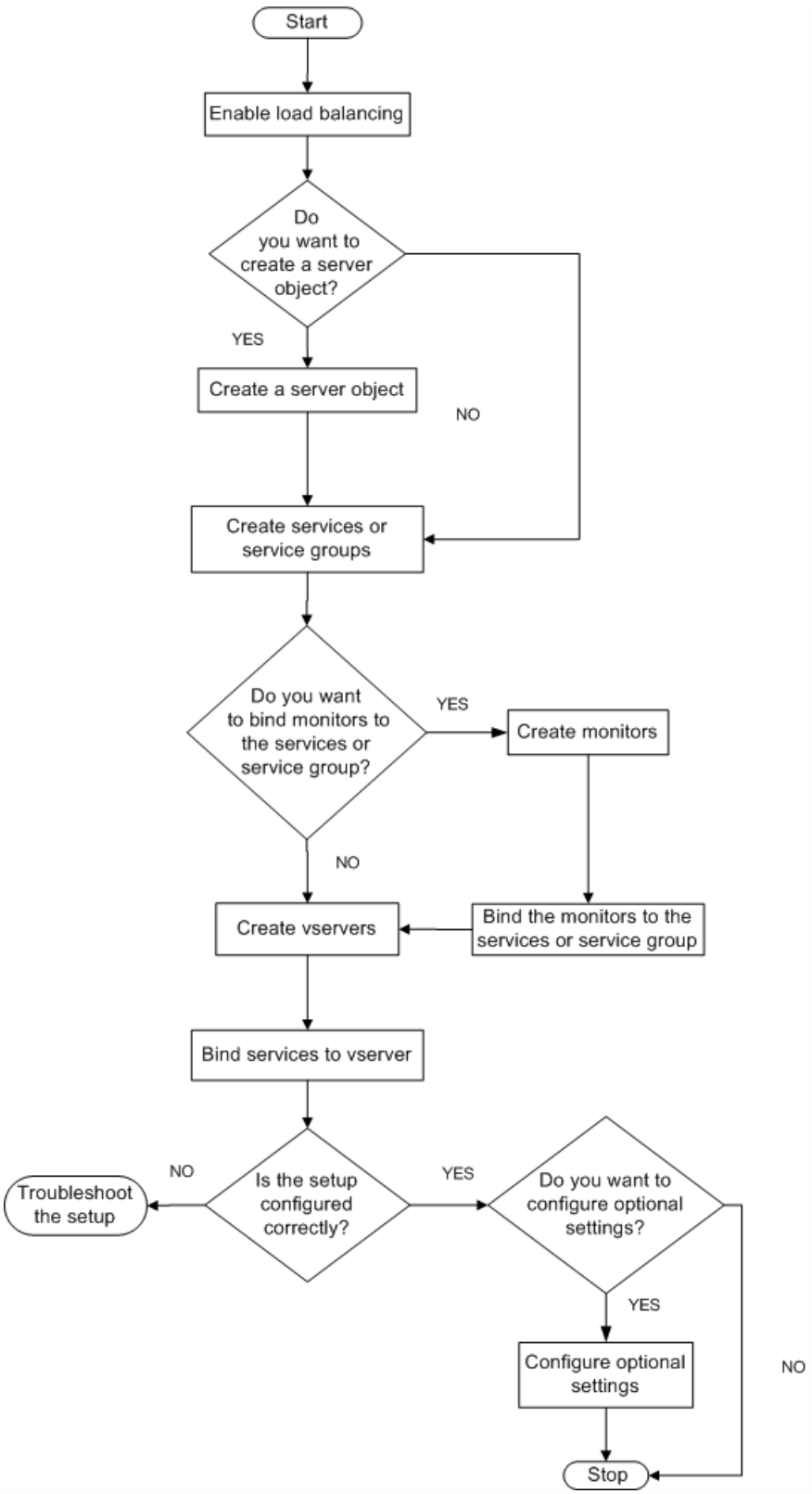
To configure load balancing, you must first create services. Then, you create virtual servers and bind the services to the virtual servers. By default, the NetScaler appliance binds a monitor to each service. After binding the services, verify your configuration by making sure that all of the settings are correct.

Note: After you deploy the configuration, you can display statistics that show how the entities in the configuration are performing. Use the statistical utility or the stat lb vserver <vserverName> command.
Optionally, you can assign weights to a service. The load balancing method then uses the assigned weight to select a service. For getting started, however, you can limit optional tasks to configuring some basic persistence settings, for sessions that must maintain a connection to a particular server, and some basic configuration-protection settings.

The following flow chart illustrates the sequence of the configuration tasks.

Figure 1. Sequence of Tasks to Configure Load Balancing

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Enabling Load Balancing

Updated: 2013-06-05

Before configuring load balancing, make sure that the load balancing feature is enabled.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To enable load balancing by using the command line interface

At the command prompt, type the following commands to enable load balancing and verify that it is enabled:

- enable feature lb
- show feature

Example

```
> enable feature lb
 Done
> show feature

   Feature              Acronym        Status
   -------              -------        ------
 1)  Web Logging          WL            OFF
 2)  Surge Protection     SP            OFF
 3)  Load Balancing       LB            ON
 .
 .
 .
 9)  SSL Offloading       SSL           ON
 .
 .
 .
Done
```

To enable load balancing by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Load Balancing check box, and then click OK.
4. In the Enable/Disable Feature(s)? message, click Yes.

Configuring Services and a Virtual Server

Updated: 2013-06-24

When you have identified the services you want to load balance, you can implement your initial load balancing configuration by creating the service objects, creating a load balancing virtual server, and binding the service objects to the virtual server.

To implement the initial load balancing configuration by using the command line interface

At the command prompt, type the following commands to implement and verify the initial configuration:

- add service <name> <IPaddress> <serviceType> <port>
- add lb vserver <vServerName> <serviceType> [<IPaddress> <port>]
- bind lb vserver <name> <serviceName>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- show service bindings <serviceName>

  Example

```
> add service service-HTTP-1 10.102.29.5 HTTP 80
 Done
> add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
 Done
> bind lb vserver vserver-LB-1 service-HTTP-1
 Done
> show service bindings service-HTTP-1
  service-HTTP-1 (10.102.29.5:80) - State : DOWN

    1)     vserver-LB-1 (10.102.29.60:80) - State : DOWN
 Done
```

To implement the initial load balancing configuration by using the configuration utility

1. Navigate to Traffic Management > Load Balancing.
2. In the details pane, under Getting Started, click Load Balancing wizard, and follow the instructions to create a basic load balancing setup.
3. Return to the navigation pane, expand Load Balancing, and then click Virtual Servers.
4. Select the virtual server that you configured and verify that the parameters displayed at the bottom of the page are correctly configured.
5. Click Open.
6. Verify that each service is bound to the virtual server by confirming that the Active check box is selected for each service on the Services tab.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Choosing and Configuring Persistence Settings

Sep 04, 2013

You must configure persistence on a virtual server if you want to maintain the states of connections on the servers represented by that virtual server (for example, connections used in e-commerce). The appliance then uses the configured load balancing method for the initial selection of a server, but forwards to that same server all subsequent requests from the same client.

If persistence is configured, it overrides the load balancing methods once the server has been selected. If the configured persistence applies to a service that is down, the appliance uses the load balancing methods to select a new service, and the new service becomes persistent for subsequent requests from the client. If the selected service is in an Out Of Service state, it continues to serve the outstanding requests but does not accept new requests or connections. After the shutdown period elapses, the existing connections are closed. The following table lists the types of persistence that you can configure.

Table 1. Limitations on Number of Simultaneous Persistent Connections

Persistence Type	Persistent Connections
Source IP, SSL Session ID, Rule, DESTIP, SRCIPDESTIP	250K
CookieInsert, URL passive, Custom Server ID	Memory limit. In case of CookieInsert, if time out is not 0, any number of connections is allowed until limited by memory.

If the configured persistence cannot be maintained because of a lack of resources on an appliance, the load balancing methods are used for server selection. Persistence is maintained for a configured period of time, depending on the persistence type. Some persistence types are specific to certain virtual servers. The following table shows the relationship.

Table 2. Persistence Types Available for Each Type of Virtual Server

Persistence TypeHeader 1	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge
Source IP	YES	YES	YES	YES	YES
CookieInsert	YES	YES	NO	NO	NO
SSL Session ID	NO	YES	NO	NO	YES
URL Passive	YES	YES	NO	NO	NO
Custom Server ID	YES	YES	NO	NO	NO
Rule	YES	YES	NO	NO	NO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Type	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge
SRCIPDESTIP	N/A	N/A	YES	YES	N/A
Header 1					
DESTIP	N/A	N/A	YES	YES	N/A

You can also specify persistence for a group of virtual servers. When you enable persistence on the group, the client requests are directed to the same selected server regardless of which virtual server in the group receives the client request. When the configured time for persistence elapses, any virtual server in the group can be selected for incoming client requests.

Two commonly used persistence types are persistence based on cookies and persistence based on server IDs in URLs.

Configuring Persistence Based on Cookies

Updated: 2013-08-23

When you enable persistence based on cookies, the NetScaler adds an HTTP cookie into the Set-Cookie header field of the HTTP response. The cookie contains information about the service to which the HTTP requests must be sent. The client stores the cookie and includes it in all subsequent requests, and the NetScaler uses it to select the service for those requests. You can use this type of persistence on virtual servers of type HTTP or HTTPS.

The NetScaler inserts the cookie <NSC_XXXX>= <ServiceIP> <ServicePort>

where:

- <NSC_XXXX> is the virtual server ID that is derived from the virtual server name.
- <ServiceIP> is the hexadecimal value of the IP address of the service.
- <ServicePort> is the hexadecimal value of the port of the service.

The NetScaler encrypts ServiceIP and ServicePort when it inserts a cookie, and decrypts them when it receives a cookie.

Note: If the client is not allowed to store the HTTP cookie, the subsequent requests do not have the HTTP cookie, and persistence is not honored.
By default, the NetScaler sends HTTP cookie version 0, in compliance with the Netscape specification. It can also send version 1, in compliance with RFC 2109.

You can configure a timeout value for persistence that is based on HTTP cookies. Note the following:

- If HTTP cookie version 0 is used, the NetScaler inserts the absolute Coordinated Universal Time (GMT) of the cookie's expiration (the expires attribute of the HTTP cookie), calculated as the sum of the current GMT time on a NetScaler, and the timeout value.
- If an HTTP cookie version 1 is used, the NetScaler inserts a relative expiration time (Max-Age attribute of the HTTP cookie). In this case, the client software calculates the actual expiration time.

Note: Most client software currently installed (Microsoft Internet Explorer and Netscape browsers) understand HTTP cookie version 0; however, some HTTP proxies understand HTTP cookie version 1.
If you set the timeout value to 0, the NetScaler does not specify the expiration time, regardless of the HTTP cookie version used. The expiration time then depends on the client software, and such cookies are not valid if that software is shut down. This persistence type does not consume any system resources. Therefore, it can accommodate an unlimited number of persistent clients.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

An administrator can use the procedure in the following table to change the HTTP cookie version.

To change the HTTP cookie version by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, click Change HTTP Parameters.
3. In the Configure HTTP Parameters dialog box, under Cookie, select Version 0 or Version 1.

Note: For information about the parameters, see "Configuring Persistence Based on Cookies."

To configure persistence based on cookies by using the command line interface

At the command prompt, type the following commands to configure persistence based on cookies and verify the configuration:

- set lb vserver <name> -persistenceType COOKIEINSERT
- show lb vserver <name>

  Example

```
> set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
 Done
> show lb vserver vserver-LB-1
   vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
   .
   .
   .
   Persistence: COOKIEINSERT (version 0)   Persistence Timeout: 2 min
   .
   .
   .
 Done
>
```

To configure persistence based on cookies by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure persistence (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Method and Persistence tab, in the Persistence list, select COOKIEINSERT.
4. In the Time-out (min) text box, type the time-out value (for example, 2).
5. Click OK.
6. Verify that the virtual server for which you configured persistence is correctly configured by selecting the virtual server and viewing the Details section at the bottom of the pane.

Configuring Persistence Based on Server IDs in URLs

Updated: 2013-08-23

The NetScaler can maintain persistence based on the server IDs in the URLs. In a technique called URL passive persistence, the NetScaler extracts the server ID from the server response and embeds it in the URL query of the client request. The

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

server ID is an IP address and port specified as a hexadecimal number. The NetScaler extracts the server ID from subsequent client requests and uses it to select the server.

URL passive persistence requires configuring either a payload expression or a policy infrastructure expression specifying the location of the server ID in the client requests. For more information about expressions, see "Policy Configuration and Reference."

Note: If the server ID cannot be extracted from the client requests, server selection is based on the load balancing method.

Example: Payload Expression

The expression, URLQUERY contains sid= configures the system to extract the server ID from the URL query of a client request, after matching token sid=. Thus, a request with the URL http://www.citrix.com/index.asp?&sid;=c0a864100050 is directed to the server with the IP address10.102.29.10 and port 80.

The timeout value does not affect this type of persistence, which is maintained as long as the server ID can be extracted from the client requests. This persistence type does not consume any system resources, so it can accommodate an unlimited number of persistent clients.

Note: For information about the parameters, see "Load Balancing."

To configure persistence based on server IDs in URLs by using the command line interface

At the command prompt, type the following commands to configure persistence based on server IDs in URLs and verify the configuration:

- set lb vserver <name> -persistenceType URLPASSIVE
- show lb vserver <name>

  Example

```
> set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
 Done
> show lb vserver vserver-LB-1
   vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
   .
   .
   .
   Persistence: URLPASSIVE Persistence Timeout: 2 min
   .
   .
   .
 Done
>
```

To configure persistence based on server IDs in URLs by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure persistence (for example, vserver-LB-1), and then click Open.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Configure Virtual Server (Load Balancing) dialog box, on the Method and Persistence tab, in the Persistence list, select URLPASSIVE.
4. In the Time-out (min) text box, type the time-out value (for example, 2).
5. In the Rule text box, enter a valid expression. Alternatively, click Configure next to the Rule text box and use the Create Expression dialog box to create an expression.
6. Click OK.
7. Verify that the virtual server for which you configured persistence is correctly configured by selecting the virtual server and viewing the Details section at the bottom of the pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Features to Protect the Load Balancing Configuration

Jun 24, 2013

You can configure URL redirection to provide notifications of virtual server malfunctions, and you can configure backup virtual servers to take over if a primary virtual server becomes unavailable.

Configuring URL Redirection

Updated: 2013-06-24

You can configure a redirect URL to communicate the status of the appliance in the event that a virtual server of type HTTP or HTTPS is down or disabled. This URL can be a local or remote link. The appliance uses HTTP 302 redirect.

Redirects can be absolute URLs or relative URLs. If the configured redirect URL contains an absolute URL, the HTTP redirect is sent to the configured location, regardless of the URL specified in the incoming HTTP request. If the configured redirect URL contains only the domain name (relative URL), the HTTP redirect is sent to a location after appending the incoming URL to the domain configured in the redirect URL.

Note: If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. In this case, a redirect is used when both the primary and backup virtual servers are down.

To configure a virtual server to redirect client requests to a URL by using the command line interface

At the command prompt, type the following commands to configure a virtual server to redirect client requests to a URL and verify the configuration:

- set lb vserver <name> -redirectURL <URL>
- show lb vserver <name>

  Example

```
> set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
 Done
> show lb vserver vserver-LB-1
   vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
   State: DOWN
   Last state change was at Wed Jun 17 08:56:34 2009 (+666 ms)
   .
   .
   .
   Redirect URL: http://www.newdomain.com/mysite/maintenance
   .
   .
   .
 Done
>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure a virtual server to redirect client requests to a URL by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure URL redirection (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Advanced tab, in the Redirect URL text box, type the URL (for example, http://www.newdomain.com/mysite/maintenance), and then click OK.
4. Verify that the redirect URL you configured for the server appears in the Details section at the bottom of the pane.

Configuring Backup Virtual Servers

Updated: 2013-06-24

If the primary virtual server is down or disabled, the appliance can direct the connections or client requests to a backup virtual server that forwards the client traffic to the services. The appliance can also send a notification message to the client regarding the site outage or maintenance. The backup virtual server is a proxy and is transparent to the client.

You can configure a backup virtual server when you create a virtual server or when you change the optional parameters of an existing virtual server. You can also configure a backup virtual server for an existing backup virtual server, thus creating a cascaded backup virtual server. The maximum depth of cascading backup virtual servers is 10. The appliance searches for a backup virtual server that is up and accesses that virtual server to deliver the content.

You can configure URL redirection on the primary for use when the primary and the backup virtual servers are down or have reached their thresholds for handling requests.

Note: If no backup virtual server exists, an error message appears, unless the virtual server is configured with a redirect URL. If both a backup virtual server and a redirect URL are configured, the backup virtual server takes precedence.

To configure a backup virtual server by using the command line interface

At the command prompt, type the following commands to configure a backup server and verify the configuration:

- set lb vserver <name> [-backupVserver <string>]
- show lb vserver <name>

  Example

```
> set lb vserver vserver-LB-1 -backupVserver vserver-LB-2
 Done
> show lb vserver vserver-LB-1
   vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
   State: DOWN
   Last state change was at Wed Jun 17 08:56:34 2009 (+661 ms)
   .
   .
   .
   Backup: vserver-LB-2
   .
   .
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

.
Done
>

To set up a backup virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the backup virtual server (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Advanced tab, in the Backup Virtual Server list, select the backup virtual server (for example, vserver-LB-2, and then click OK.
4. Verify that the backup virtual server you configured appears in the Details section at the bottom of the pane.
   Note: If the primary server goes down and then comes back up, and you want the backup virtual server to function as the primary server until you explicitly reestablish the primary virtual server, select the Disable Primary When Down check box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

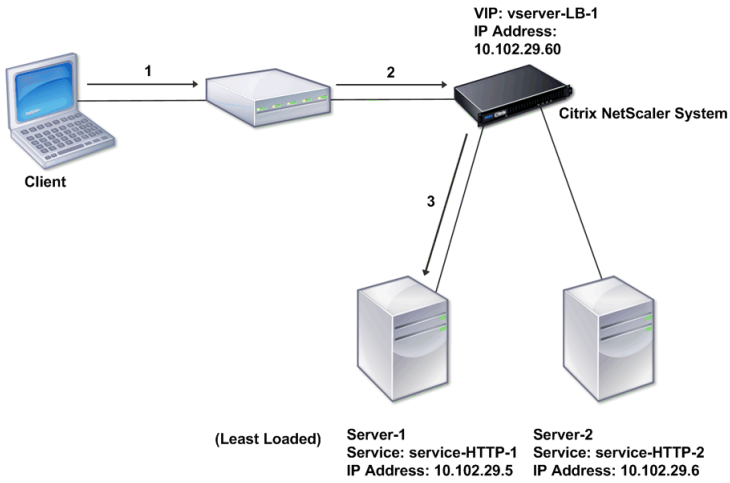
A Typical Load Balancing Scenario

Aug 23, 2013

In a load balancing setup, the NetScaler appliances are logically located between the client and the server farm, and they manage traffic flow to the servers.

The following figure shows the topology of a basic load balancing configuration.

Figure 1. Basic Load Balancing Topology



The virtual server selects the service and assigns it to serve client requests. Consider the scenario in the preceding figure, where the services service-HTTP-1 and service-HTTP-2 are created and bound to the virtual server named virtual server-LB-1. Virtual server-LB-1 forwards the client request to either service-HTTP-1 or service-HTTP-2. The system selects the service for each request by using the Least Connections load balancing method. The following table lists the names and values of the basic entities that must be configured on the system.

Table 1. LB Configuration Parameter Values

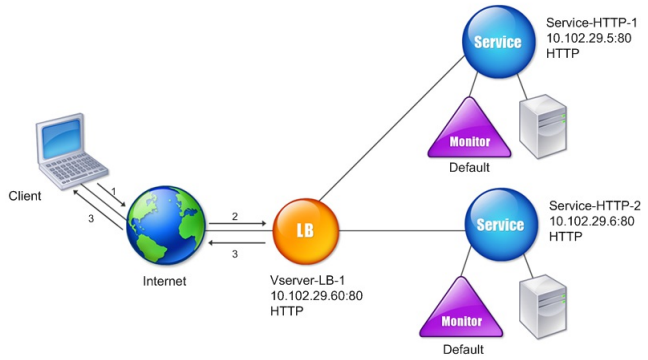
Entity Type	Required parameters and sample values			
	Name	**IP Address**	**Port**	**Protocol**
Virtual Server	vserver-LB-1	10.102.29.60	80	HTTP
Services	service-HTTP-1	10.102.29.5	8083	HTTP
	service-HTTP-2	10.102.29.6	80	HTTP
Monitors	Default	None	None	None

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity Type	Required parameters and sample values			
	Name	IP Address	Port	Protocol

The following figure shows the load balancing sample values and required parameters that are described in the preceding table.

Figure 2. Load Balancing Entity Model



The following tables list the commands used to configure this load balancing setup by using the command line interface.

Table 2. Initial Configuration Tasks

Task	Command
To enable load balancing	enable feature lb
To create a service named service-HTTP-1	add service service-HTTP-1 10.102.29.5 HTTP 80
To create a service named service-HTTP-2	add service service-HTTP-2 10.102.29.6 HTTP 80
To create a virtual server named vserver-LB-1	add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
To bind a service named service-HTTP-1 to a virtual server named vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-1
To bind a service named service-HTTP-2 to a virtual server named vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-2

For more information about the initial configuration tasks, see "Enabling Load Balancing" and "Configuring Services and a Vserver."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 3. Verification Tasks

Task	Command
To view the properties of a virtual server named vserver-LB-1	show lb vserver vserver-LB-1
To view the statistics of a virtual server named vserver-LB-1	stat lb vserver vserver-LB-1
To view the properties of a service named service-HTTP-1	show service service-HTTP-1
To view the statistics of a service named service-HTTP-1	stat service service-HTTP-1
To view the bindings of a service named service-HTTP-1	show service bindings service-HTTP-1

Table 4. Customization Tasks

Task	Command
To configure persistence on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType SOURCEIP -persistenceMask 255.255.255.255 -timeout 2
To configure COOKIEINSERT persistence on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
To configure URLPassive persistence on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
To configure a virtual server to redirect the client request to a URL on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
To set a backup virtual server on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -backupVserver vserver-LB-2

For more information about configuring persistence, see "Choosing and Configuring Persistence Settings." For information about configuring a virtual server to redirect a client request to a URL and setting up a backup virtual server, see "Configuring Features to Protect the Load Balancing Configuration."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Accelerating Load Balanced Traffic by Using Compression

Aug 22, 2013

Compression is a popular means of optimizing bandwidth usage, and most web browsers support compressed data. If you enable the compression feature, the NetScaler appliance intercepts requests from clients and determines whether the client can accept compressed content. After receiving the HTTP response from the server, the appliance examines the content to determine whether it is compressible. If the content is compressible, the appliance compresses it, modifies the response header to indicate the type of compression performed, and forwards the compressed content to the client.

NetScaler compression is a policy-based feature. A policy filters requests and responses to identify responses to be compressed, and specifies the type of compression to apply to each response. The appliance provides several built-in policies to compress common MIME types such as text/html, text/plain, text/xml, text/css, text/rtf, application/msword, application/vnd.ms-excel, and application/vnd.ms-powerpoint. You can also create custom policies. The appliance does not compress compressed MIME types such as application/octet-stream, binary, bytes, and compressed image formats such as GIF and JPEG.

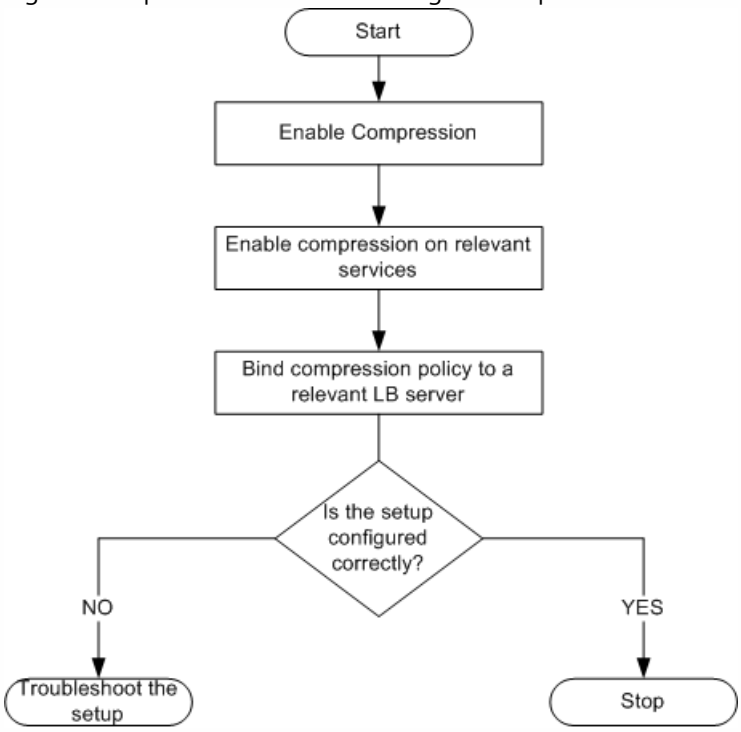
To configure compression, you must enable it globally and on each service that will provide responses that you want compressed. If you have configured virtual servers for load balancing or content switching, you should bind the polices to the virtual servers. Otherwise, the policies apply to all traffic that passes through the appliance.

Compression Configuration Task Sequence

Updated: 2013-08-22

The following flow chart shows the sequence of tasks for configuring basic compression in a load balancing setup.

Figure 1. Sequence of Tasks to Configure Compression



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: The steps in the above figure assume that load balancing has already been configured.

Enabling Compression

Updated: 2013-06-07

By default, compression is not enabled. You must enable the compression feature to allow compression of HTTP responses that are sent to the client.

To enable compression by using the command line interface

At the command prompt, type the following commands to enable compression and verify the configuration:

- enable ns feature CMP
- show ns feature

  Example

```
> enable ns feature CMP
Done
> show ns feature

    Feature              Acronym        Status
    -------              -------        ------
1)  Web Logging          WL             ON
2)  Surge Protection     SP             OFF
.
7)  Compression Control  CMP            ON
8)  Priority Queuing     PQ             OFF
.
Done
```

To enable compression by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Compression check box, and then click OK.
4. In the Enable/Disable Feature(s)? dialog box, click Yes.

Configuring Services to Compress Data

Updated: 2013-08-22

In addition to enabling compression globally, you must enable it on each service that will deliver files to be compressed.

To enable compression on a service by using the command line

At the command prompt, type the following commands to enable compression on a service and verify the configuration:

- set service <name> -CMP YES
- show service <name>

  Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> show service SVC_HTTP1
SVC_HTTP1 (10.102.29.18:80) - HTTP
State: UP
Last state change was at Tue Jun 16 06:19:14 2009 (+737 ms)
Time since last state change: 0 days, 03:03:37.200
Server Name: 10.102.29.18
Server ID : 0   Monitor Threshold : 0
Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
Use Source IP: NO
Client Keepalive(CKA): NO
Access Down Service: NO
TCP Buffering(TCPB): NO
HTTP Compression(CMP): YES
Idle timeout: Client: 180 sec   Server: 360 sec
Client IP: DISABLED
Cacheable: NO
SC: OFF
SP: OFF
Down state flush: ENABLED

1)      Monitor Name: tcp-default
State: DOWN     Weight: 1
Probes: 1095    Failed [Total: 1095 Current: 1095]
Last response: Failure - TCP syn sent, reset received.
Response Time: N/A
Done
```

To enable compression on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, select the service for which you want to configure compression (for example, service-HTTP-1), and then click Open.
3. On the Advanced tab, under Settings, select the Compression check box, and then click OK.
4. Verify that, when the service is selected, HTTP Compression(CMP): ON appears in the **Details** section at the bottom of the pane.

Binding a Compression Policy to a Virtual Server

Updated: 2013-09-04

If you bind a policy to a virtual server, the policy is evaluated only by the services associated with that virtual server. You can bind compression policies to a virtual server either from the Configure Virtual Server (Load Balancing) dialog box or from the Compression Policy Manager dialog box. This topic includes instructions to bind compression policies to a load balancing virtual server by using the Configure Virtual Server (Load Balancing) dialog box. For information about how you can bind a compression policy to a load balancing virtual server by using the Compression Policy Manager dialog box, see "Configuring and Binding Policies with the Policy Manager."

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To bind or unbind a compression policy to a virtual server by using the command line

At the command prompt, type the following commands to bind or unbind a compression policy to a load balancing virtual server and verify the configuration:

- (bind | unbind) lb vserver <name> -policyName <string>
- show lb vserver <name>

Example

```
> bind lb vserver lbvip -policyName ns_cmp_msapp
Done
> show lb vserver lbvip
lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
State: UP
Last state change was at Thu May 28 05:37:21 2009 (+685 ms)
Time since last state change: 19 days, 04:26:50.470
Effective State: UP
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Port Rewrite : DISABLED
No. of Bound Services :  1 (Total)       1 (Active)
Configured Method: LEASTCONNECTION
Current Method: Round Robin, Reason: Bound service's state changed to UP
Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED  Push VServer:
Push Multi Clients: NO
Push Label Rule:

Bound Service Groups:
1)     Group Name: Service-Group-1

1) Service-Group-1 (10.102.29.252: 80) - HTTP State: UP Weight:                    1

1)     Policy : ns_cmp_msapp Priority:0
Done
```

To bind or unbind a compression policy to a load balancing virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server to which you want to bind or unbind a compression policy (for example, Vserver-LB-1), and then click Open.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Configure Virtual Server (Load Balancing) dialog box, on the Policies tab, click Compression.
4. Do one of the following:
   - To bind a compression policy, click Insert Policy, and then select the policy you want to bind to the virtual server.
   - To unbind a compression policy, click the name of the policy you want to unbind from the virtual server, and then click Unbind Policy.
5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Securing Load Balanced Traffic by Using SSL

Jan 31, 2011

The Citrix NetScaler SSL offload feature transparently improves the performance of web sites that conduct SSL transactions. By offloading CPU-intensive SSL encryption and decryption tasks from the local web server to the appliance, SSL offloading ensures secure delivery of web applications without the performance penalty incurred when the server processes the SSL data. Once the SSL traffic is decrypted, it can be processed by all standard services. The SSL protocol works seamlessly with various types of HTTP and TCP data and provides a secure channel for transactions using such data.

To configure SSL, you must first enable it. Then, you configure HTTP or TCP services and an SSL virtual server on the appliance, and bind the services to the virtual server. You must also add a certificate-key pair and bind it to the SSL virtual server. If you use Outlook Web Access servers, you must create an action to enable SSL support and a policy to apply the action. An SSL virtual server intercepts incoming encrypted traffic and decrypts it by using a negotiated algorithm. The SSL virtual server then forwards the decrypted data to the other entities on the appliance for appropriate processing.

This document includes the following:

- SSL Configuration Task Sequence
- Enabling SSL Offload
- Creating HTTP Services
- Adding an SSL-Based Virtual Server
- Binding Services to the SSL Virtual Server
- Adding a Certificate Key Pair
- Binding an SSL Certificate Key Pair to the Virtual Server
- Configuring Support for Outlook Web Access

SSL Configuration Task Sequence

To configure SSL, you must first enable it. Then, you must create an SSL virtual server and HTTP or TCP services on the NetScaler. Finally, you must bind a valid SSL certificate and the configured services to the SSL virtual server.

An SSL virtual server intercepts incoming encrypted traffic and decrypts it using a negotiated algorithm. The SSL virtual server then forwards the decrypted data to the other entities on the NetScaler for appropriate processing.

The following flow chart shows the sequence of tasks for configuring a basic SSL offload setup.

Figure 1. Sequence of Tasks to Configure SSL Offloading

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling SSL Offload

You should enable the SSL feature before configuring SSL offload. You can configure SSL-based entities on the appliance without enabling the SSL feature, but they will not work until you enable SSL.

To enable SSL by using the command line interface

At the command prompt, type the following commands to enable SSL Offload and verify the configuration:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- enable ns feature SSL
- show ns feature

Example

```
> enable ns feature ssl
Done
> show ns feature
Feature Acronym Status
------- ------- ------
1) Web Logging WL ON
2) SurgeProtection SP OFF
3) Load Balancing LB ON . . .
 9) SSL Offloading SSL ON
10) Global Server Load Balancing GSLB ON . .
Done >
```

To enable SSL by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Change basic features.
3. Select the SSL Offloading check box, and then click OK.
4. In the Enable/Disable Feature(s)? message box, click Yes.

Creating HTTP Services

A service on the appliance represents an application on a server. Once configured, services are in the disabled state until the appliance can reach the server on the network and monitor its status. This topic covers the steps to create an HTTP service.

Note: For TCP traffic, perform the procedures in this and the following topics, but create TCP services instead of HTTP services.

To add an HTTP service by using the command line interface

At the command prompt, type the following commands to add a HTTP service and verify the configuration:

- add service <name> (<IP> | <serverName>) <serviceType> <port>
- show service <name>

```
> add service SVC_HTTP1 10.102.29.18 HTTP 80
 Done
> show service SVC_HTTP1
      SVC_HTTP1 (10.102.29.18:80) - HTTP
      State: UP
      Last state change was at Wed Jul 15 06:13:05 2009
      Time since last state change: 0 days, 00:00:15.350
      Server Name: 10.102.29.18
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
    TCP Buffering(TCPB): NO
    HTTP Compression(CMP): YES
    Idle timeout: Client: 180 sec   Server: 360 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: OFF
    Down state flush: ENABLED

1)    Monitor Name: tcp-default
        State: UP       Weight: 1
        Probes: 4       Failed [Total: 0 Current: 0]
        Last response: Success - TCP syn+ack received.
        Response Time: N/A
Done
```

To add an HTTP service by using the configuration utility

1. Navigate to Traffic Management > SSL Offload > Services.
2. In details pane, click Add.
3. In the Create Service dialog box, in the Service Name, Server, and Port text boxes, type the name of the service, IP address, and port (for example, SVC_HTTP1, 10.102.29.18, and 80).
4. In the Protocol list, select the type of the service (for example, HTTP).
5. Click Create, and then click Close. The HTTP service you configured appears in the Services page.
6. Verify that the parameters you configured are correctly configured by selecting the service and viewing the Details section at the bottom of the pane.

Adding an SSL-Based Virtual Server

In a basic SSL offloading setup, the SSL virtual server intercepts encrypted traffic, decrypts it, and sends the clear text messages to the services that are bound to the virtual server. Offloading CPU-intensive SSL processing to the appliance allows the back-end servers to process a greater number of requests.

To add an SSL-based virtual server by using the command line interface

At the command prompt, type the following commands to create an SSL-based virtual server and verify the configuration:

- add lb vserver <name> <serviceType> [<IPAddress> <port>]
- show lb vserver <name>

  Example

```
> add lb vserver vserver-SSL-1 SSL 10.102.29.50 443
Done
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> show lb vserver vserver-SSL-1
vserver-SSL-1 (10.102.29.50:443) - SSL Type: ADDRESS
State: DOWN[Certkey not bound] Last state change was at Tue Jun 16 06:33:08 2009 (+176 ms)
Time since last state change: 0 days, 00:03:44.120
Effective State: DOWN Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
No. of Bound Services : 0 (Total) 0 (Active)
Configured Method: LEASTCONNECTION Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule: Done
```

Caution: To ensure secure connections, you must bind a valid SSL certificate to the SSL-based virtual server before you enable it.

To add an SSL-based virtual server by using the configuration utility

1. Navigate to Traffic Management > SSL Offload > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (SSL Offload) dialog box, in the Name, IP Address, and Port text boxes, type the name of the virtual server, IP address, and port (for example, Vserver-SSL-1, 10.102.29.50, and 443).
4. In the Protocol list, select the type of the virtual server, for example, SSL.
5. Click Create, and then click Close.
6. Verify that the parameters you configured are correctly configured by selecting the virtual server and viewing the Details section at the bottom of the pane. The virtual server is marked as DOWN because a certificate-key pair and services have not been bound to it.

Caution: To ensure secure connections, you must bind a valid SSL certificate to the SSL-based virtual server before you enable it.

Binding Services to the SSL Virtual Server

After decrypting the incoming data, the SSL virtual server forwards the data to the services that you have bound to the virtual server.

Data transfer between the appliance and the servers can be encrypted or in clear text. If the data transfer between the appliance and the servers is encrypted, the entire transaction is secure from end to end. For more information about configuring the system for end-to-end security, see "SSL Offload and Acceleration."

To bind a service to a virtual server by using the command line interface

At the command prompt, type the following commands to bind service to the SSL virtual server and verify the configuration:

- bind lb vserver <name> <serviceName>
- show lb vserver <name>

Example

```
> bind lb vserver vserver-SSL-1 SVC_HTTP1
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> show lb vserver vserver-SSL-1 vserver-SSL-1 (10.102.29.50:443) - SSL Type:
ADDRESS State: DOWN[Certkey not bound]
Last state change was at Tue Jun 16 06:33:08 2009 (+174 ms)
Time since last state change: 0 days, 00:31:53.70
Effective State: DOWN Client Idle
Timeout: 180 sec
Down state flush: ENABLED Disable Primary Vserver On Down :
DISABLED No. of Bound Services : 1 (Total) 0 (Active)
Configured Method: LEASTCONNECTION Mode: IP Persistence: NONE Vserver IP and
Port insertion: OFF Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule:

1) SVC_HTTP1 (10.102.29.18: 80) - HTTP
State: DOWN Weight: 1
Done
```

To bind a service to a virtual server by using the configuration utility

1. Navigate to Traffic Management > SSL Offload > Virtual Servers.
2. In the details pane, select a virtual server, and then click Open.
3. On the Services tab, in the Active column, select the check boxes next to the services that you want to bind to the selected virtual server.
4. Click OK.
5. Verify that the Number of Bound Services counter in the Details section at the bottom of the pane is incremented by the number of services that you bound to the virtual server.

Adding a Certificate Key Pair

An SSL certificate is an integral element of the SSL Key-Exchange and encryption/decryption process. The certificate is used during SSL handshake to establish the identity of the SSL server. You can use a valid, existing SSL certificate that you have on the NetScaler appliance, or you can create your own SSL certificate. The appliance supports RSA/DSA certificates of up to 4096 bits.

Note: Citrix recommends that you use a valid SSL certificate that has been issued by a trusted certificate authority. Invalid certificates and self-created certificates are not compatible with all SSL clients.
Before a certificate can be used for SSL processing, you must pair it with its corresponding key. The certificate key pair is then bound to the virtual server and used for SSL processing.

To add a certificate key pair by using the command line interface

At the command prompt, type the following commands to create a certificate key pair and verify the configuration:

- add ssl certKey <certkeyName> -cert <string> [-key <string>]
- show sslcertkey <name>

Example

```
> add ssl certKey CertKey-SSL-1 -cert ns-root.cert -key ns-root.key
 Done
> show sslcertkey CertKey-SSL-1
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
  Name: CertKey-SSL-1 Status: Valid,
  Days to expiration:4811 Version: 3
  Serial Number: 00 Signature Algorithm: md5WithRSAEncryption Issuer: C=US,ST=California,L=San
  Jose,O=Citrix ANG,OU=NS Internal,CN=de fault
  Validity Not Before: Oct 6 06:52:07 2006 GMT Not After : Aug 17 21:26:47 2022 GMT
  Subject: C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,CN=d efault Public Key
  Algorithm: rsaEncryption Public Key
  size: 1024
Done
```

To add a certificate key pair by using the configuration utility

1. Navigate to Traffic Management > SSL > Certificates.
2. In the details pane, click Add.
3. In the Install Certificate dialog box, in the Certificate-Key Pair Name text box, type a name for the certificate key pair you want to add, for example, Certkey-SSL-1.
4. Under Details, in Certificate File Name, click Browse (Appliance) to locate the certificate. Both the certificate and the key are stored in the /nsconfig/ssl/ folder on the appliance. To use a certificate present on the local system, select Local.
5. Select the certificate you want to use, and then click Select.
6. In Private Key File Name, click Browse (Appliance) to locate the private key file. To use a private key present on the local system, select Local.
7. Select the key you want to use and click Select. To encrypt the key used in the certificate key pair, type the password to be used for encryption in the Password text box.
8. Click Install.
9. Double-click the certificate key pair and, in the Certificate Details window, verify that the parameters have been configured correctly and saved.

Binding an SSL Certificate Key Pair to the Virtual Server

After you have paired an SSL certificate with its corresponding key, you must bind the certificate key pair to the SSL virtual server so that it can be used for SSL processing. Secure sessions require establishing a connection between the client computer and an SSL-based virtual server on the appliance. SSL processing is then carried out on the incoming traffic at the virtual server. Therefore, before enabling the SSL virtual server on the appliance, you need to bind a valid SSL certificate to the SSL virtual server.

To bind an SSL certificate key pair to a virtual server by using the command line interface

At the command prompt, type the following commands to bind an SSL certificate key pair to a virtual server and verify the configuration:

- bind ssl vserver <vServerName> -certkeyName <string>
- show ssl vserver <name>

Example

```
 > bind ssl vserver Vserver-SSL-1 -certkeyName CertKey-SSL-1
Done
> show ssl vserver Vserver-SSL-1
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Advanced SSL configuration for VServer Vserver-SSL-1:
    DH: DISABLED
    Ephemeral RSA: ENABLED Refresh Count: 0
    Session Reuse: ENABLED Timeout: 120 seconds
    Cipher Redirect: ENABLED
    SSLv2 Redirect: ENABLED
    ClearText Port: 0
    Client Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

  1) CertKey Name: CertKey-SSL-1 Server Certificate
  1) Cipher Name: DEFAULT
     Description: Predefined Cipher Alias
  Done
```

To bind an SSL certificate key pair to a virtual server by using the configuration utility

1. Navigate to Traffic Management > SSL Offload > Virtual Servers.
2. Select the virtual server to which you want to bind the certificate key pair, for example, Vserver-SSL-1, and click Open.
3. In the Configure Virtual Server (SSL Offload) dialog box, on the SSL Settings tab, under Available, select the certificate key pair that you want to bind to the virtual server (for example, Certkey-SSL-1), and then click Add.
4. Click OK.
5. Verify that the certificate key pair that you selected appears in the Configured area.

Configuring Support for Outlook Web Access

If you use Outlook Web Access (OWA) servers on your NetScaler appliance, you must configure the appliance to insert a special header field, FRONT-END-HTTPS: ON, in HTTP requests directed to the OWA servers, so that the servers generate URL links as https:// instead of http://.

Note: You can enable OWA support for HTTP-based SSL virtual servers and services only. You cannot apply it for TCP-based SSL virtual servers and services.
To configure OWA support, do the following:

- Create an SSL action to enable OWA support.
- Create an SSL policy.
- Bind the policy to the SSL virtual server.

Creating an SSL Action to Enable OWA Support

Updated: 2013-06-24

Before you can enable Outlook Web Access (OWA) support, you must create an SSL action. SSL actions are bound to SSL policies and triggered when incoming data matches the rule specified by the policy.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To create an SSL action to enable OWA support by using the command line interface

At the command prompt, type the following commands to create an SSL action to enable OWA support and verify the configuration:

- add ssl action <name> -OWASupport ENABLED
- show SSL action <name>

```
> add ssl action Action-SSL-OWA -OWASupport enabled
Done
> show SSL action Action-SSL-OWA
  Name: Action-SSL-OWA
  Data Insertion Action: OWA
  Support: ENABLED
Done
```

To create an SSL action to enable OWA support by using the configuration utility

1. Navigate to Traffic Management > SSL > Policies.
2. In the details pane, on the Actions tab, click Add.
3. In the Create SSL Action dialog box, in the Name text box, type Action-SSL-OWA.
4. Under Outlook Web Access, select Enabled.
5. Click Create, and then click Close.
6. Verify that Action-SSL-OWA appears in the **SSL Actions** page.

Creating SSL Policies

Updated: 2013-09-04

SSL policies are created by using the policy infrastructure. Each SSL policy has an SSL action bound to it, and the action is carried out when incoming traffic matches the rule that has been configured in the policy.

To create an SSL policy by using the command line interface

At the command prompt, type the following commands to configure an SSL policy and verify the configuration:

- add ssl policy <name> -rule <expression> -reqAction <string>
- show ssl policy <name>

Example

```
> add ssl policy Policy-SSL-1 -rule ns_true -reqaction Action-SSL-OWA
Done
> show ssl policy Policy-SSL-1
Name: Policy-SSL-1      Rule: ns_true
Action: Action-SSL-OWA  Hits: 0
Policy is bound to following entities
1)     PRIORITY : 0
Done
```

To create an SSL policy by using the configuration utility

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Traffic Management > SSL > Policies.
2. In the details pane, click Add.
3. In the Create SSL Policy dialog box, in the Name text box, type the name of the SSL Policy (for example, Policy-SSL-1).
4. In Request Action, select the configured SSL action that you want to associate with this policy (for example, Action-SSL-OWA). The ns_true general expression applies the policy to all successful SSL handshake traffic. However, if you need to filter specific responses, you can create policies with a higher level of detail. For more information about configuring granular policy expressions, see "Understanding Policies and Expressions."
5. In Named Expressions, choose the built-in general expression ns_true and click Add Expression. The expression ns_true now appears in the Expression text box.
6. Click Create, and then click Close.
7. Verify that the policy is correctly configured by selecting the policy and viewing the Details section at the bottom of the pane.

Binding the SSL Policy to an SSL Virtual Server

Updated: 2013-06-24

After you configure an SSL policy for Outlook Web Access, bind the policy to a virtual server that will intercept incoming Outlook traffic. If the incoming data matches any of the rules configured in the SSL policy, the policy is triggered and the action associated with it is carried out.

To bind an SSL policy to an SSL virtual server by using the command line interface

At the command prompt, type the following commands to bind an SSL policy to an SSL virtual server and verify the configuration:

- bind ssl vserver <vServerName> -policyName <string>
- show ssl vserver <name>

Example

```
> bind ssl vserver Vserver-SSL-1 -policyName Policy-SSL-1
 Done
> show ssl vserver Vserver-SSL-1
      Advanced SSL configuration for VServer Vserver-SSL-1:
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 0
      Session Reuse: ENABLED        Timeout: 120 seconds
      Cipher Redirect: ENABLED
      SSLv2 Redirect: ENABLED
      ClearText Port: 0
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

  1)    CertKey Name: CertKey-SSL-1 Server Certificate

  1)    Policy Name: Policy-SSL-1
                  Priority: 0
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
1)     Cipher Name: DEFAULT
       Description: Predefined Cipher Alias
 Done
>
```

To bind an SSL policy to an SSL virtual server by using the configuration utility

1. Navigate to Traffic Management > SSL Offload > Virtual Servers.
2. In the details pane, select the virtual server (for example, Vserver-SSL-1), and then click Open.
3. In the Configure Virtual Server (SSL Offload) dialog box, click Insert Policy, and then select the policy that you want to bind to the SSL virtual server. Optionally, you can double-click the Priority field and type a new priority level.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Features at a Glance

Sep 04, 2013

Citrix NetScaler features can be configured independently or in combinations to address specific needs. Although some features fit more than one category, the numerous NetScaler features can generally be categorized as application switching and traffic management features, application acceleration features, and application security and firewall features, and an application visibility feature.

To understand the order in which the features perform their processing, see "Processing Order of Features."

This document includes the following:

- Application Switching and Traffic Management Features
- Application Acceleration Features
- Application Security and Firewall Features
- Application Visibility Feature
- Cloud Integration Feature

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Switching and Traffic Management Features

Aug 30, 2016

SSL Offloading

Transparently offloads SSL encryption and decryption from web servers, freeing server resources to service content requests. SSL places a heavy burden on an application's performance and can render many optimization measures ineffective. SSL offload and acceleration allow all the benefits of Citrix Request Switching technology to be applied to SSL traffic, ensuring secure delivery of web applications without degrading end-user performance.
For more information, see "SSL Offload and Acceleration."

Access Control Lists

Compares incoming packets to Access Control Lists (ACLs). If a packet matches an ACL rule, the action specified in the rule is applied to the packet. Otherwise, the default action (ALLOW) is applied and the packet is processed normally. For the appliance to compare incoming packets to the ACLs, you have to apply the ACLs. All ACLs are enabled by default, but you have to apply them in order for the NetScaler to compare incoming packets against them. If an ACL is not required to be a part of the lookup table, but still needs to be retained in the configuration, it should be disabled before the ACLs are applied. A NetScaler does not compare incoming packets to disabled ACLs.
For more information, see "Access Control List."

Load Balancing

Load balancing decisions are based on a variety of algorithms, including round robin, least connections, weighted least bandwidth, weighted least packets, minimum response time, and hashing based on URL, domain source IP, or destination IP. Both the TCP and UDP protocols are supported, so the NetScaler can load balance all traffic that uses those protocols as the underlying carrier (for example, HTTP, HTTPS, UDP, DNS, NNTP, and general firewall traffic). In addition, the NetScaler can maintain session persistence based on source IP, cookie, server, group, or SSL session. It allows users to apply custom Extended Content Verification (ECV) to servers, caches, firewalls and other infrastructure devices to ensure that these systems are functioning properly and are providing the right content to users. It can also perform health checks using ping, TCP, or HTTP URL, and the user can create monitors based on Perl scripts. To provide high-scale WAN optimization, the CloudBridge appliances deployed at data centers can be load balanced through NetScaler appliances. The bandwidth and number of concurrent sessions can be improved significantly.
For more information, see "Load Balancing."

Traffic Domains

Traffic domains provide a way to create logical ADC partitions within a single NetScaler appliance. They enable you to segment network traffic for different applications. You can use traffic domains to create multiple isolated environments whose resources do not interact with each other. An application belonging to a specific traffic domain communicates only with entities, and processes traffic, within that domain. Traffic belonging to one traffic domain cannot cross the boundary of another traffic domain. Therefore, you can use duplicate IP addresses on the appliance as long as an addresses is not duplicated within the same domain.
For more information, see "Traffic Domains."

Network Address Translation

Network address translation (NAT) involves modification of the source and/or destination IP addresses, and/or the TCP/UDP port numbers, of IP packets that pass through the NetScaler appliance. Enabling NAT on the appliance enhances

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the security of your private network, and protects it from a public network such as the Internet, by modifying your network's source IP addresses when data passes through the NetScaler.

The NetScaler appliance supports the following types of network address translation:

INAT—In Inbound NAT (INAT), an IP address (usually public) configured on the NetScaler appliance listens to connection requests on behalf of a server. For a request packet received by the appliance on a public IP address, the NetScaler replaces the destination IP address with the private IP address of the server. In other words, the appliance acts as a proxy between clients and the server. INAT configuration involves INAT rules, which define a 1:1 relationship between the IP address on the NetScaler appliance and the IP address of the server.

RNAT—In Reverse Network Address Translation (RNAT), for a session initiated by a server, the NetScaler appliance replaces the source IP address in the packets generated by the server with an IP address (type SNIP) configured on the appliance. The appliance thereby prevents exposure of the server's IP address in any of the packets generated by the server. An RNAT configuration involves an RNAT rule, which specifies a condition. The appliance performs RNAT processing on those packets that match the condition.

Stateless NAT46 Translation—Stateless NAT46 enables communication between IPv4 and IPv6 networks, by way of IPv4 to IPv6 packet translation and vice versa, without maintaining any session information on the NetScaler appliance. A stateless NAT46 configuration involves an IPv4-IPv6 INAT rule and an NAT46 IPv6 prefix.

Stateful NAT64 Translation—The stateful NAT64 feature enables communication between IPv4 clients and IPv6 servers through IPv6 to IPv4 packet translation, and vice versa, while maintaining session information on the NetScaler appliance. A stateful NAT64 configuration involves an NAT64 rule and an NAT64 IPv6 prefix.

For more information, see "Configuring Network Address Translation."

Multipath TCP Support

NetScaler appliances support Multipath TCP (MPTCP). MPTCP is a TCP/IP protocol extension that identifies and uses multiple paths available between hosts to maintain the TCP session. You must enable MPTCP on a TCP profile and bind it to a virtual server. When MPTCP is enabled, the virtual server functions as an MPTCP gateway and converts MPTCP connections with the clients to TCP connections that it maintains with the servers.
For more information, see "MPTCP (Multi-Path TCP)."

Content Switching

Determines the server to which to send the request on the basis of configured content switching policies. Policy rules can be based on the IP address, URL, and HTTP headers. This allows switching decisions to be based on user and device characteristics such as who the user is, what type of agent is being used, and what content the user requested.
For more information, see "Content Switching."

Global Server Load Balancing (GSLB)

Extends the traffic management capabilities of a NetScaler to include distributed Internet sites and global enterprises. Whether installations are spread across multiple network locations or multiple clusters in a single location, the NetScaler maintains availability and distributes traffic across them. It makes intelligent DNS decisions to prevent users from being sent to a site that is down or overloaded. When the proximity-based GSLB method is enabled, the NetScaler can make load balancing decisions based on the proximity of the client's local DNS server (LDNS) in relation to different sites. The main benefit of the proximity-based GSLB method is faster response time resulting from the selection of the closest available site.
For more information, see "Global Server Load Balancing."

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Dynamic Routing

Enables routers to obtain topology information, routes, and IP addresses from neighboring routers automatically. When dynamic routing is enabled, the corresponding routing process listens to route updates and advertises routes. The routing processes can also be placed in passive mode. Routing protocols enable an upstream router to load balance traffic to identical virtual servers hosted on two standalone NetScaler units using the Equal Cost Multipath technique.
For more information, see "Configuring Dynamic Routes."

Link Load Balancing

Load balances multiple WAN links and provides link failover, further optimizing network performance and ensuring business continuity. Ensures that network connections remain highly available, by applying intelligent traffic control and health checks to distribute traffic efficiently across upstream routers. Identifies the best WAN link to route both incoming and outbound traffic based on policies and network conditions, and protects applications against WAN or Internet link failure by providing rapid fault detection and failover.
For more information, see " Link Load Balancing."

TCP Optimization

You can use TCP profiles to optimize TCP traffic. TCP profiles define the way that NetScaler virtual servers process TCP traffic. Administrators can use the built-in TCP profiles or configure custom profiles. After defining a TCP profile, you can bind it to a single virtual server or to multiple virtual servers.
Some of the key optimization features that can be enabled by TCP profiles are:

- TCP keep-alive—Checks the operational status of the peers at specified time intervals to prevent the link from being broken.
- Selective Acknowledgment (SACK)— Improves the performance of data transmission, especially in long fat networks (LFNs).
- TCP window scaling— Allows efficient transfer of data over long fat networks (LFNs).

For more information on TCP Profiles, see "Configuring TCP Profiles."

Web Interface on NetScaler

Provides access to XenApp and XenDesktop resources, which include applications, content, and desktops. Users access resources through a standard Web browser or by using the Citrix XenApp plug-in. The Web Interface runs as a service on port 8080 on the NetScaler appliance. To create Web Interface sites, Java is executed on Apache Tomcat Web server version 6.0.26 on the NetScaler appliance.
Note: Web Interface is supported only on NetScaler nCore releases.
For more information, see "Web Interface."

CloudBridge Connector

The Citrix NetScaler CloudBridge Connector feature, a fundamental part of the Citrix OpenCloud framework, is a tool used to build a cloud-extended data center. The OpenCloud Bridge enables you to connect one or more NetScaler appliances or NetScaler virtual appliances on the cloud-to your network without reconfiguring your network. Cloud hosted applications appear as though they are running on one contiguous enterprise network. The primary purpose of the OpenCloud Bridge is to enable companies to move their applications to the cloud while reducing costs and the risk of application failure. In addition, the OpenCloud Bridge increases network security in cloud environments. An OpenCloud Bridge is a Layer-2 network bridge that connects a NetScaler appliance or NetScaler virtual appliance on a cloud instance to a NetScaler appliance or NetScaler virtual appliance on your LAN. The connection is made through a tunnel that uses the Generic Routing Encapsulation (GRE) protocol. The GRE protocol provides a mechanism for encapsulating packets from a wide variety of network protocols to be forwarded over another protocol. Then Internet Protocol security (IPsec) protocol suite is used to secure the communication between the peers in the OpenCloud Bridge.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

For more information, see "CloudBridge."

DataStream

The NetScaler DataStream feature provides an intelligent mechanism for request switching at the database layer by distributing requests on the basis of the SQL query being sent.

When deployed in front of database servers, a NetScaler ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and on the basis of database names, user names, character sets, and packet size.

You can configure load balancing to switch requests according to load balancing algorithms, or you can elaborate the switching criteria by configuring content switching to make a decision based on SQL query parameters, such as user name, database names, and command parameters. You can further configure monitors to track the states of database servers.

The advanced policy infrastructure on the NetScaler appliance includes expressions that you can use to evaluate and process the requests. The advanced expressions evaluate traffic associated with MySQL database servers. You can use request-based expressions (expressions that begin with MYSQL.CLIENT and MYSQL.REQ) in advanced policies to make request switching decisions at the content switching virtual server bind point and response-based expressions (expressions that begin with MYSQL.RES) to evaluate server responses to user-configured health monitors.

Note: DataStream is supported for MySQL and MS SQL databases.
For more information, see "DataStream."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Acceleration Features

Sep 06, 2013

AppCompress

Uses the gzip compression protocol to provide transparent compression for HTML and text files. The typical 4:1 compression ratio yields up to 50% reduction in bandwidth requirements out of the data center. It also results in significantly improved end-user response time, because it reduces the amount of data that must be delivered to the user's browser.

For more information, see "Compression."

Cache Redirection

Manages the flow of traffic to a reverse proxy, transparent proxy, or forward proxy cache farm. Inspects all requests, and identifies non-cacheable requests and sends them directly to the origin servers over persistent connections. By intelligently redirecting non-cacheable requests back to the origin web servers, the NetScaler appliance frees cache resources and increases cache hit rates while reducing overall bandwidth consumption and response delays for these requests.

For more information, see "Cache Redirection."

AppCache

Helps optimize web content and application data delivery by providing a fast in-memory HTTP/1.1 and HTTP/1.0 compliant web caching for both static and dynamic content. This on-board cache stores the results of incoming application requests even when an incoming request is secured or the data compressed, and then reuses the data to fulfill subsequent requests for the same information. By serving data directly from the on-board cache, the appliance can reduce page regeneration times by eliminating the need to funnel static and dynamic content requests to the server.

For more information, see "Integrated Caching."

TCP Buffering

Buffers the server's response and delivers it to the client at the client's speed, thus offloading the server faster and thereby improving the performance of web sites.

For more information, see "TCP Buffering."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Security and Firewall Features

Oct 30, 2013

Denial of Service Attack (DoS) Defense

Detects and stops malicious distributed denial-of-service (DDoS) attacks and other types of malicious attacks before they reach your servers, preventing them from affecting network and application performance. The NetScaler appliance identifies legitimate clients and elevates their priority, leaving suspect clients unable to consume a disproportionate percentage of resources and cripple your site. The appliance provides application-level protection from the following types of malicious attacks:

- SYN flood attacks
- Pipeline attacks
- Teardrop attacks
- Land attacks
- Fraggle attacks
- Zombie connection attacks

The appliance aggressively defends against these types of attacks by preventing the allocation of server resources for these connections. This insulates servers from the overwhelming flood of packets associated with these events.

The appliance also protects network resources from ICMP based attacks by using ICMP rate limiting and aggressive ICMP packet inspection. It performs strong IP reassembly, drops a variety of suspicious and malformed packets, and applies Access Control Lists (ACLs) to site traffic for further protection.

For more information, see "HTTP Denial-of-Service Protection."

Content Filtering

Provides protection from malicious attacks for web sites at the Layer 7 level. The appliance inspects each incoming request according to user-configured rules based on HTTP headers, and performs the action the user configured. Actions can include resetting the connection, dropping the request, or sending an error message to the user's browser. This allows the appliance to screen unwanted requests and reduces your servers' exposure to attacks.
This feature can also analyze HTTP GET and POST requests and filter out known bad signatures, allowing it to defend your servers against HTTP-based attacks.

For more information, see "Content Filtering."

Responder

Functions like an advanced filter and can be used to generate responses from the appliance to the client. Some common uses of this feature are generation of redirect responses, user defined responses, and resets.
For more information, see "Responder."

Rewrite

Modifies HTTP headers and body text. You can use the rewrite feature to add HTTP headers to an HTTP request or response, make modifications to individual HTTP headers, or delete HTTP headers. It also enables you to modify the HTTP body in requests and responses.
When the appliance receives a request or sends a response, it checks for rewrite rules, and if applicable rules exist, it applies them to the request or response before passing it on to the web server or client computer.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For more information, see "Rewrite."

Priority Queuing

Prioritizes user requests to ensure that the most important traffic is serviced first during surges in request volume. You can establish priority based on request URLs, cookies, or a variety of other factors. The appliance places requests in a three-tier queue based on their configured priority, enabling business-critical transactions to flow smoothly even during surges or site attacks.
For more information, see "Priority Queuing."

Surge Protection

Regulates the flow of user requests to servers and controls the number of users that can simultaneously access the resources on the servers, queuing any additional requests once your servers have reached their capacity. By controlling the rate at which connections can be established, the appliance blocks surges in requests from being passed on to your servers, thus preventing site overload.
For more information, see "Surge Protection."

NetScaler Gateway

NetScaler Gateway is a secure application access solution that provides administrators granular application-level policy and action controls to secure access to applications and data while allowing users to work from anywhere. It gives IT administrators a single point of control and tools to help ensure compliance with regulations and the highest levels of information security across and outside the enterprise. At the same time, it empowers users with a single point of access—optimized for roles, devices, and networks—to the enterprise applications and data they need. This unique combination of capabilities helps maximize the productivity of today's mobile workforce.

For more information, see "NetScaler Gateway."

Application Firewall

Protects applications from misuse by hackers and malware, such as cross site scripting attacks, buffer overflow attacks, SQL injection attacks, and forceful browsing, by filtering traffic between each protected web server and users that connect to any web site on that web server. The application firewall examines all traffic for evidence of attacks on web server security or misuse of web server resources, and takes the appropriate action to prevent these attacks from succeeding.
For more information, see "Application Firewall."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Visibility Feature

Sep 04, 2013

NetScaler Insight Center

NetScaler Insight Center is a high performance collector that provides end-to-end user experience visibility across Web and HDX (ICA) traffic. It collects HTTP and ICA AppFlow records generated by NetScaler ADC appliances and populates analytical reports covering Layer 3 to Layer 7 statistics. NetScaler Insight Center provides in-depth analysis for the last five minutes of real-time data, and for historical data collected for the last one hour, one day, one week, and one month.
HDX (ICA) analytic dashboard enables you to drill down from HDX Users, Applications, Desktops, and even from gateway-level information. Similarly, HTTP analytics provide a bird's eye view of Web Applications, URLs Accessed, Client IP Addresses and Server IP Addresses, and other dashboards. The administrator can drill down and identify the pain points from any of these dashboards, as appropriate for the use case.

EdgeSight for NetScaler

Support for application performance monitoring based on end user experience. This solution leverages the HTML injection feature to obtain various time values, which are used by EdgeSight server for analysis and report generation. EdgeSight for NetScaler provides a way to monitor the performance benefits of a NetScaler and determine potential bottlenecks in a network.
For more information, see "EdgeSight Monitoring for NetScaler."

Enhanced Application Visibility Using AppFlow

The Citrix NetScaler appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. AppFlow transmits this information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information.
Using UDP as the transport protocol, AppFlow transmits the collected data, called *flow records*, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor.

To limit the types of flows to monitor, by sampling and filtering the application traffic, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

For more information, see "AppFlow."

Stream Analytics

The performance of your web site or application depends on how well you optimize the delivery of the most frequently requested content. Techniques such as caching and compression help accelerate the delivery of services to clients, but you need to be able to identify the resources that are requested most frequently, and then cache or compress those resources. You can identify the most frequently used resources by aggregating real-time statistics about website or application traffic. Statistics such as how frequently a resource is accessed relative to other resources and how much bandwidth is consumed

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

by those resources help you determine whether those resources need to be cached or compressed to improve server performance and network utilization. Statistics such as response times and the number of concurrent connections to the application help you determine whether you must enhance server-side resources.

If the web site or application does not change frequently, you can use products that collect statistical data, and then manually analyze the statistics and optimize the delivery of content. However, if you do not want to perform manual optimizations, or if your web site or application is dynamic in nature, you need infrastructure that can not only collect statistical data but can also automatically optimize the delivery of resources on the basis of the statistics. On the NetScaler appliance, this functionality is provided by the Stream Analytics feature. The feature operates on a single NetScaler appliance and collects run-time statistics on the basis of criteria that you define. When used with NetScaler policies, the feature also provides you with the infrastructure that you need for automatic, real-time traffic optimization.

For more information, see "Stream Analytics."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cloud Integration Feature

Aug 23, 2013

AutoScale

All applications have specific usage patterns that comprise peaks and troughs. These load variations can be dynamic in nature and difficult to predict, given that they depend on several factors that are intrinsic to the use case. Cloud users have to constantly monitor the load on their application fleet and make sure that these variations have minimum impact on end users. During periods of peak usage, when the application fleet is overloaded and end users experience significant latency, they have to deploy additional application instances. During trough periods, the expanded fleet is underutilized. So they might have to remove additional instances or bear unnecessary cost overheads. In most cases, they have to perform these tasks manually.

If your organization uses Citrix CloudPlatform to deploy and manage the cloud environment, users can use the *AutoScale* feature in CloudPlatform, in conjunction with a Citrix NetScaler appliance, to automatically scale their applications as needed. The AutoScale feature is part of the elastic load balancing feature in CloudPlatform. A CloudPlatform user can use the AutoScale feature to specify thresholds for various conditions for automatically scaling the application fleet upward and downward. CloudPlatform, in turn, configures the NetScaler appliance (by using the NetScaler NITRO API) to load balance traffic to the application virtual machines (VMs), monitor application thresholds and performance, and trigger scale-up and scale-down actions to add or remove VMs to or from the application fleet.

As the NetScaler administrator, you do not have to perform any tasks for configuring AutoScale on the NetScaler appliance. However, you might have to be aware of certain prerequisites, and you might have to troubleshoot the configuration if issues arise in the AutoScale configuration. To troubleshoot the configuration, you have to be aware of how CloudPlatform works and what configuration CloudPlatform pushes to the NetScaler appliance. You also need a working knowledge of how to troubleshoot issues on a NetScaler appliance.

For more information about AutoScale, see "AutoScale: Automatic Scaling in the Citrix CloudPlatform Environment."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Getting Started with Citrix NetScaler VPX

Jul 11, 2016

Intended for system and network administrators who install and configure complex networking equipment, this section of the library describes initial setup and basic configuration of the NetScaler virtual appliance.

Understanding the NetScaler

The Citrix NetScaler product is an application switch that performs application-specific traffic analysis to intelligently distribute, optimize, and secure Layer 4-Layer 7 (L4–L7) network traffic for web applications. For example, a NetScaler bases load balancing decisions on individual HTTP requests instead of on long-lived TCP connections, so that the failure or slowdown of a server is managed much more quickly and with less disruption to clients. The NetScaler feature set can be broadly categorized as consisting of switching features, security and protection features, and server-farm optimization features.

Switching Features

When deployed in front of application servers, a NetScaler ensures optimal distribution of traffic by the way in which it directs client requests. Administrators can segment application traffic according to information in the body of an HTTP or TCP request, and on the basis of L4–L7 header information such as URL, application data type, or cookie. Numerous load balancing algorithms and extensive server health checks improve application availability by ensuring that client requests are directed to the appropriate servers.

Security and Protection Features

NetScaler security and protection features protect web applications from Application Layer attacks. A NetScaler allows legitimate client requests and can block malicious requests. It provides built-in defenses against denial-of-service (DoS) attacks and supports features that protect against legitimate surges in application traffic that would otherwise overwhelm the servers. An available built-in firewall protects web applications from Application Layer attacks, including buffer overflow exploits, SQL injection attempts, cross-site scripting attacks, and more. In addition, the firewall provides identity theft protection by securing confidential corporate information and sensitive customer data.

Optimization Features

Optimization features offload resource-intensive operations, such as Secure Sockets Layer (SSL) processing, data compression, client keep-alive, TCP buffering, and the caching of static and dynamic content from servers. This improves the performance of the servers in the server farm and therefore speeds up applications. A NetScaler supports several transparent TCP optimizations, which mitigate problems caused by high latency and congested network links, accelerating the delivery of applications while requiring no configuration changes to clients or servers.

Understanding Policies and Expressions

A policy defines specific details of traffic filtering and management on a NetScaler. It consists of two parts: the expression and the action. The expression defines the types of requests that the policy matches. The action tells the NetScaler what to do when a request matches the expression. As an example, the expression might be to match a specific URL pattern to a type of security attack, with the action being to drop or reset the connection. Each policy has a priority, and the priorities determine the order in which the policies are evaluated.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

When a NetScaler receives traffic, the appropriate policy list determines how to process the traffic. Each policy on the list contains one or more expressions, which together define the criteria that a connection must meet to match the policy.

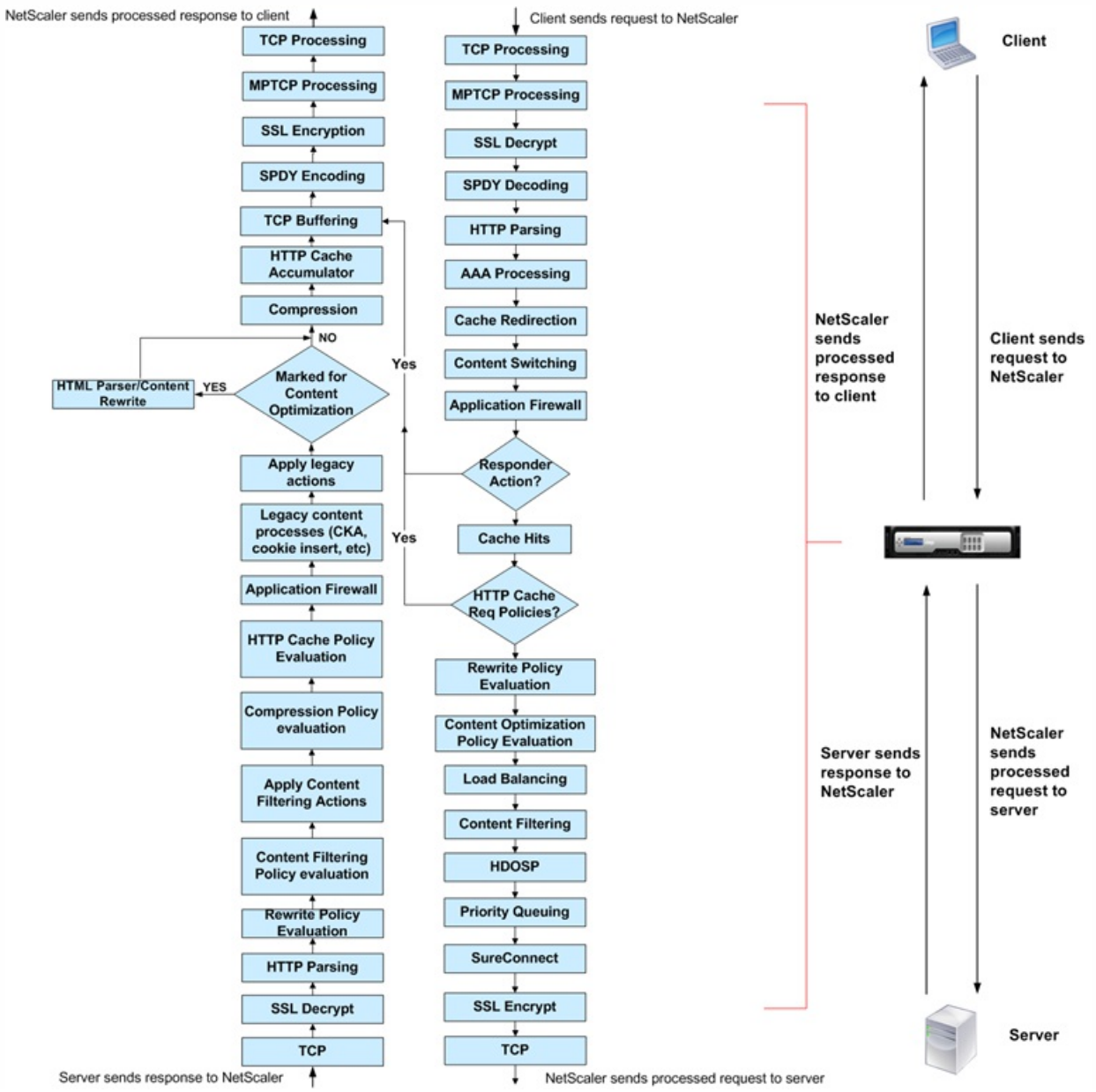
For all policy types except Rewrite policies, a NetScaler implements only the first policy that a request matches, not any additional policies that it might also match. For Rewrite policies, the NetScaler evaluates the policies in order and, in the case of multiple matches, performs the associated actions in that order. Policy priority is important for getting the results you want.

Processing Order of Features

Depending on requirements, you can choose to configure multiple features. For example, you might choose to configure both compression and SSL offload. As a result, an outgoing packet might be compressed and then encrypted before being sent to the client.

The following figure shows the L7 packet flow in the NetScaler.

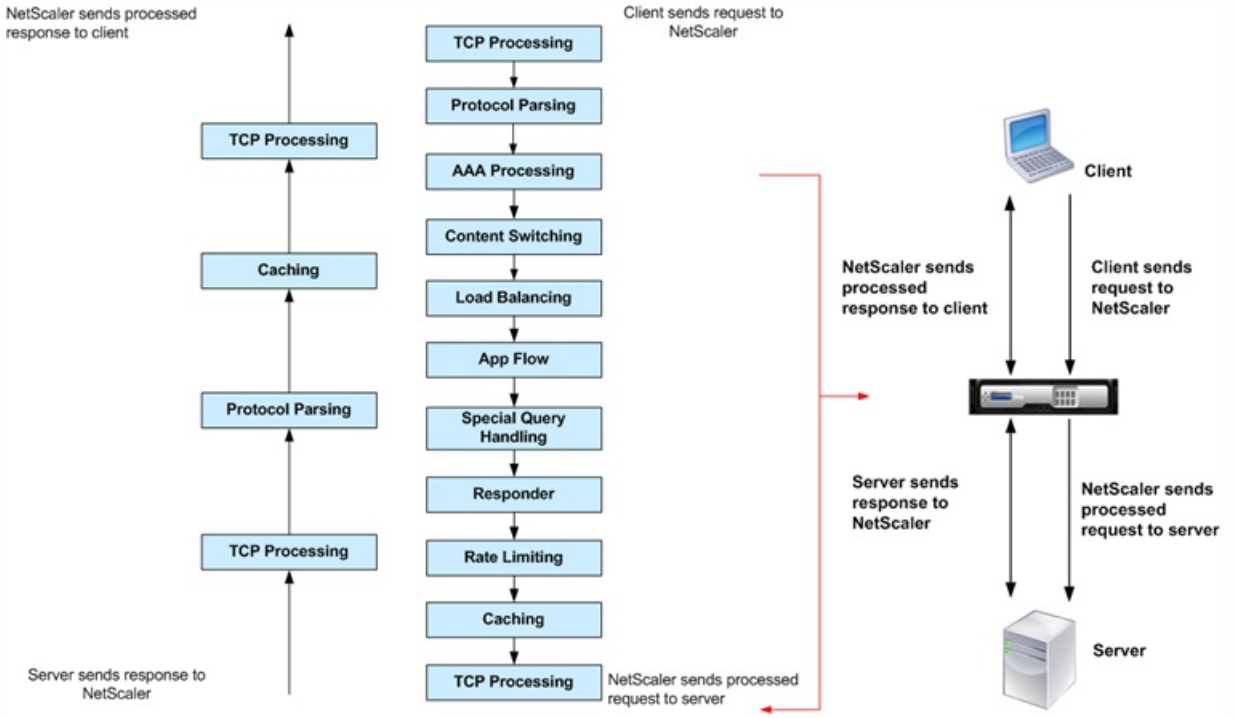
Figure 1. L7 Packet Flow Diagram



© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following figure shows the DataStream packet flow in the NetScaler. DataStream is supported for MySQL and MS SQL databases. For information about the DataStream feature, see "DataStream."

Figure 2. DataStream Packet Flow Diagram



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Citrix NetScaler Virtual Appliance Overview

Dec 27, 2016

The NetScaler virtual appliance product is a virtual NetScaler appliance that can be hosted on Citrix XenServer®, VMware ESX or ESXi, Linux-KVM, Microsoft Hyper-V and Azure, and Amazon Web Services (AWS) virtualization platforms.

For the VLAN tagging feature to work, do one of the following:

- On the VMware ESX, set the port group's VLAN ID to 1 - 4095 on the VSwitch of VMware ESX server. For more information about setting a VLAN ID on the VSwitch of VMware ESX server, see http://www.vmware.com/pdf/esx3_vlan_wp.pdf.

This overview covers only aspects that are unique to NetScaler virtual appliance. For an overview of NetScaler virtual appliance functionality, see Understanding the NetScaler.

Note: The terms *NetScaler*, *NetScaler appliance*, and *appliance* are used interchangeably with *NetScaler virtual appliance* unless stated otherwise.

NetScaler Virtual Appliance Setup for the XenServer Platform

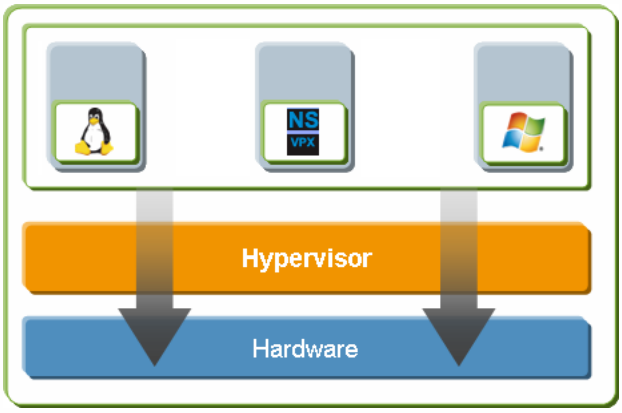
When you set up NetScaler virtual appliance on XenServer, you must use the XenCenter client to install the first NetScaler virtual appliance. Subsequent virtual appliances can be added by using either the XenCenter client or Citrix Command Center.

XenServer

The XenServer® product is a server virtualization platform that offers near bare-metal virtualization performance for virtualized server and client operating systems. XenServer uses the Xen® hypervisor to virtualize each server on which it is installed, enabling each server to host multiple virtual machines simultaneously.

The following figure shows the bare-metal solution architecture of NetScaler virtual appliance on XenServer.

Figure 1. NetScaler Virtual Appliance on XenServer



The bare-metal solution architecture has the following components:

Hardware or physical layer:

Physical hardware components including memory, CPU, network cards, and disk drives.

Xen hypervisor:

Thin layer of software that runs on top of the hardware. The Xen hypervisor gives each virtual machine a dedicated view of the hardware.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Virtual machine:

Operating system hosted on the hypervisor and appearing to the user as a separate physical computer. However, the machine shares physical resources with other virtual machines, and it is portable because the virtual machine is abstracted from physical hardware.

A NetScaler virtual machine, or *virtual appliance*, is installed on the Xen hypervisor and uses paravirtualized drivers to access storage and network resources. It appears to the users as an independent NetScaler appliance with its own network identity, user authorization and authentication capabilities, configuration, applications, and data. The paravirtualization technique enables the virtual machines and the hypervisor to work together to achieve high performance for I/O and for CPU and memory virtualization.

For more information about XenServer, see the XenServer documentation at http://support.citrix.com/product/xens/.

XenCenter

XenCenter® is a graphical virtualization-management interface for XenServer®, enabling you to manage servers, resource pools, and shared storage, and to deploy, manage, and monitor virtual machines from your Windows desktop machine.

Use XenCenter to install NetScaler virtual appliance on XenServer.

For more information about XenCenter, see the XenServer documentation at http://support.citrix.com/product/xens/.

Command Center

Command Center is a management and monitoring solution for Citrix application networking products that include NetScaler, NetScaler virtual appliance, NetScaler Gateway Enterprise Edition, Citrix® Branch Repeater™, Branch Repeater VPX™, and Citrix Repeater™. Command Center enables network administrators and operations teams to manage, monitor, and troubleshoot the entire global application delivery infrastructure from a single, unified console.

This centralized management solution simplifies operations by providing administrators with enterprise-wide visibility and automating management tasks that need to be executed across multiple devices.

Command Center is available with Citrix NetScaler Enterprise and Platinum editions.

You can use Command Center to provision NetScaler virtual appliance on XenServer, and then you can manage and monitor the virtual appliances from Command Center.

Note: You must use the XenCenter client to manage XenServer. You cannot manage XenServer from Command Center. For more information about Command Center, see the Command Center documentation.

NetScaler Virtual Appliance Setup for the VMware ESX Platform

Updated: 2013-08-23

The NetScaler virtual appliance setup for the VMware ESX platform requires a VMware ESX or ESXi server and the vSphere client.

VMware ESX and ESXi are virtualization products based on bare-metal architecture, offered by VMware, Inc. Citrix NetScaler virtual appliance can be hosted on a VMware ESX or ESXi server.

For more information about VMware ESX, see http://www.vmware.com/.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The vSphere client is a graphical interface for managing virtual machines on VMware ESX servers. You use the vSphere client to allocate resources on the ESX server to virtual appliances installed on the server or to deallocate resources. For example, you can allocate virtual network ports to a virtual appliance.

For more information about VMware vSphere client, see http://www.vmware.com/.

NetScaler Virtual Appliance Setup for the Microsoft Hyper-V Platform

The NetScaler virtual appliance setup for the Microsoft Hyper-V platform requires Windows Server with the Hyper-V role installed. Like all virtualization systems, Hyper-V enables you to create a virtualized computing environment that results in better utilization of your hardware resources.
Hyper-V is a type 1 hypervisor that comes preinstalled with Windows Server, and it needs to be enabled as a role on the Windows Server.

For more information about Hyper-V, see http://technet.microsoft.com/en-us/library/cc816638(WS.10).aspx.

NetScaler Virtual Appliance Setup for Linux-KVM Platform

Updated: 2014-03-07

The NetScaler® VPX™ is a virtual NetScaler appliance that can be hosted on a kernel based Virtualization Machine(KVM). The host Linux operating system must be installed on suitable hardware by using virtualization tools such as KVM Module and QEMU. NetScaler VPX runs as a virtual appliance on Linux-KVM server. Like all virtualization systems, KVM enables you to create a virtualized computing environment that results in better utilization of your hardware resources.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Where Does a NetScaler Appliance Fit in the Network?

Jun 24, 2013

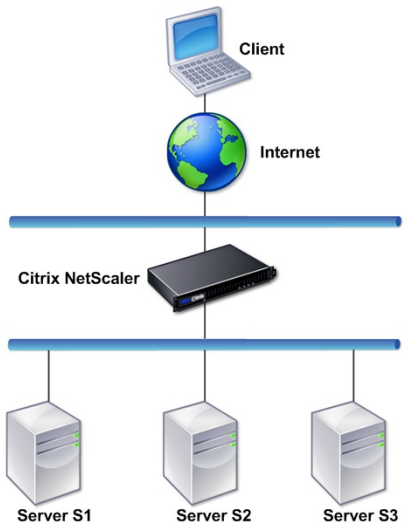
A NetScaler appliance resides between the clients and the servers, so that client requests and server responses pass through it. In a typical installation, virtual servers configured on the appliance provide connection points that clients use to access the applications behind the appliance. In this case, the appliance owns public IP addresses that are associated with its virtual servers, while the real servers are isolated in a private network. It is also possible to operate the appliance in a transparent mode as an L2 bridge or L3 router, or even to combine aspects of these and other modes.

Physical Deployment Modes

Updated: 2013-09-04

A NetScaler appliance logically residing between clients and servers can be deployed in either of two physical modes: inline and one-arm. In inline mode, multiple network interfaces are connected to different Ethernet segments, and the appliance is placed between the clients and the servers. The appliance has a separate network interface to each client network and a separate network interface to each server network. The appliance and the servers can exist on different subnets in this configuration. It is possible for the servers to be in a public network and the clients to directly access the servers through the appliance, with the appliance transparently applying the L4-L7 features. Usually, virtual servers (described later) are configured to provide an abstraction of the real servers. The following figure shows a typical inline deployment.

Figure 1. Inline Deployment



In one-arm mode, only one network interface of the appliance is connected to an Ethernet segment. The appliance in this case does not isolate the client and server sides of the network, but provides access to applications through configured virtual servers. One-arm mode can simplify network changes needed for NetScaler installation in some environments.

For examples of inline (two-arm) and one-arm deployment, see "Understanding Common Network Topologies."

Citrix NetScaler as an L2 Device

Updated: 2013-09-04

© 1999-2017 Citrix Systems, Inc. All rights reserved.

A NetScaler functioning as an L2 device is said to operate in L2 mode. In L2 mode, the NetScaler forwards packets between network interfaces when all of the following conditions are met:

- The packets are destined to another device's media access control (MAC) address.
- The destination MAC address is on a different network interface.
- The network interface is a member of the same virtual LAN (VLAN).

By default, all network interfaces are members of a pre-defined VLAN, VLAN 1. Address Resolution Protocol (ARP) requests and responses are forwarded to all network interfaces that are members of the same VLAN. To avoid bridging loops, L2 mode must be disabled if another L2 device is working in parallel with the NetScaler.

For information about how the L2 and L3 modes interact, see "Configuring Modes of Packet Forwarding."

For information about configuring L2 mode, see "Enabling and Disabling Layer 2 Mode."

Citrix NetScaler as a Packet Forwarding Device

Updated: 2014-03-14

A NetScaler appliance can function as a packet forwarding device, and this mode of operation is called L3 mode. With L3 mode enabled, the appliance forwards any received unicast packets that are destined for an IP address that does not belong to the appliance, if there is a route to the destination. The appliance can also route packets between VLANs.

In both modes of operation, L2 and L3, the appliance generally drops packets that are in:

- Multicast frames
- Unknown protocol frames destined for an appliance's MAC address (non-IP and non-ARP)
- Spanning Tree protocol (unless BridgeBPDUs is ON)

For information about how the L2 and L3 modes interact, see "Configuring Modes of Packet Forwarding."

For information about configuring the L3 mode, see "Enabling and Disabling Layer 3 Mode."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How a NetScaler Communicates with Clients and Servers

Nov 10, 2016

A NetScaler appliance is usually deployed in front of a server farm and functions as a transparent TCP proxy between clients and servers, without requiring any client-side configuration. This basic mode of operation is called Request Switching technology and is the core of NetScaler functionality. Request Switching enables an appliance to multiplex and offload the TCP connections, maintain persistent connections, and manage traffic at the request (application layer) level. This is possible because the appliance can separate the HTTP request from the TCP connection on which the request is delivered.

Depending on the configuration, an appliance might process the traffic before forwarding the request to a server. For example, if the client attempts to access a secure application on the server, the appliance might perform the necessary SSL processing before sending traffic to the server.

To facilitate efficient and secure access to server resources, an appliance uses a set of IP addresses collectively known as NetScaler-owned IP addresses. To manage your network traffic, you assign NetScaler-owned IP addresses to virtual entities that become the building blocks of your configuration. For example, to configure load balancing, you create virtual servers to receive client requests and distribute them to services, which are entities representing the applications on your servers.

Understanding NetScaler-Owned IP Addresses

To function as a proxy, a NetScaler appliance uses a variety of IP addresses. The key NetScaler-owned IP addresses are:

NetScaler IP (NSIP) address

The NSIP address is the IP address for management and general system access to the appliance itself, and for communication between appliances in a high availability configuration.

Virtual server IP (VIP) address

A VIP address is the IP address associated with a virtual server. It is the public IP address to which clients connect. An appliance managing a wide range of traffic may have many VIPs configured.

Subnet IP (SNIP) address

A SNIP address is used in connection management and server monitoring. You can specify multiple SNIP addresses for each subnet. SNIP addresses can be bound to a VLAN.

IP Set

An IP set is a set of IP addresses, which are configured on the appliance as SNIP . An IP set is identified with a meaningful name that helps in identifying the usage of the IP addresses contained in it.

Net Profile

A net profile (or network profile) contains an IP address or an IP set. A net profile can be bound to load balancing or content switching virtual servers, services, service groups, or monitors. During communication with physical servers or peers, the appliance uses the addresses specified in the profile as source IP addresses.

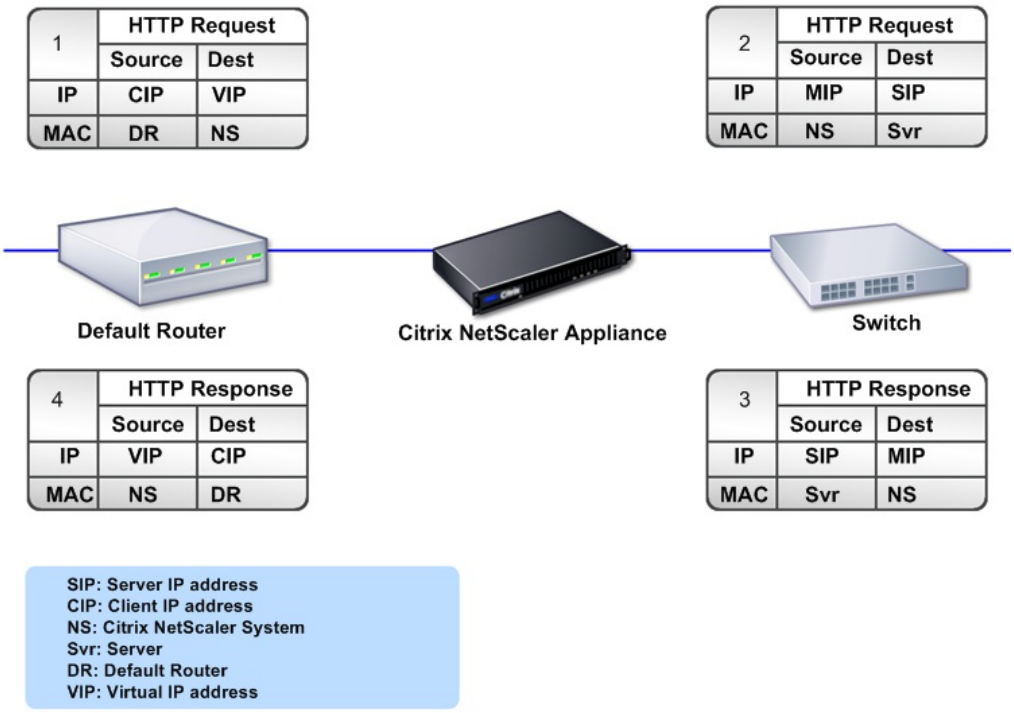
How Traffic Flows Are Managed

Because a NetScaler appliance functions as a TCP proxy, it translates IP addresses before sending packets to a server. When you configure a virtual server, clients connect to a VIP address on the NetScaler instead of directly connecting to a

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

server. As determined by the settings on the virtual server, the appliance selects an appropriate server and sends the client's request to that server. By default, the appliance uses a SNIP address to establish connections with the server, as shown in the following figure.

Figure 1. Virtual Server Based Connections



In the absence of a virtual server, when an appliance receives a request, it transparently forwards the request to the server. This is called the transparent mode of operation. When operating in transparent mode, an appliance translates the source IP addresses of incoming client requests to the SNIP address but does not change the destination IP address. For this mode to work, L2 or L3 mode has to be configured appropriately.

For cases in which the servers need the actual client IP address, the appliance can be configured to modify the HTTP header by inserting the client IP address as an additional field, or configured to use the client IP address instead of a SNIP address for connections to the servers.

Traffic Management Building Blocks

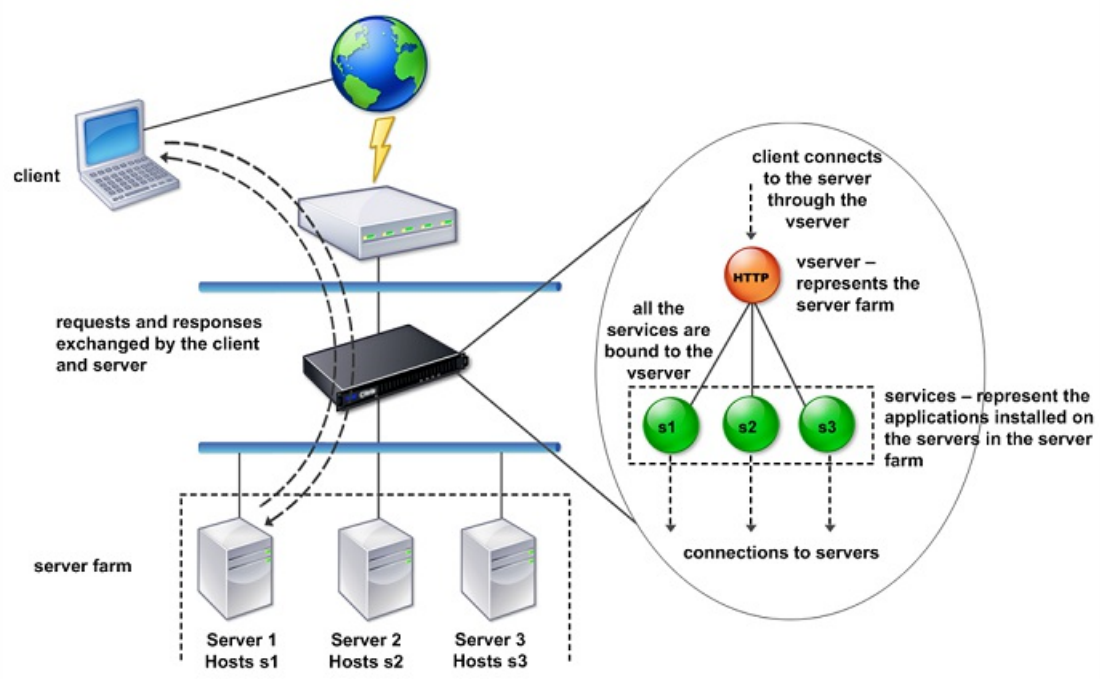
The configuration of a NetScaler appliance is typically built up with a series of virtual entities that serve as building blocks for traffic management. The building block approach helps separate traffic flows. Virtual entities are abstractions, typically representing IP addresses, ports, and protocol handlers for processing traffic. Clients access applications and resources through these virtual entities. The most commonly used entities are virtual servers and services. Virtual servers represent

© 1999-2017 Citrix Systems, Inc. All rights reserved.

groups of servers in a server farm or remote network, and services represent specific applications on each server.

Most features and traffic settings are enabled through virtual entities. For example, you can configure an appliance to compress all server responses to a client that is connected to the server farm through a particular virtual server. To configure the appliance for a particular environment, you need to identify the appropriate features and then choose the right mix of virtual entities to deliver them. Most features are delivered through a cascade of virtual entities that are bound to each other. In this case, the virtual entities are like blocks being assembled into the final structure of a delivered application. You can add, remove, modify, bind, enable, and disable the virtual entities to configure the features. The following figure shows the concepts covered in this section.

Figure 2. How Traffic Management Building Blocks Work



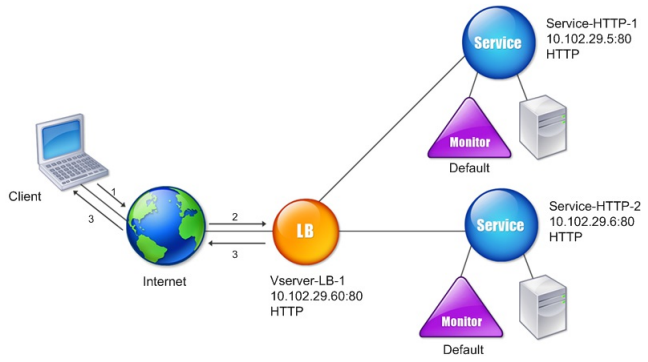
A Simple Load Balancing Configuration

In the example shown in the following figure, the NetScaler appliance is configured to function as a load balancer. For this configuration, you need to configure virtual entities specific to load balancing and bind them in a specific order. As a load balancer, an appliance distributes client requests across several servers and thus optimizes the utilization of resources.

The basic building blocks of a typical load balancing configuration are services and load balancing virtual servers. The services represent the applications on the servers. The virtual servers abstract the servers by providing a single IP address to which the clients connect. To ensure that client requests are sent to a server, you need to bind each service to a virtual server. That is, you must create services for every server and bind the services to a virtual server. Clients use the VIP address to connect to a NetScaler appliance. When the appliance receives client requests sent to the VIP address, it sends them to a server determined by the load balancing algorithm. Load balancing uses a virtual entity called a monitor to track whether a specific configured service (server plus application) is available to receive requests.

Figure 3. Load Balancing Virtual Server, Services, and Monitors

© 1999-2017 Citrix Systems, Inc. All rights reserved.



In addition to configuring the load balancing algorithm, you can configure several parameters that affect the behavior and performance of the load balancing configuration. For example, you can configure the virtual server to maintain persistence based on source IP address. The appliance then directs all requests from any specific IP address to the same server.

Understanding Virtual Servers

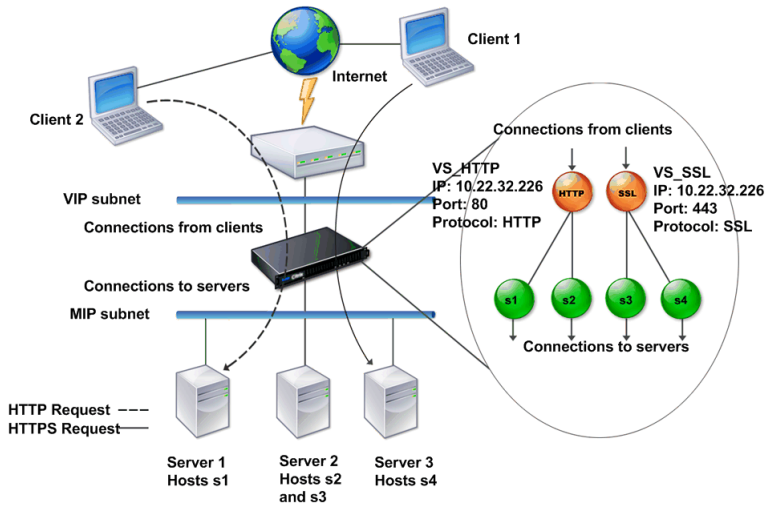
A virtual server is a named NetScaler entity that external clients can use to access applications hosted on the servers. It is represented by an alphanumeric name, virtual IP (VIP) address, port, and protocol. The name of the virtual server is of only local significance and is designed to make the virtual server easier to identify. When a client attempts to access applications on a server, it sends a request to the VIP instead of the IP address of the physical server. When the appliance receives a request at the VIP address, it terminates the connection at the virtual server and uses its own connection with the server on behalf of the client. The port and protocol settings of the virtual server determine the applications that the virtual server represents. For example, a web server can be represented by a virtual server and a service whose port and protocol are set to 80 and HTTP, respectively. Multiple virtual servers can use the same VIP address but different protocols and ports.

Virtual servers are points for delivering features. Most features, like compression, caching, and SSL offload, are normally enabled on a virtual server. When the appliance receives a request at a VIP address, it chooses the appropriate virtual server by the port on which the request was received and its protocol. The appliance then processes the request as appropriate for the features configured on the virtual server.

In most cases, virtual servers work in tandem with services. You can bind multiple services to a virtual server. These services represent the applications running on physical servers in a server farm. After the appliance processes requests received at a VIP address, it forwards them to the servers as determined by the load balancing algorithm configured on the virtual server. The following figure illustrates these concepts.

Figure 4. Multiple Virtual Servers with a Single VIP Address

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The preceding figure shows a configuration consisting of two virtual servers with a common VIP address but different ports and protocols. Each of the virtual servers has two services bound to it. The services s1 and s2 are bound to VS_HTTP and represent the HTTP applications on Server 1 and Server 2. The services s3 and s4 are bound to VS_SSL and represent the SSL applications on Server 2 and Server 3 (Server 2 provides both HTTP and SSL applications). When the appliance receives an HTTP request at the VIP address, it processes the request as specified by the settings of VS_HTTP and sends it to either Server 1 or Server 2. Similarly, when the appliance receives an HTTPS request at the VIP address, it processes it as specified by the settings of VS_SSL and it sends it to either Server 2 or Server 3.

Virtual servers are not always represented by specific IP addresses, port numbers, or protocols. They can be represented by wildcards, in which case they are known as wildcard virtual servers. For example, when you configure a virtual server with a wildcard instead of a VIP, but with a specific port number, the appliance intercepts and processes all traffic conforming to that protocol and destined for the predefined port. For virtual servers with wildcards instead of VIPs and port numbers, the appliance intercepts and processes all traffic conforming to the protocol.

Virtual servers can be grouped into the following categories:

Load balancing virtual server

Receives and redirects requests to an appropriate server. Choice of the appropriate server is based on which of the various load balancing methods the user configures.

Cache redirection virtual server

Redirects client requests for dynamic content to origin servers, and requests for static content to cache servers. Cache redirection virtual servers often work in conjunction with load balancing virtual servers.

Content switching virtual server

Directs traffic to a server on the basis of the content that the client has requested. For example, you can create a content switching virtual server that directs all client requests for images to a server that serves images only. Content switching virtual servers often work in conjunction with load balancing virtual servers.

Virtual private network (VPN) virtual server

Decrypts tunneled traffic and sends it to intranet applications.

SSL virtual server

Receives and decrypts SSL traffic, and then redirects to an appropriate server. Choosing the appropriate server is similar to

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

choosing a load balancing virtual server.

Understanding Services

Services represent applications on a server. While services are normally combined with virtual servers, in the absence of a virtual server, a service can still manage application-specific traffic. For example, you can create an HTTP service on a NetScaler appliance to represent a web server application. When the client attempts to access a web site hosted on the web server, the appliance intercepts the HTTP requests and creates a transparent connection with the web server.

In service-only mode, an appliance functions as a proxy. It terminates client connections, uses a SNIP address to establish a connection to the server, and translates the source IP addresses of incoming client requests to a SNIP address. Although the clients send requests directly to the IP address of the server, the server sees them as coming from the SNIP address. The appliance translates the IP addresses, port numbers, and sequence numbers.

A service is also a point for applying features. Consider the example of SSL acceleration. To use this feature, you must create an SSL service and bind an SSL certificate to the service. When the appliance receives an HTTPS request, it decrypts the traffic and sends it, in clear text, to the server. Only a limited set of features can be configured in the service-only case.

Services use entities called monitors to track the health of applications. Every service has a default monitor, which is based on the service type, bound to it. As specified by the settings configured on the monitor, the appliance sends probes to the application at regular intervals to determine its state. If the probes fail, the appliance marks the service as down. In such cases, the appliance responds to client requests with an appropriate error message or re-routes the request as determined by the configured load balancing policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Supported Hypervisors, Features, and Limitations

Jan 05, 2017

Table 1 lists the different hypervisors, and table 2 lists the different VPX features and limitations for the different hypervisors supported on a NetScaler virtual appliance.

Table 3 lists the supported web browsers that allows you access the NetScaler GUI and Dashboard. Besides a supported web browser, you must have version 1.6 or above of the Java applet plug-in installed.

Table 1. Hypervisors Supported on a NetScaler Virtual Appliance

	VPX on XenServer	VPX on VMware ESX	VPX on Microsoft Hyper-V	VPX on generic KVM	VPX on AWS	VPX on Azure
Hypervisor Version	6.2, 6.5	5.5 (build number: 3568722) 6.0 (build number: 3620759)	2012, 2012R2	RHEL 7.2, Ubuntu 15	N/A	N/A
SysID	450000	450010	450020	450070	450040	450020
Models	VPX 10 VPX 25 VPX 200 VPX 1000 VPX 3000 VPX 5000 VPX 8000	VPX 10 VPX 25 VPX 200 VPX 1000 VPX 3000 VPX 5000 VPX 8000 VPX 10G VPX 15G VPX 25G VPX 40G	VPX 10 VPX 25 VPX 200 VPX 1000 VPX 3000 VPX 8000	VPX 10 VPX 25 VPX 200 VPX 1000 VPX 3000 VPX 5000 VPX 8000	VPX 10 VPX 200 VPX 1000 VPX BYOL	VPX 10 VPX 200 VPX 1000 VPX BYOL

Table 2. VPX Feature Matrix

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	VPX on XenServer	VPX on VMware ESX			VPX on Microsoft Hyper-V	VPX on generic KVM	VPX on AWS	VPX on Azure
Features	PV	PV	SRIOV	Emulated	PV	PV		
Multi-PE Support	√	√	√	√	√	√	√	√
Clustering Support	√	√	√[1]	√	√	√	X	X
VLAN Tagging	√	√	√	√	√ ((Only on 2012R2))	√	X	X
Detecting Link Events	X[2]	X[2]	√[3]	X[2]	X[2]	X[2]	X	X
Interface Parameter Configuration	X	X	X	X	X	X	X	X
Static LA	√[2]	√[2]	X	√[2]	√[2]	√[2]	X	X
LACP	X	√[2]	X	√[2]	X	√[2]	X	X
Static CLAG	X	X	X	X	X	X	X	X
LACP CLAG	X	√[2]	X	√[2]	X	√[2]	X	X

[1] Clustering support is available on SRIOV for client- and server-facing interfaces and not for the backplane.

[2]Interface DOWN events are not recorded in VPX.

In case of Static LA, traffic might still be sent on the interface whose physical status is DOWN.

In case of LACP, peer device knows interface DOWN event based on LACP timeout mechanism.

Short timeout: 3 seconds

Long timeout: 90 seconds

For LACP, interfaces should not be shared across VMs.

In case of Dynamic routing, convergence time will depend on the Routing Protocol since link events are not detected.

Monitored static Route functionality will fail if monitors are not bound to static routes since Route state is dependent on the VLAN status. The VLAN status is dependent on the link status.

Partial failure detection will not happen in HA in case of link failure.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HA-Split brain condition might happen in case of link failure.

[3]When any link event (disable/enable, reset) is generated from a NetScaler appliance, the physical status of the link does not change. In case of static LA, any traffic initiated by the peer gets dropped on the NetScaler appliance.

In case of LACP, peer device knows interface DOWN event based on LACP timeout mechanism.

Short timeout: 3 seconds

Long timeout: 90 seconds

For LACP, interfaces should not be shared across VMs.

Table 3. Supported Browsers

Operating System	Browser	Versions
Windows 7	Internet Explorer	8, 9, and 10
	Mozilla Firefox	3.6.25 and above
	Google Chrome	15 and above
Windows 64 bit	Internet Explorer	8 and 9
	Google Chrome	15 and above
MAC	Mozilla Firefox	12 and above
	Safari	5.1.3
	Google Chrome	15 and above

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing NetScaler Virtual Appliances on XenServer

Aug 23, 2013

To install NetScaler virtual appliances on Citrix XenServer, you must first install XenServer on a machine with adequate system resources. To perform the NetScaler virtual appliance installation, you use Citrix XenCenter, which must be installed on a remote machine that can connect to the XenServer host through the network.

Note: After the initial configuration of the NetScaler appliance, if you want to upgrade the appliance to the latest software release, see Upgrading or Downgrading the System Software.

Prerequisites for Installing NetScaler Virtual Appliances on XenServer

Before you begin installing a virtual appliance, do the following:

- Install XenServer® version 6.0 or later on hardware that meets the minimum requirements.
- Install XenCenter® on a management workstation that meets the minimum system requirements.
- Obtain virtual appliance license files. For more information about virtual appliance licenses, see the *NetScaler VPX Licensing Guide* at http://support.citrix.com/article/ctx122426.

XenServer Hardware Requirements

The following table describes the minimum hardware requirements for a XenServer platform running NetScaler.

Table 1. Minimum System Requirements for XenServer Running NetScaler nCore virtual appliance

Component	Requirement
CPU	2 or more 64-bit x86 CPUs with virtualization assist (Intel-VT or AMD-V) enabled Note: To run NetScaler virtual appliance, hardware support for virtualization must be enabled on the XenServer host. Make sure that the BIOS option for virtualization support is not disabled. Consult your BIOS documentation for more details.
RAM	3 gigabytes (GB)
Disk space	Locally attached storage (PATA, SATA, SCSI) with 40 GB of disk space Note: XenServer installation creates a 4 GB partition for the XenServer host control domain; the remaining space is available for NetScaler virtual appliance and other virtual machines.
Network Interface Card (NIC)	One 1-Gbps NIC Recommended: Two 1-Gbps NICs

For information about installing XenServer, see the XenServer documentation at http://support.citrix.com/product/xens/.

The following table lists the virtual computing resources that XenServer must provide for each NetScaler nCore virtual

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appliance .

Table 2. Minimum Virtual Computing Resources Required for Running NetScaler ncore virtual appliance

Component	Requirement
Memory	2 GB
Virtual CPU (VCPU)	2
Virtual network interfaces	2

Note: For production use of NetScaler virtual appliance, Citrix recommends that CPU priority (in virtual machine properties) be set to the highest level, in order to improve scheduling behavior and network latency.

XenCenter System Requirements

XenCenter® is a Windows client application. It cannot run on the same machine as the XenServer® host. The following table describes the minimum system requirements.

Table 3. Minimum System Requirements for XenCenter Installation

Component	Requirement
Operating system	Windows 7, Windows XP, Windows Server 2003, or Windows Vista
.NET framework	Version 2.0 or later
CPU	750 megahertz (MHz) Recommended: 1 gigahertz (GHz) or faster
RAM	1 GB Recommended: 2 GB
Network Interface Card (NIC)	100 megabits per second (Mbps) or faster NIC

For information about installing XenCenter, see the XenServer documentation at http://support.citrix.com/product/xens/.

Installing NetScaler Virtual Appliances on XenServer by Using XenCenter

Updated: 2013-08-23

After you have installed and configured XenServer and XenCenter, you can use XenCenter to install virtual appliances on XenServer. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running XenServer.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

After you have used XenCenter to install the initial NetScaler virtual appliance (.xva image) on XenServer, you have the option to use Command Center to provision NetScaler virtual appliance. For more information, see the Command Center documentation.

To install NetScaler virtual appliances on XenServer by using XenCenter

1. Start XenCenter on your workstation.
2. On the Server menu, click Add.
3. In the Add New Server dialog box, in the Hostname text box, type the IP address or DNS name of the XenServer that you want to connect to.
4. In the User Name and Password text boxes, type the administrator credentials, and then click Connect. The XenServer name appears in the navigation pane with a green circle, which indicates that the XenServer is connected.
5. In the navigation pane, click the name of the XenServer on which you want to install NetScaler virtual appliance.
6. On the VM menu, click Import.
7. In the Import dialog box, in Import file name, browse to the location at which you saved the NetScaler virtual appliance .xva image file. Make sure that the Exported VM option is selected, and then click Next.
8. Select the XenServer on which you want to install the virtual appliance, and then click Next.
9. Select the local storage repository in which to store the virtual appliance, and then click Import to begin the import process.
10. You can add, modify, or delete virtual network interfaces as required. When finished, click Next.
11. Click Finish to complete the import process.
    Note: To view the status of the import process, click the **Log** tab.
12. If you want to install another virtual appliance, repeat steps 5 through 11.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing NetScaler Virtual Appliances on VMware ESX

Jul 27, 2017

Important: You cannot install standard VMware Tools or upgrade the VMware Tools version available on a NetScaler virtual appliance. VMware Tools for a NetScaler virtual appliance are delivered as part of the NetScaler software release.
Before installing NetScaler virtual appliances on VMware ESX, make sure that VMware ESX Server is installed on a machine with adequate system resources. To install NetScaler virtual appliances on VMware ESXi version 4.1 or later, you use VMware vSphere client. The client or tool must be installed on a remote machine that can connect to VMware ESX through the network.

After the installation, you can use vSphere client or vSphere Web Client to manage virtual appliances on VMware ESX 5.1 or later release.

Note:
The VMware vSphere client shows the guest operating system as "Sun Solaris 10" for NetScaler virtual machine. This is by design because VMware ESXi does not recognize FreeBSD.

After the initial configuration of the NetScaler appliance, if you want to upgrade the appliance to the latest software release, see "Upgrading or Downgrading the System Software."

Prerequisites for Installing NetScaler Virtual Appliances on VMware

Before you begin installing a virtual appliance, do the following:

- Install VMware ESX version 5.1 or later on hardware that meets the minimum requirements.
- Install VMware Client on a management workstation that meets the minimum system requirements.
- Download the NetScaler virtual appliance setup files.
- Label the physical network ports of VMware ESX.
- Obtain NetScaler license files. For more information about NetScaler virtual appliance licenses, see the *NetScaler VPX Licensing Guide* at http://support.citrix.com/article/ctx131110.

VMware ESX Hardware Requirements

The following table describes the minimum system requirements for VMware ESX servers running NetScaler ncore virtual appliance.

Table 1. Minimum System Requirements for VMware ESX Servers Running NetScaler nCore virtual appliance

Component	Requirement
CPU	2 or more 64-bit x86 CPUs with virtualization assist (Intel-VT or AMD-V) enabled Note: To run NetScaler virtual appliance, hardware support for virtualization must be enabled on the VMware ESX host. Make sure that the BIOS option for virtualization support is not disabled. For more information, see your BIOS documentation.
RAM	3 GB
Disk space	40 GB of disk space available

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Component	Requirement
Network	One 1-Gbps NIC; Two 1-Gbps NICs recommended Supported network interfaces are: • E1000 • VMXNET3 • SR-IOV

For information about installing VMware ESX, see http://www.vmware.com/.

The following table lists the virtual computing resources that the VMware ESX server must provide for each NetScaler ncore virtual appliance.

Table 2. Minimum Virtual Computing Resources Required for Running NetScaler ncore virtual appliance

Component	Requirement
Memory	2 GB
Virtual CPU (VCPU)	2
Virtual network interfaces	1 Note: With ESX 5.1 or later, you can install a maximum of 10 virtual network interfaces if the VPX hardware is upgraded version to 7 or higher.
Disk space	20 GB Note: This is in addition to any disk requirements for the hypervisor.

Note: For production use of NetScaler virtual appliance, the full memory allocation must be reserved. CPU cycles (in MHz) equal to at least the speed of one CPU core of the ESX should also be reserved.

VMware vSphere Client System Requirements

VMware vSphere is a client application that can run on Windows and Linux operating systems. It cannot run on the same machine as the VMware ESX server. The following table describes the minimum system requirements.

Table 3. Minimum System Requirements for VMware vSphere Client Installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "vSphere Compatibility Matrixes" PDF file at http://kb.vmware.com/.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Component	Requirement
CPU	750 megahertz (MHz); 1 gigahertz (GHz) or faster recommended
RAM	1 GB; 2 GB recommended
Network Interface Card (NIC)	100 Mbps or faster NIC

OVF Tool 1.0 System Requirements

OVF Tool is a client application that can run on Windows and Linux systems. It cannot run on the same machine as the VMware ESX server. The following table describes the minimum system requirements.

Table 4. Minimum System Requirements for OVF Tool Installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "OVF Tool User Guide" PDF file at http://kb.vmware.com/.
CPU	750 MHz minimum, 1 GHz or faster recommended.
RAM	1 GB Minimum, 2 GB recommended.
Network Interface Card (NIC)	100 Mbps or faster NIC

For information about installing OVF, search for the "OVF Tool User Guide" PDF file at http://kb.vmware.com/.

Downloading the NetScaler virtual appliance Setup Files

The NetScaler virtual appliance setup package for VMware ESX follows the Open Virtual Machine (OVF) format standard. You can download the files from MyCitrix.com. You need a My Citrix account to log on. If you do not have a My Citrix account, access the home page at http://www.mycitrix.com, click the New Users link, and follow the instructions to create a new My Citrix account.

Once logged on, navigate the following path from the My Citrix home page:

MyCitrix.com > Downloads > NetScaler > Virtual Appliances.

Copy the following files to a workstation on the same network as the ESX server. Copy all three files into the same folder.

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk (for example, NSVPX-ESX-9.3-39.8-disk1.vmdk)
- NSVPX-ESX-<release number>-<build number>.ovf (for example, NSVPX-ESX-9.3-39.8.ovf)
- NSVPX-ESX-<release number>-<build number>.mf (for example, NSVPX-ESX-9.3-39.8.mf)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Labeling the Physical Network Ports of VMware ESX

Before installing a NetScaler virtual appliance, label of all the interfaces that you plan to assign to virtual appliances, in a unique format, for example, NS_NIC_1_1, NS_NIC_1_2, and so on. In large deployments, labeling in a unique format helps in quickly identifying the interfaces that are allocated to the NetScaler virtual appliance among other interfaces used by other virtual machines, such as Windows and Linux. Such labeling is especially important when different types of virtual machines share the same interfaces.

To label the physical network ports of VMware ESX server

1. Log on to the VMware ESX server by using the vSphere client.
2. On the vSphere client, select the Configuration tab, and then click Networking.
3. At the top-right corner, click Add Networking.
4. In the Add Network Wizard, for **Connection Type**, select **Virtual Machine**, and then click Next.
5. Scroll through the list of vSwitch physical adapters, and choose the physical port that will map to interface 1/1 on the virtual appliances.
6. Enter the label of the interface, for example, **NS_NIC_1_1** as the name of the vSwitch that will be associated with interface 1/1 of the virtual appliances.
7. Click Next to finish the vSwitch creation. Repeat the procedure, beginning with step 2, to add any additional interfaces to be used by your virtual appliances. Label the interfaces sequentially, in the correct format (for example, NS_NIC_1_2).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing NetScaler Virtual Appliances on VMware ESX 5.1 or Later

May 02, 2016

After you have installed and configured VMware ESX 5.1 or later, you can use the VMware vSphere client to install virtual appliances on the VMware ESX server. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running VMware ESX.

> **Note**
>
> By default, the NetScaler Virtual Appliance uses E1000 network interfaces.

To install NetScaler virtual appliances on VMware ESX 5.1 or later by using VMware vSphere Client:

1. Start the VMware vSphere client on your workstation.
2. In the **IP address / Name** text box, type the IP address of the VMware ESX server that you want to connect to.
3. In the **User Name** and **Password** text boxes, type the administrator credentials, and then click Login.
4. On the **File** menu, click **Deploy OVF Template**.
5. In the **Deploy OVF Template** dialog box, in **Deploy from file**, browse to the location at which you saved the NetScaler virtual appliance setup files, select the .ovf file, and click **Next**.
6. Map the networks shown in the virtual appliance OVF template to the networks that you configured on the ESX host. Click **Next** to start installing a virtual appliance on VMware ESX. When installation is complete, a pop-up window informs you of the successful installation.
7. You are now ready to start the NetScaler virtual appliance. In the navigation pane, select the NetScaler virtual appliance that you have just installed and, from the right-click menu, select **Power On**. Click the **Console** tab to emulate a console port.
8. If you want to install another virtual appliance, repeat steps **4** through **6**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring NetScaler Virtual Appliances to use VMXNET3 Network Interface

Sep 06, 2017

> **Important**
>
> You can configure VMXNET3 network interface only on NetScaler VPX 11.0.65.72 and 11.0 68.x onwards.

After you have installed and configured the NetScaler virtual appliance on the VMware ESX 5.5 or later, you can use the VMware vSphere web client to configure the virtual appliance to use VMXNET3 network interfaces.

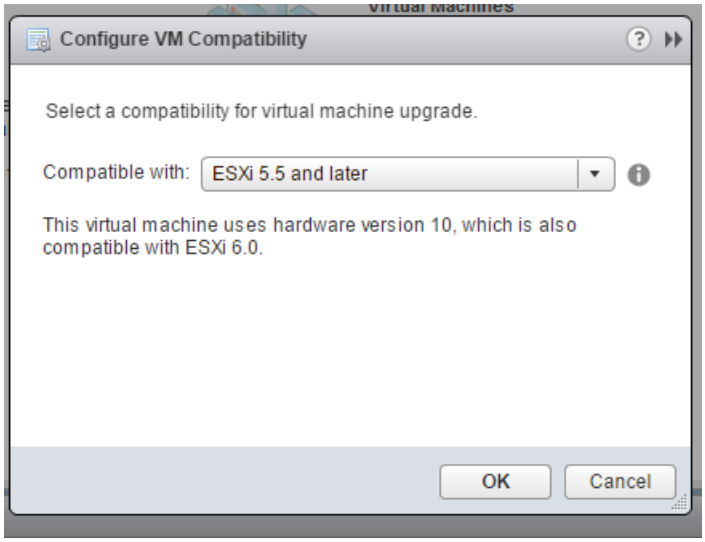
To configure NetScaler Virtual Appliances to use VMXNET3 network interfaces by using the VMware vSphere Web Client:

1. In the vSphere Web Client, select **Hosts and Clusters**.
2. Upgrade the Compatibility setting of the NetScaler virtual machine to ESX 5.5 or later, as follows:

   a. Power off the NetScaler virtual machine.

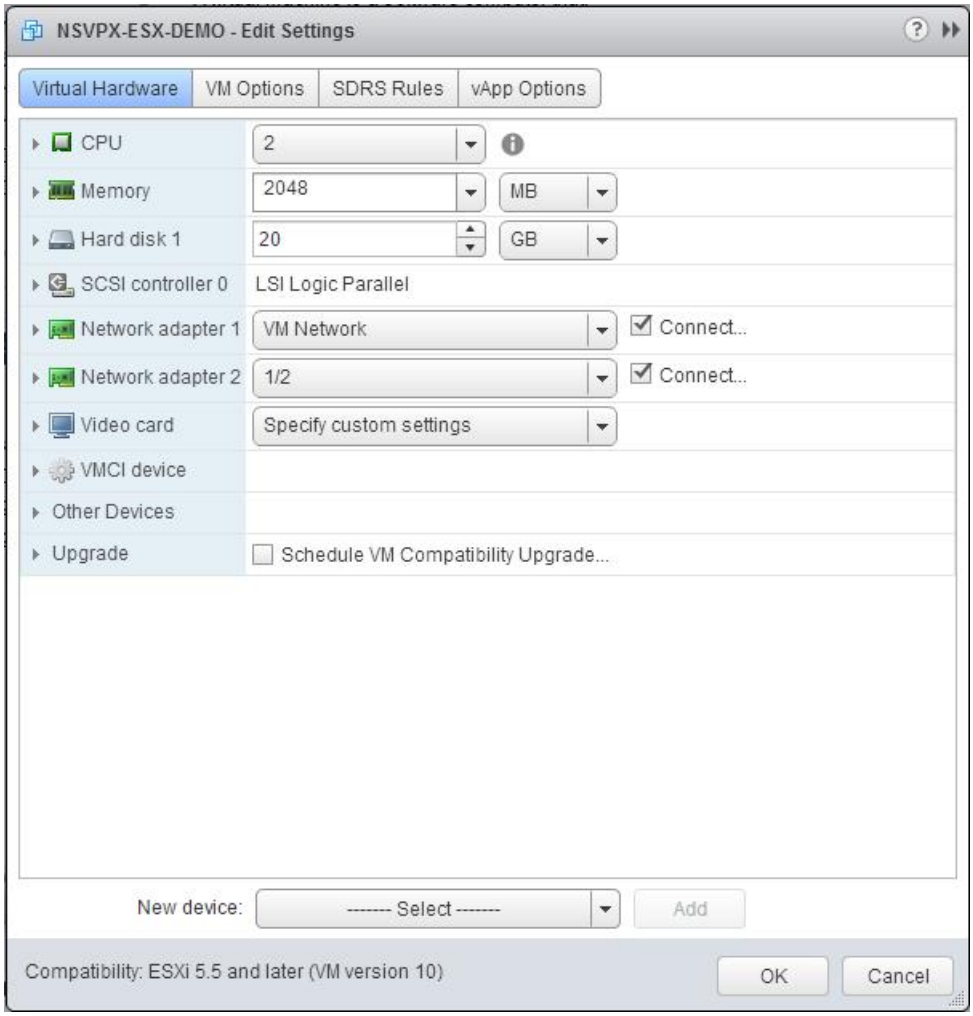
   b. Right-click the NetScaler virtual machine and select **Compatibility > Upgrade VM Compatibility**.

   c. In the **Configure VM Compatibility** dialog box, select **ESXi 5.5 and later** from the **Compatible with** drop-down list and click **OK**.



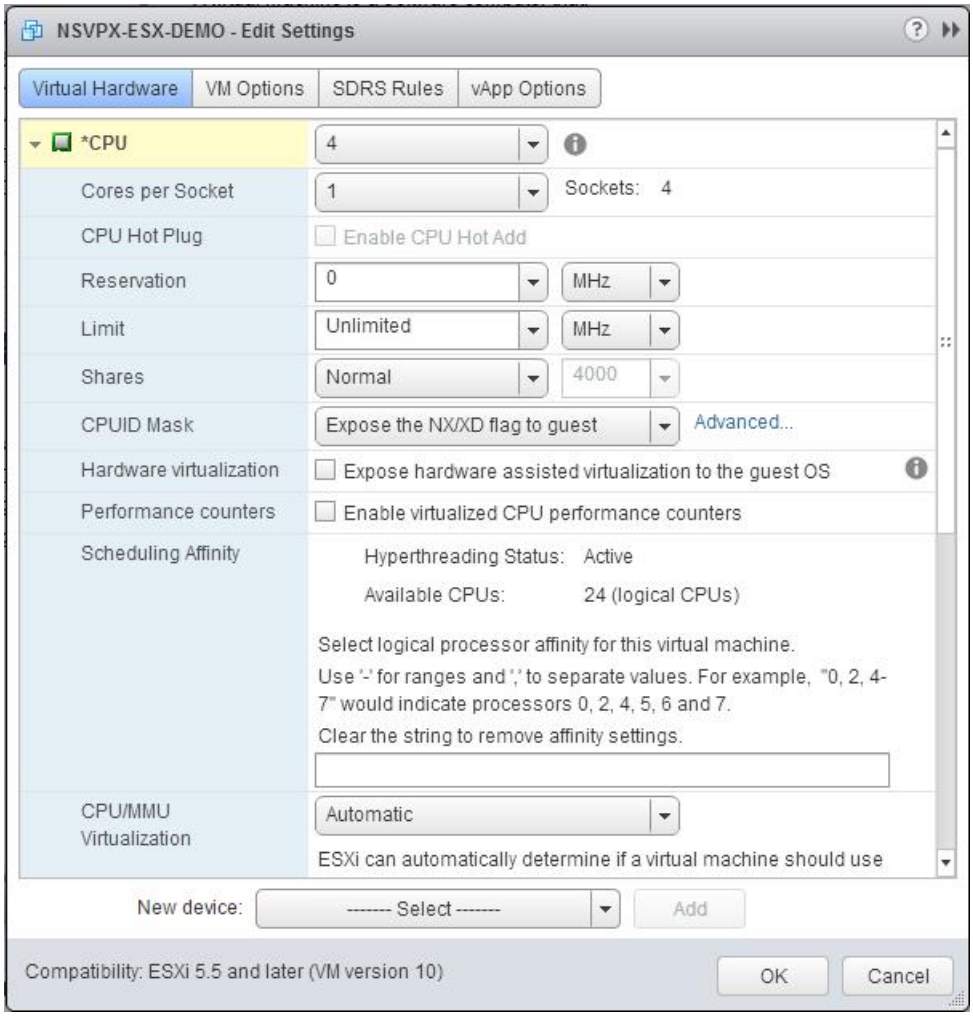
3. Right-click on the NetScaler virtual appliance and click **Edit Settings**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



4. In the **<virtual_appliance> - Edit Settings** dialog box, click the **CPU** section.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



5. In the **CPU** section, update the following:

   - Number of CPUs

   - Number of Sockets

   - Reservations

   - Limit

   - Shares

   Set the values as follows:

   a. In the **CPU** drop-down list, select the number of CPUs to assign to the virtual appliance.

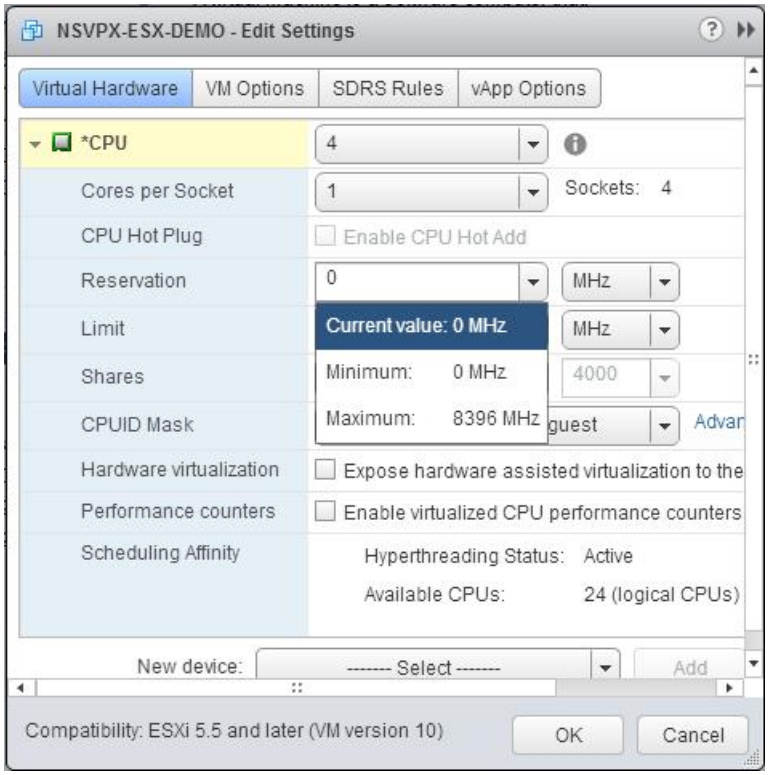
   b. In the **Cores per Socket** drop-down list, select the number of sockets.

   c. (Optional) In the **CPU Hot Plug** field, select or unselect the **Enable CPU Hot Add** checkbox.

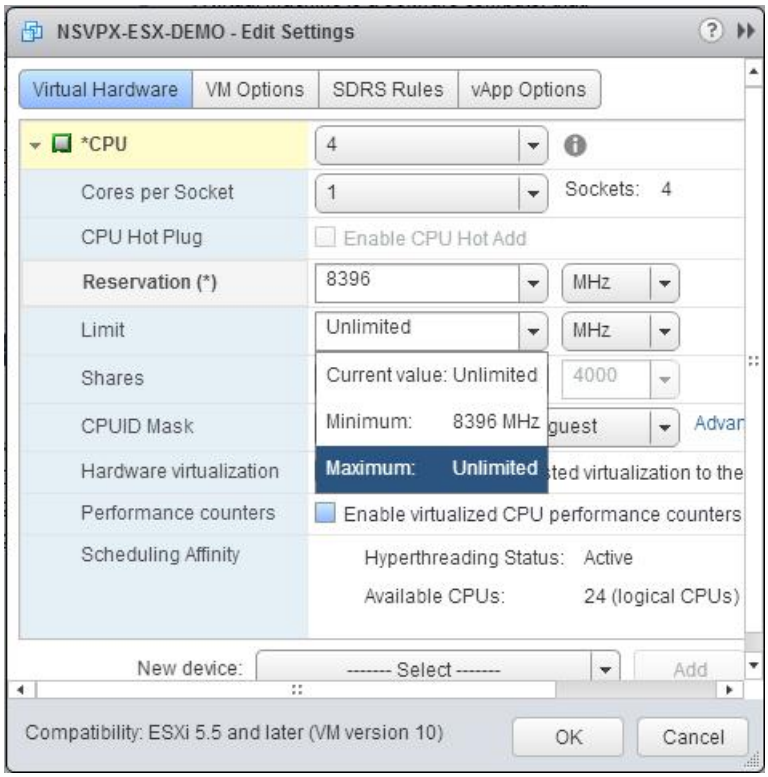
   Note: Citrix recommends accepting the default (disabled).

   d. In the **Reservation** drop-down list, select the number that is shown as the maximum value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

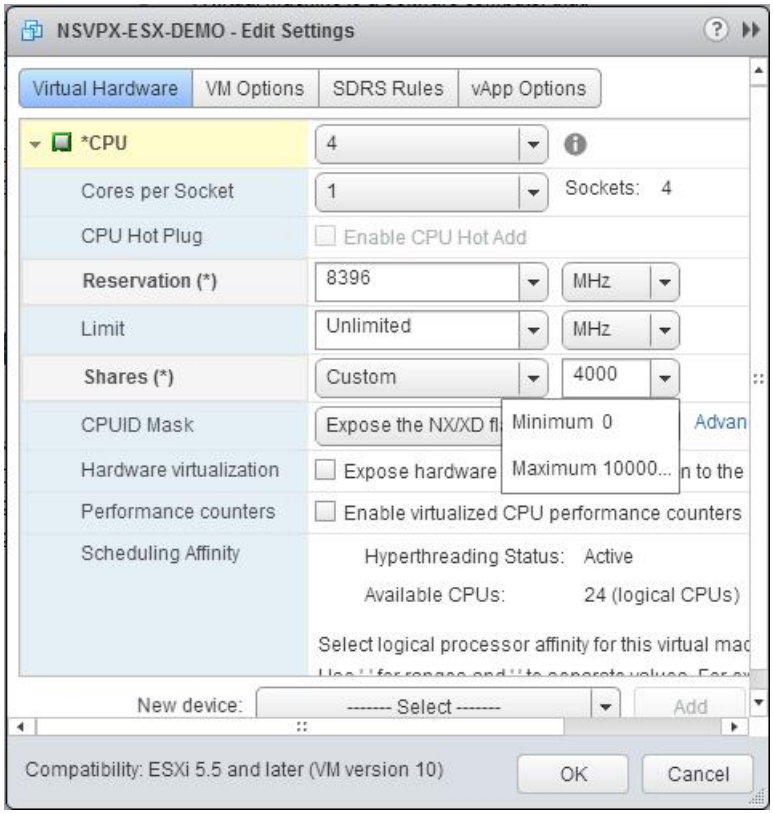


e. In the **Limit** drop-down list, select the number that is shown as the maximum value.



f. In the **Shares** drop-down lists, select **Custom** and the number that is shown as the maximum value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



6. In the **Memory** section, update the following:

- Size of RAM

- Reservations

- Limit

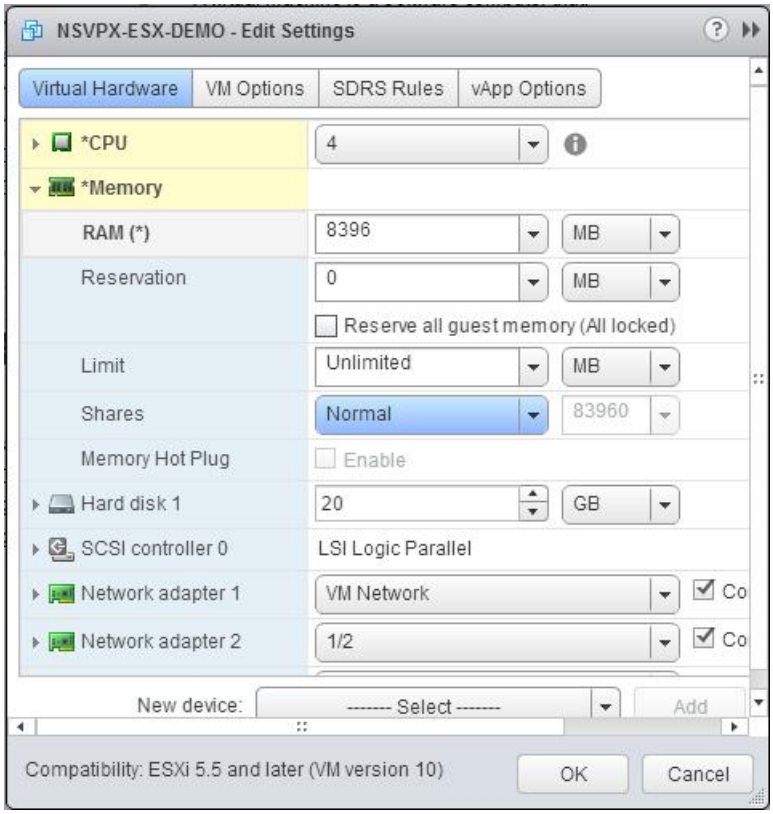
- Shares

Set the values as follows:

a. In the **RAM** drop-down list, select the size of the RAM. It should be number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the RAM should be 4 x 2 GB = 8 GB.

Note: For an Enterprise or Platinum edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.

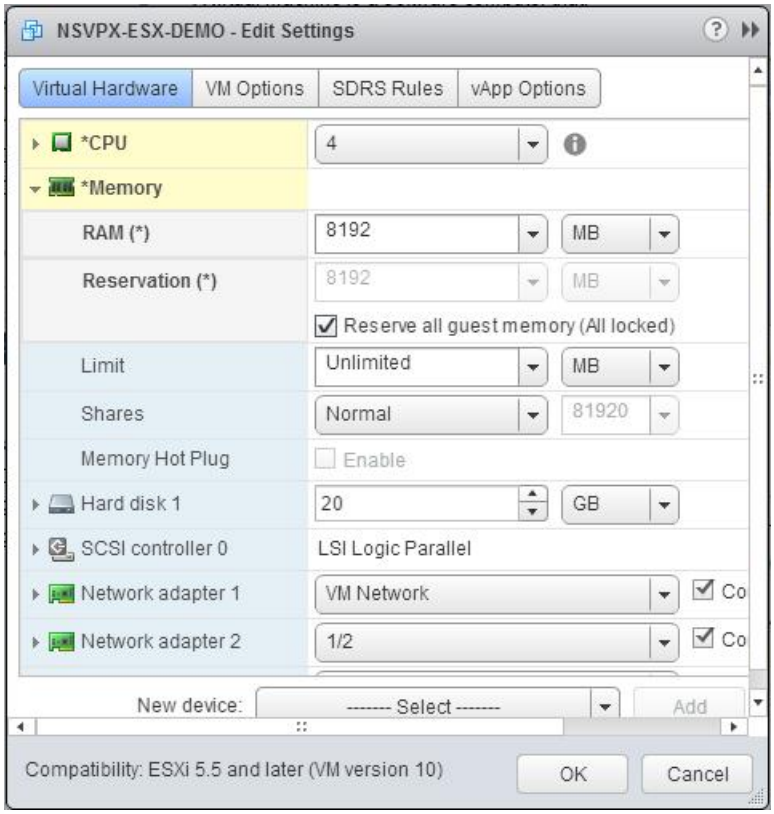
© 1999-2017 Citrix Systems, Inc. All rights reserved.



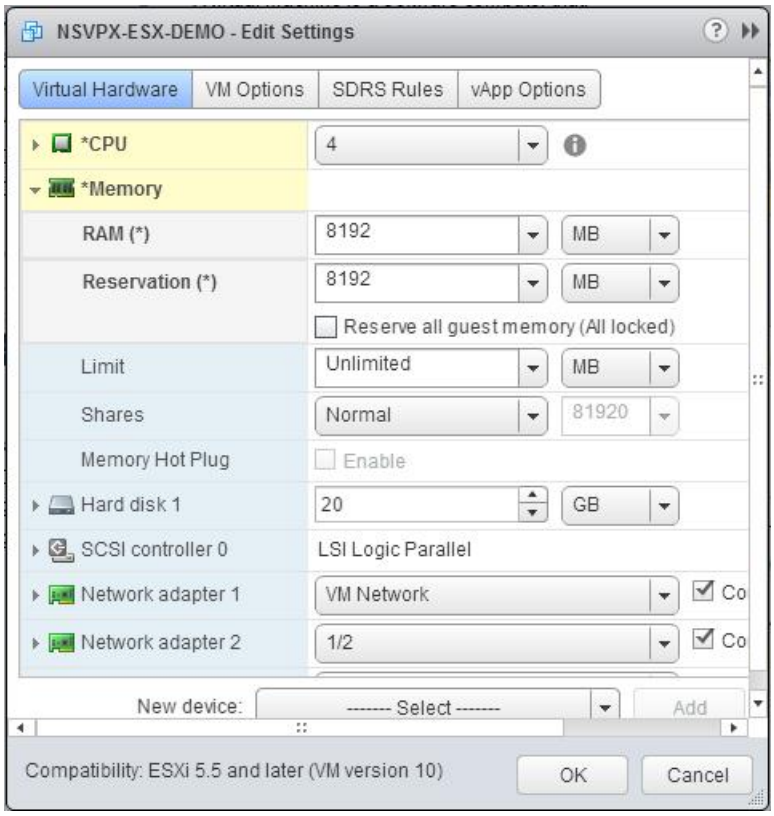
b. In the **Reservation** drop-down list, enter the value for the memory reservation, and select the **Reserve all guest memory (All locked)** checkbox. The memory reservation should be the number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the memory reservation should be 4 x 2 GB = 8 GB.

Note: For an Enterprise or Platinum edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

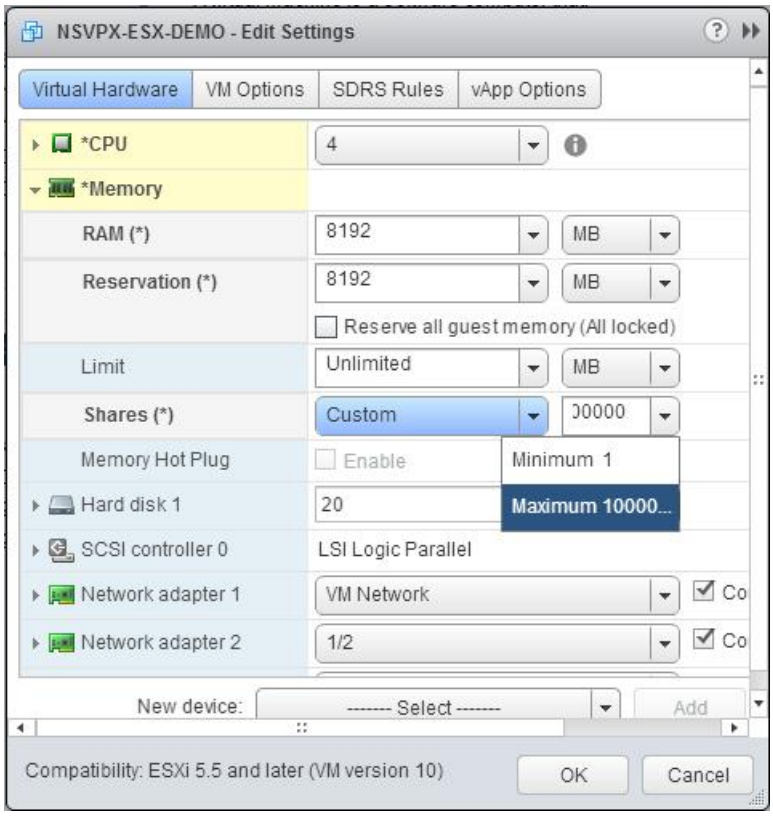


c. In the **Limit** drop-down list, select the number that is shown as the maximum value.

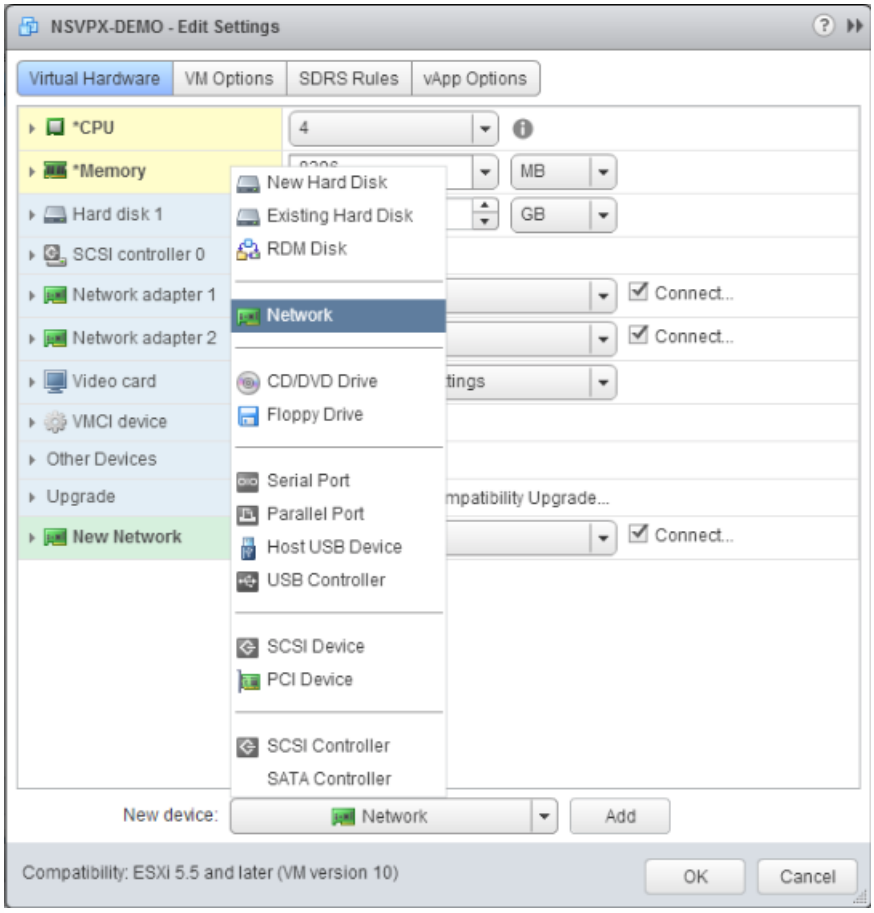


d. In the **Shares** drop-down lists, select **Custom** and the number that is shown as the maximum value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



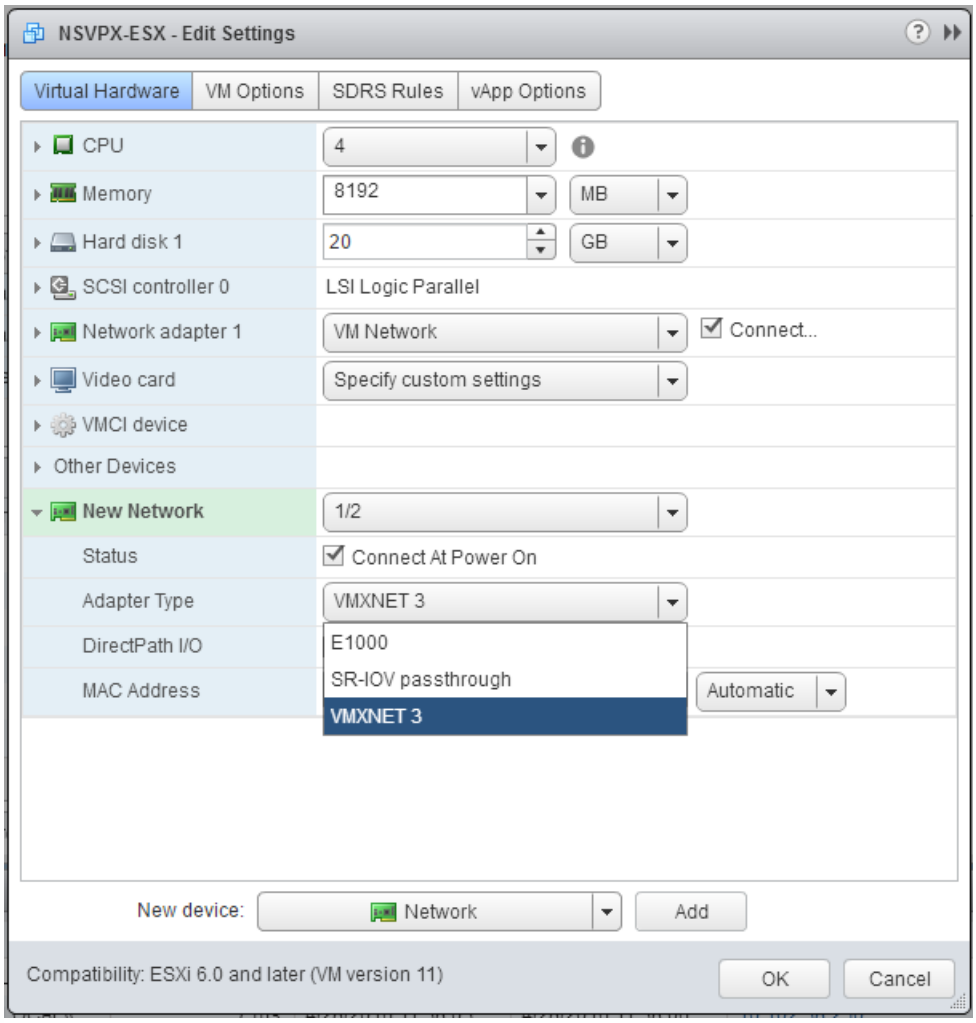
7. Add a VMXNET3 network interface. From the **New device** drop-down list, select **Network** and click **Add**.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

8. In the **New Network** section, from the drop-down list, select the network interface, and do the following:
   a. In the **Adapter Type** drop-down list, select **VMXNET3**.

> **Important**
>
> The default E1000 network interface and VMXNET3 cannot coexist, make sure that you remove the E1000 network interface and use VMXNET3 (0/1) as the management interface.



9. Click **OK**.

10. Power on the NetScaler virtual appliance.

11. Once the NetScaler virtual appliance powers on, you can use the following command to verify the configuration:

Command COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> show interface summary
```

The output should show all the interfaces that you configured:

command COPY

```
> show interface summary

--------------------------------------------------------------------------

    Interface  MTU      MAC               Suffix

--------------------------------------------------------------------------

1    0/1      1500     00:0c:29:89:1d:0e   NetScaler Vir...rface, VMXNET3

2    1/1      9000     00:0c:29:89:1d:18   NetScaler Vir...rface, VMXNET3

3    1/2      9000     00:0c:29:89:1d:22   NetScaler Vir...rface, VMXNET3

4    LO/1     9000     00:0c:29:89:1d:0e   Netscaler Loopback interface
```

Note

After you add a VMXNET3 interface and restart the NetScaler VPX appliance, the VMWare ESX hypervisor might change the order in which the NIC is presented to the VPX appliance. So, network adapter 1 might not always remain 0/1, resulting in loss of management connectivity to the VPX appliance. To avoid this issue, change the virtual network of the network adapter accordingly.

This is a VMWare ESX hypervisor limitation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring NetScaler Virtual Appliances to use Single Root I/O Virtualization (SR-IOV) Network Interface

Dec 12, 2016

> **Important**
>
> You can configure SR-IOV network interface only on NetScaler VPX 11.0.65.72 and 11.0.68.x onwards.

After you have installed and configured the NetScaler virtual appliance on the VMware ESX 5.5 or later, you can use the VMware vSphere web client to configure the virtual appliance to use SR-IOV network interfaces.

Prerequisite

Make sure that you:

- Add the Intel 82599 Network Interface Card (NIC) to the ESX Host.

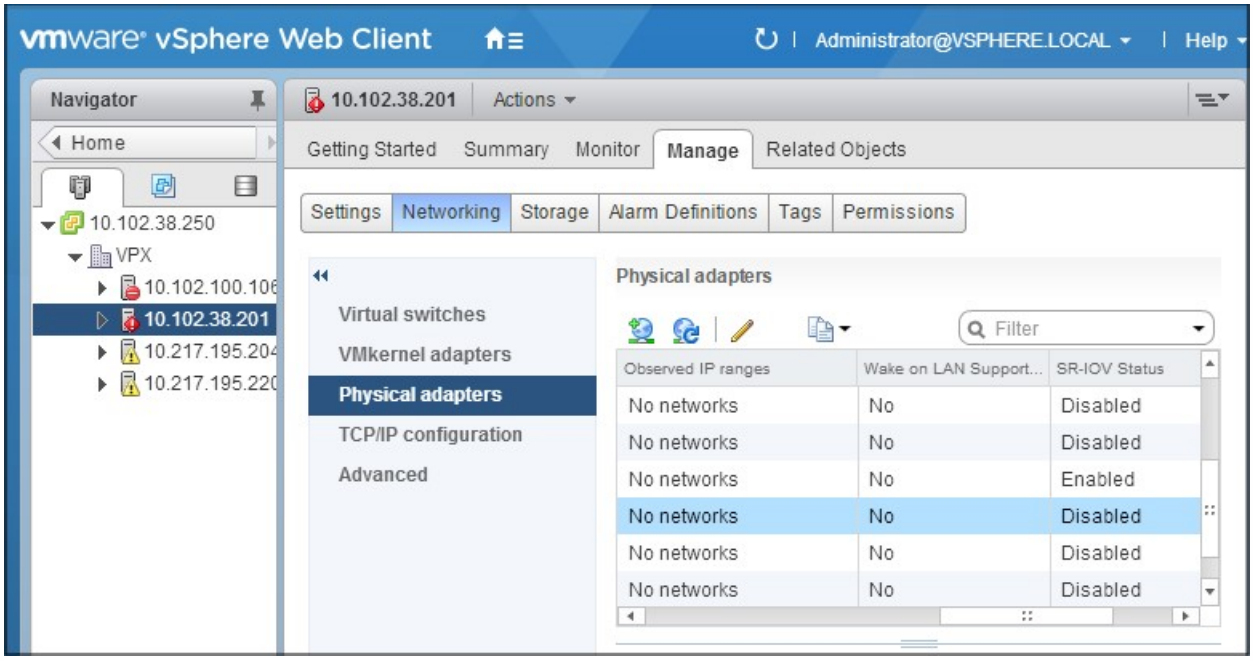
> **Note**
>
> IXGBE driver version 3.21.4.3 is recommended. Intel Corporation has reported that later versions of the IXGBE driver have issues in the VLAN-trunk mode of operation.

- Enable SR-IOV on the host physical adapter, as follows:

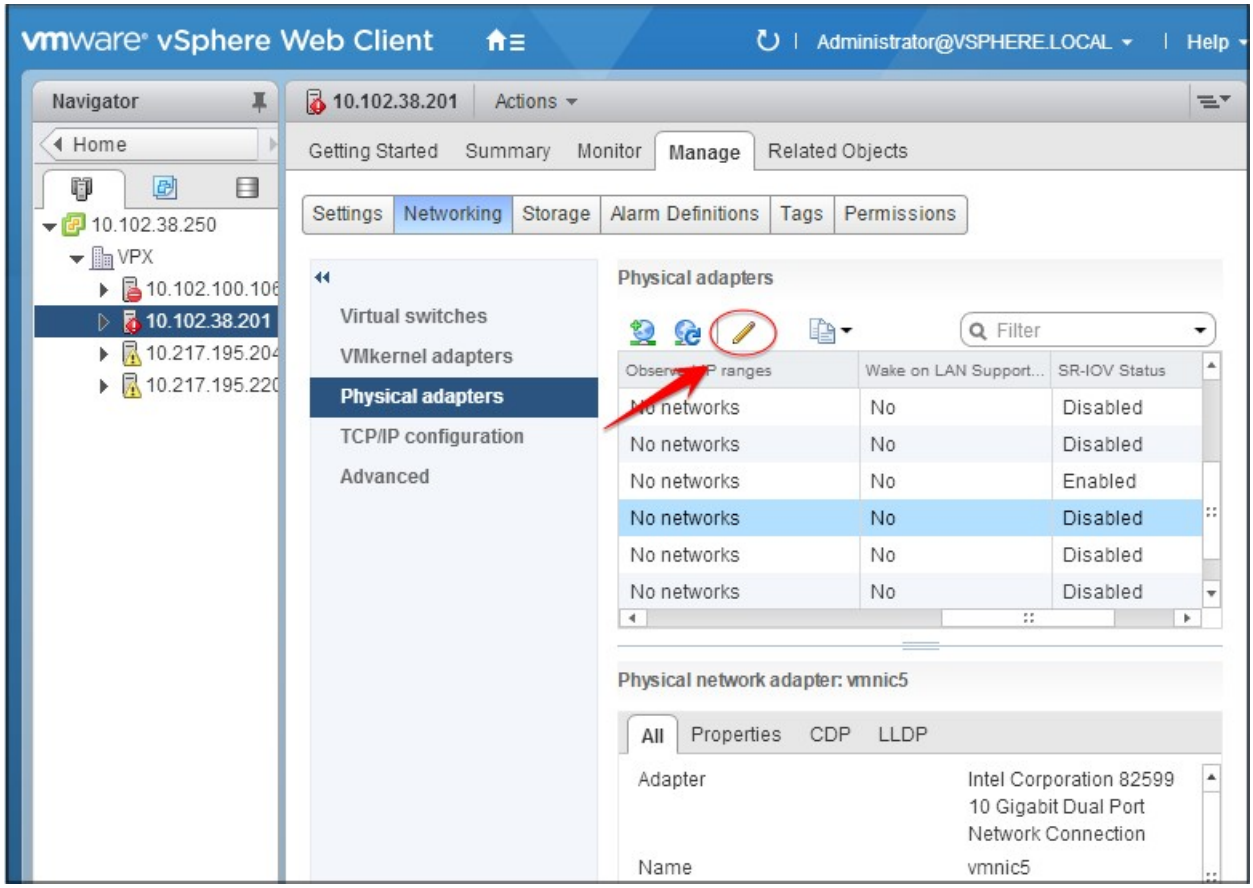
1. In the vSphere Web Client, navigate to the Host.

2. On the **Manage > Networking** tab, select **Physical adapters**. The SR-IOV Status field shows whether a physical adapter supports SR-IOV.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

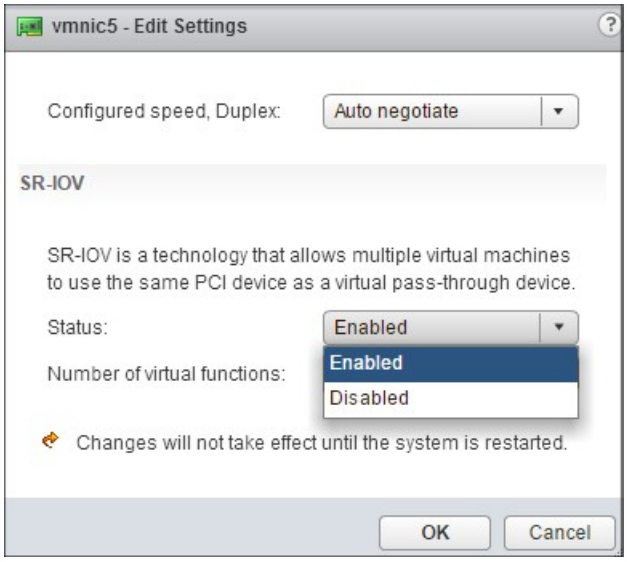


3. Select the physical adapter, and then click the pencil icon to open the **Edit Settings** dialog box.

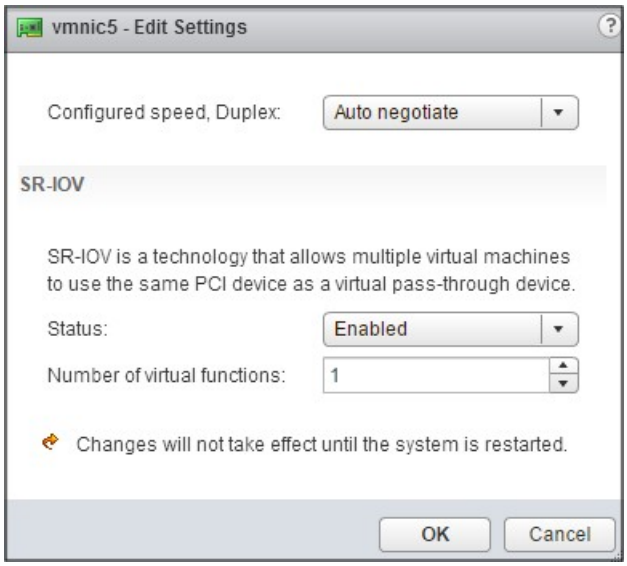


4. Under SR-IOV, select **Enabled** from the **Status** drop-down list.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



5. In the **Number of virtual functions** field, enter the number of virtual functions that you want to configure for the adapter.



6. Click **OK**.
7. Restart the host.

- Create a Distributed Virtual Switch (DVS) and Portgroups. For instructions, see the VMware Documentation.

Note

Citrix has qualified the SR-IOV configuration on DVS and Portgroups only.

Important

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> The default E1000 network interface and SR-IOV cannot coexist, make sure that you remove the E1000 network interface and use VMXNET3 (0/1) as the management interface.

To configure NetScaler Virtual Appliances to use SR-IOV network interface by using VMware vSphere Web Client:

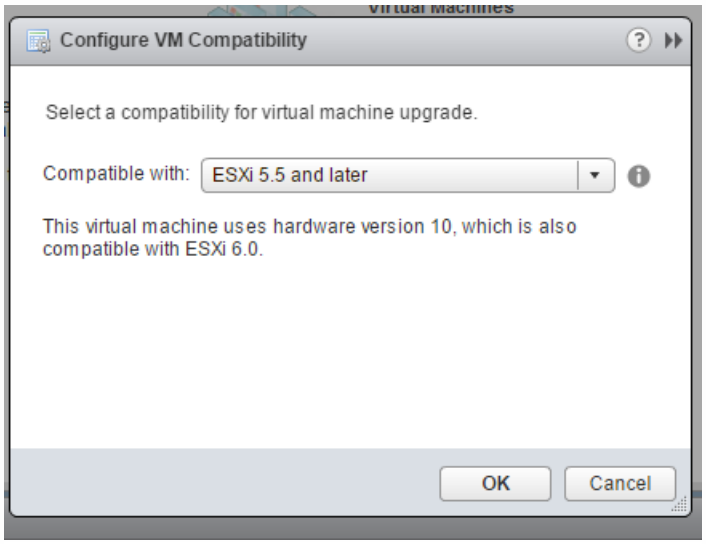
1. In the vSphere Web Client, select **Hosts and Clusters**.

2. Upgrade the Compatibility setting of the NetScaler virtual machine to ESX 5.5 or later, as follows:

   a. Power off the NetScaler virtual machine.

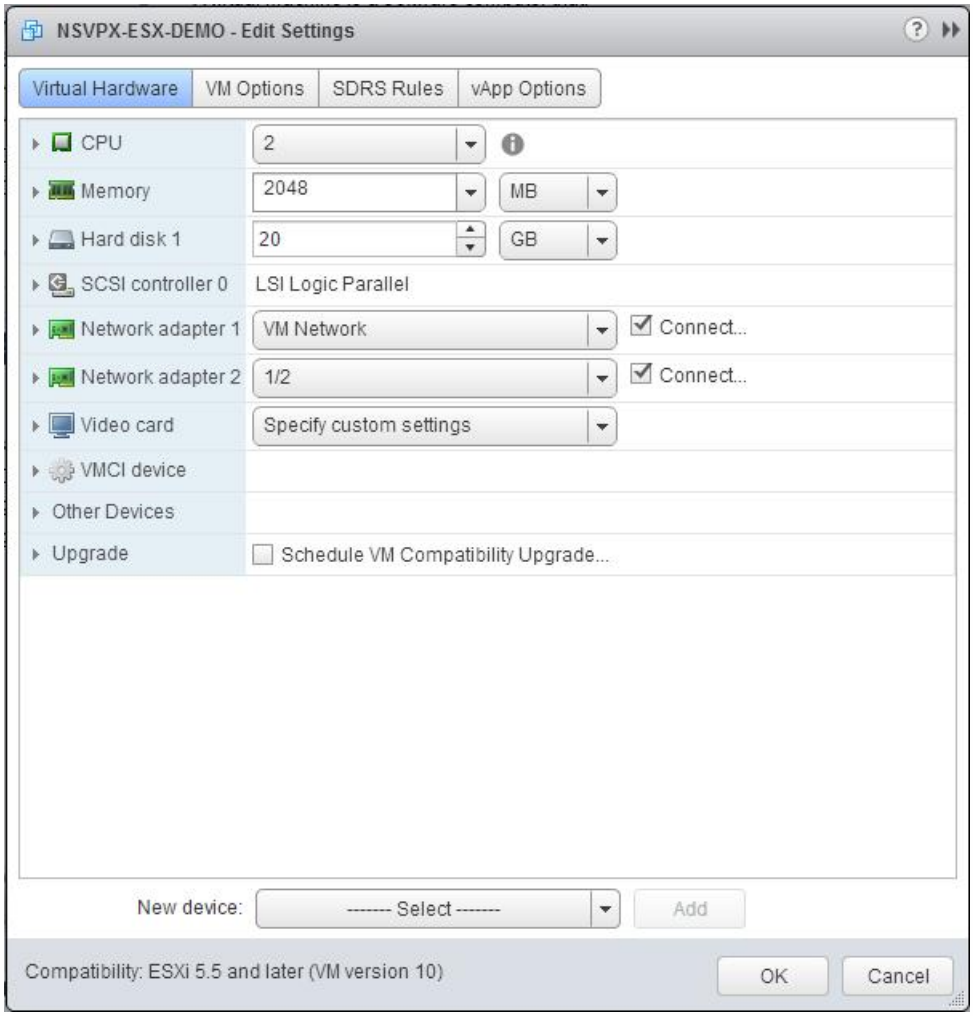
   b. Right-click the NetScaler virtual machine and select **Compatibility > Upgrade VM Compatibility**.

   c. In the **Configure VM Compatibility** dialog box, select **ESXi 5.5 and later** from the **Compatible with** drop-down list and click **OK**.



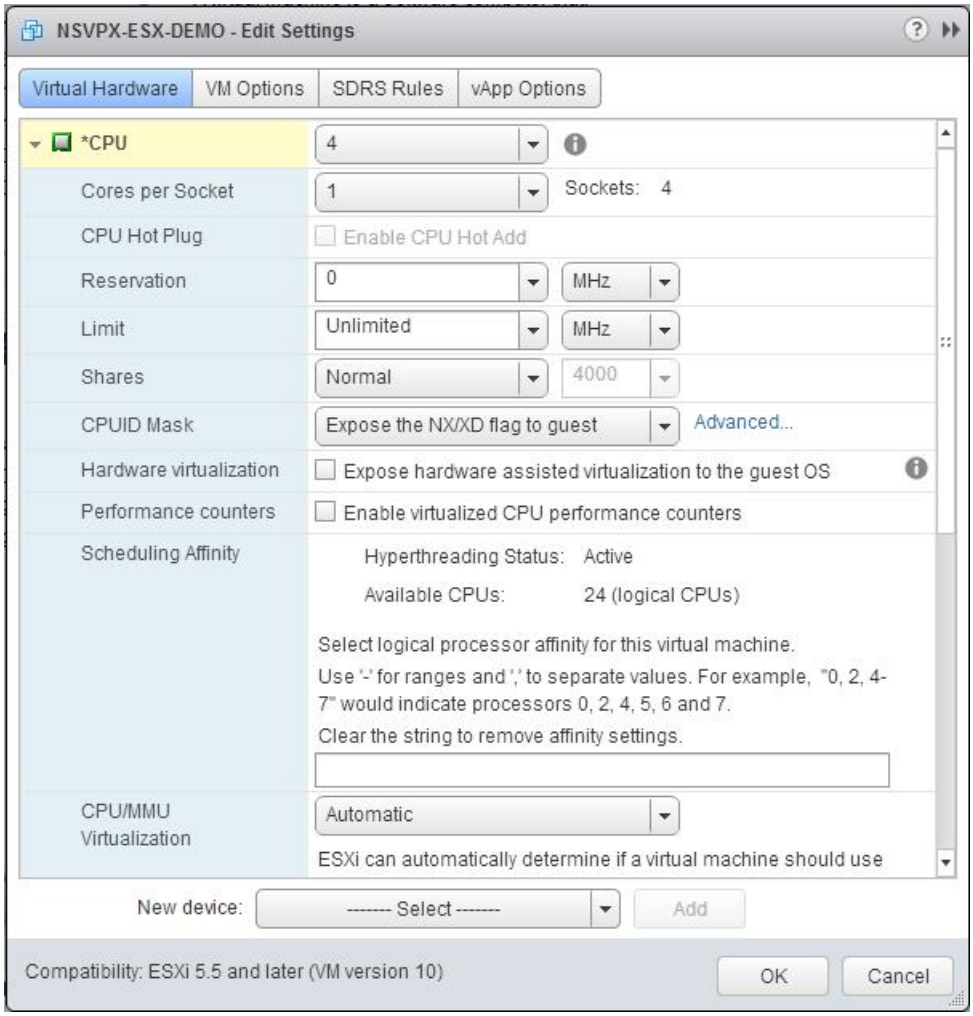
3. Right-click on the NetScaler virtual appliance and click **Edit Settings**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



4. In the **<virtual_appliance> - Edit Settings** dialog box, click the **CPU** section.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



5. In the **CPU** section, update the following settings:

   - Number of CPUs

   - Number of Sockets

   - Reservations

   - Limit

   - Shares

   Set the values as follows:

   a. In the **CPU** drop-down list, select the number of CPUs to assign to the virtual appliance.

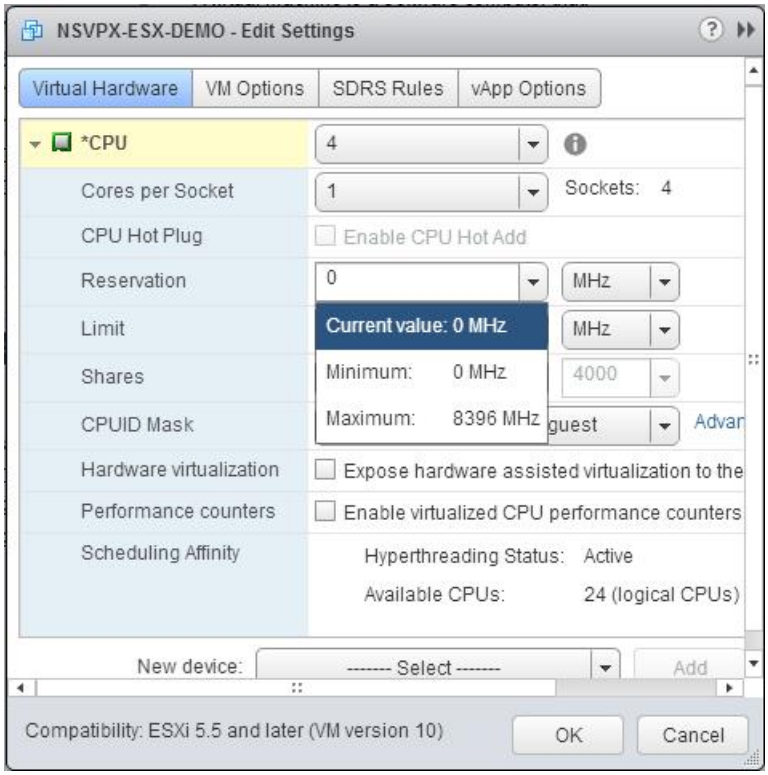
   b. In the **Cores per Socket** drop-down list, select the number of sockets.

   c. (Optional) In the **CPU Hot Plug** field, select or clear the **Enable CPU Hot Add** check box.

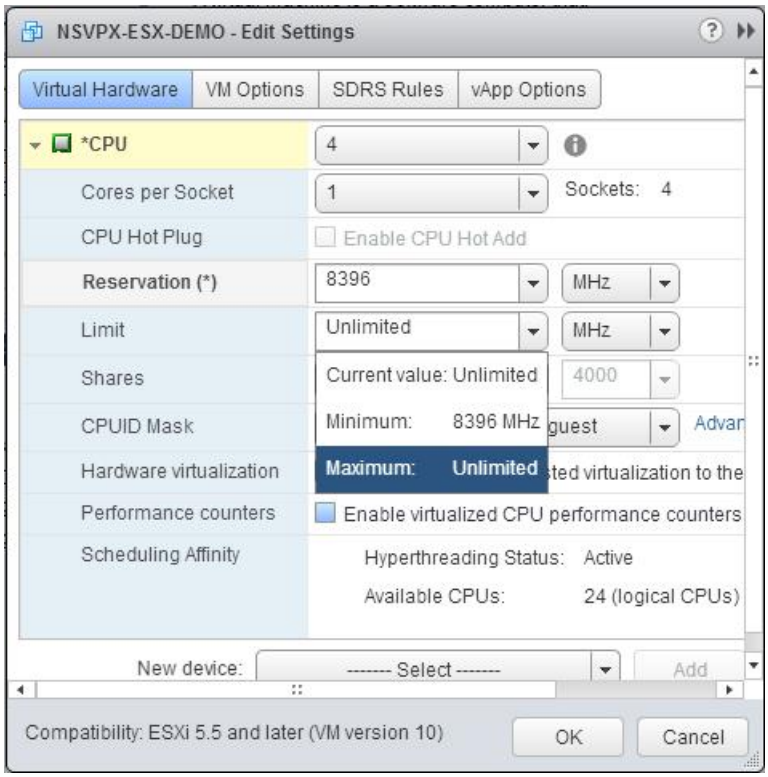
   Note: Citrix recommends accepting the default (disabled).

   d. In the **Reservation** drop-down list, select the number that is shown as the maximum value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

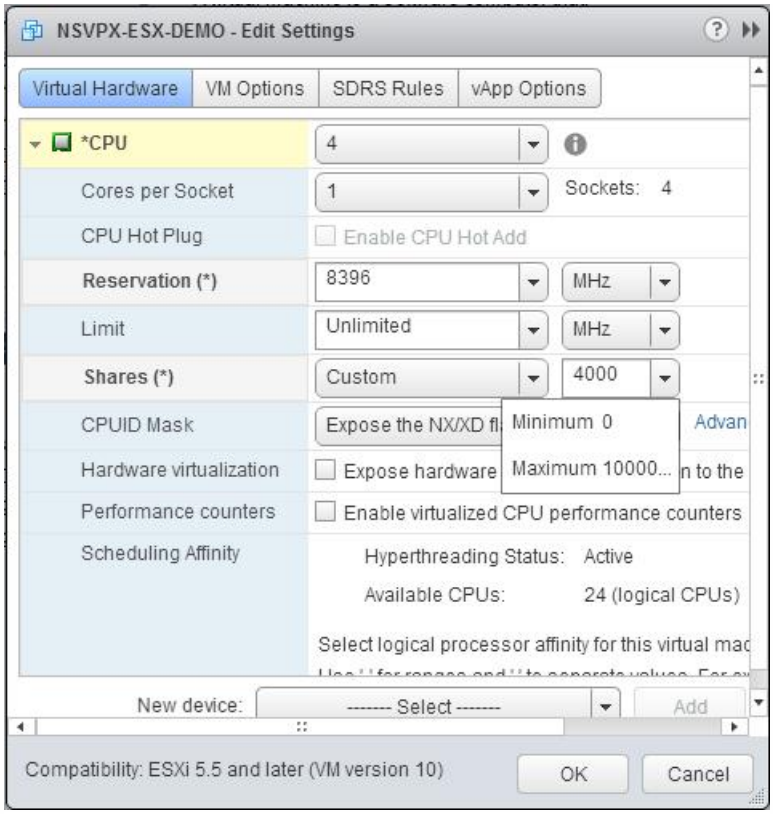


e. In the **Limit** drop-down list, select the number that is shown as the maximum value.



f. In the **Shares** drop-down lists, select **Custom** and the number that is shown as the maximum value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



6. In the **Memory** section, update the following settings:

- Size of RAM

- Reservations

- Limit

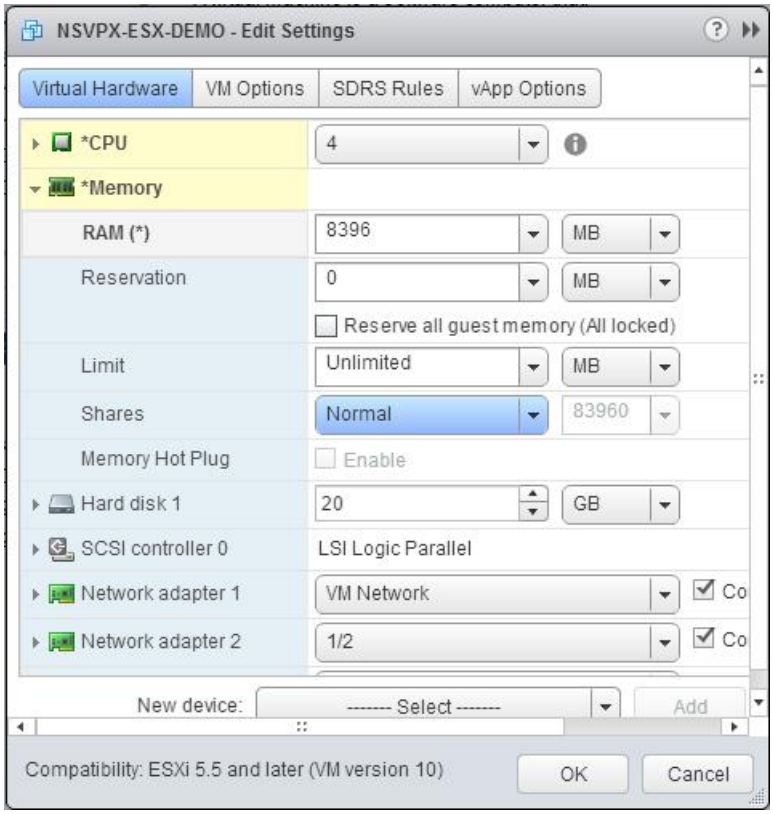
- Shares

Set the values as follows:

a. In the **RAM** drop-down list, select the size of the RAM. It should be number of vCPUs x 2 GB. For example, if the number of vCPU is 4 then RAM = 4 x 2 GB = 8 GB.

Note: For Enterprise or Platinum edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.

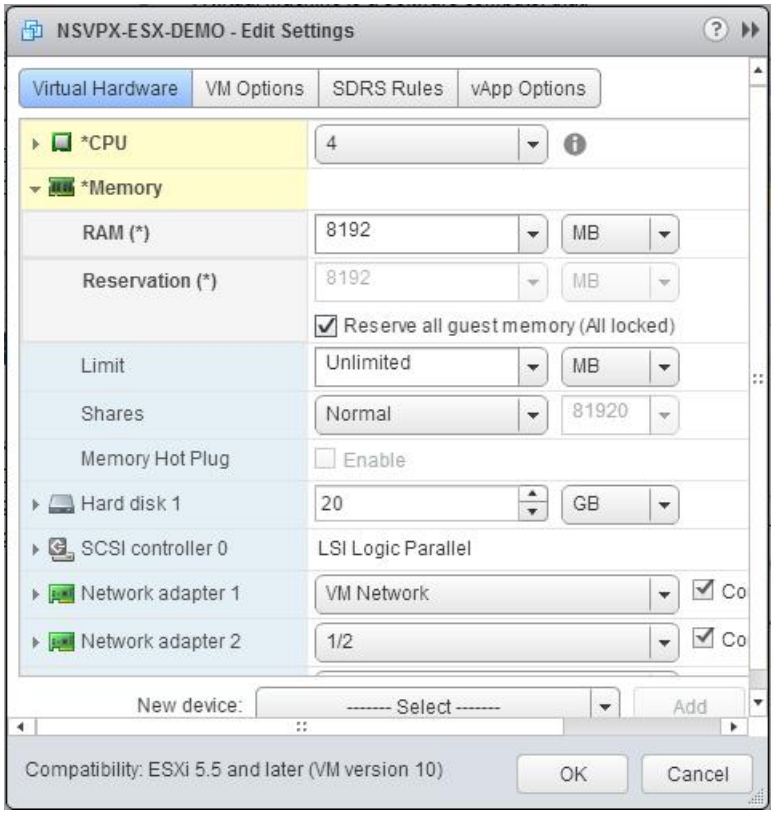
© 1999-2017 Citrix Systems, Inc. All rights reserved.



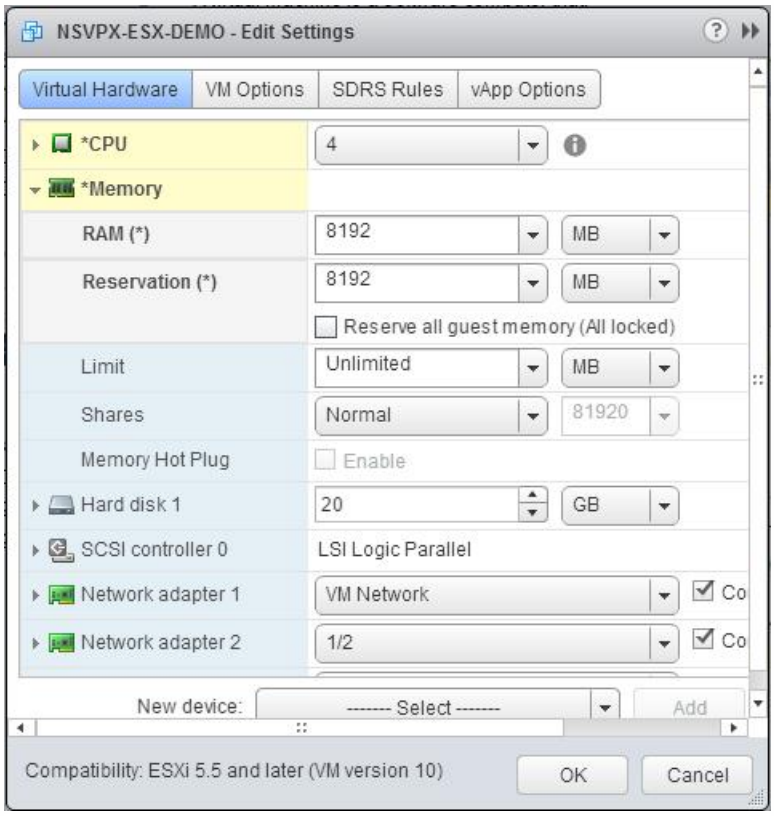
b. In the **Reservation** drop-down list, enter the value for the memory reservation, and select the **Reserve all guest memory (All locked)** check box. The memory reservation should be number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the memory reservation should be 4 x 2 GB = 8 GB.

Note: For Enterprise or Platinum edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

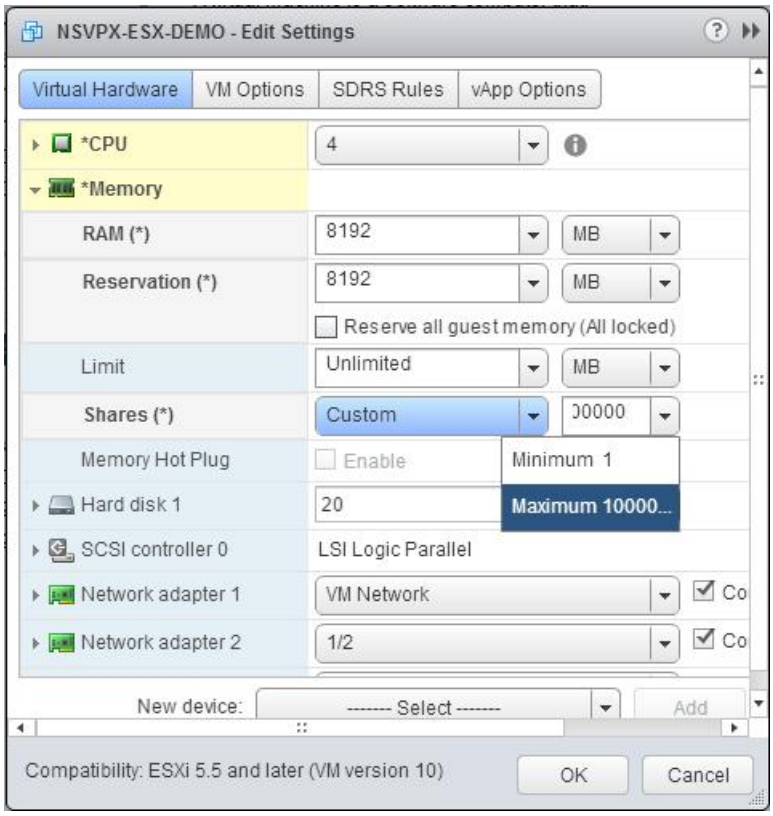


c. In the **Limit** drop-down list, select the number that is shown as the maximum value.



d. In the **Shares** drop-down lists, select **Custom**, and select the number that is shown as the maximum value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

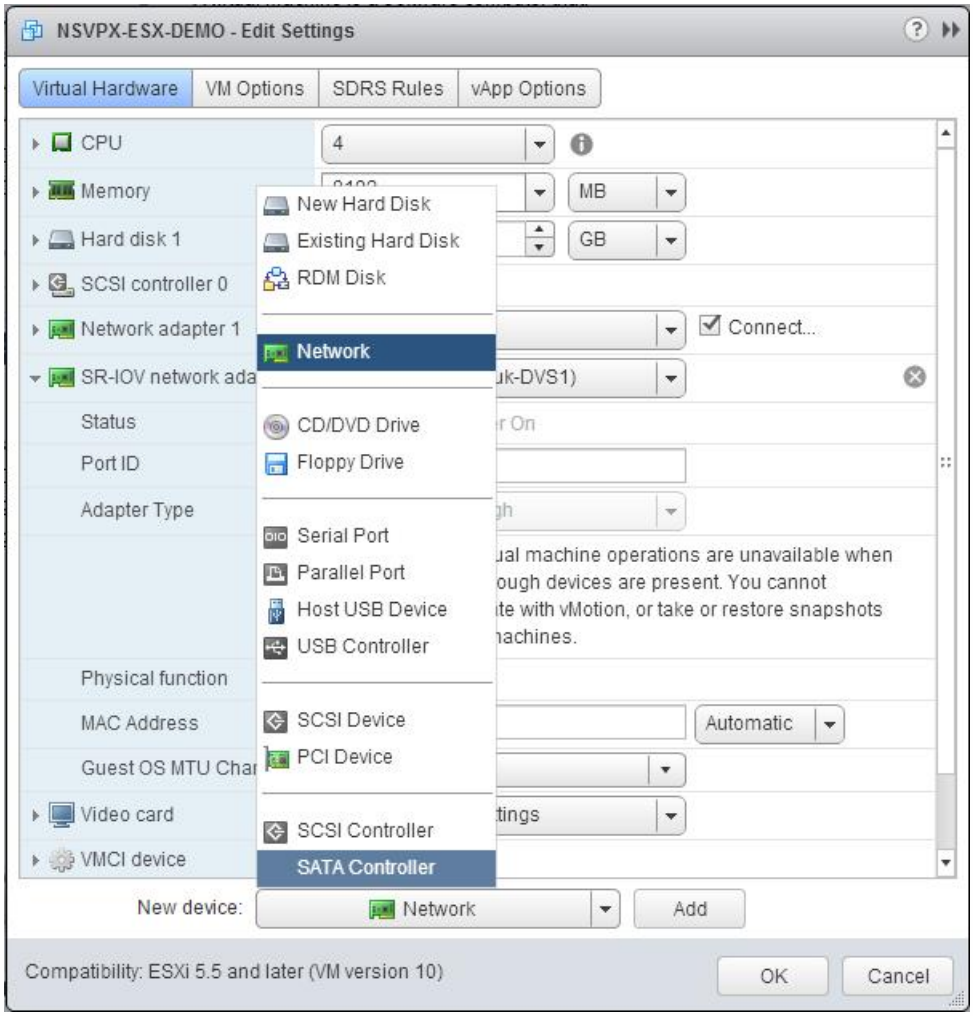


7. Add a SR-IOV network interface. From the **New device** drop-down list, select **Network** and click **Add**.

Note

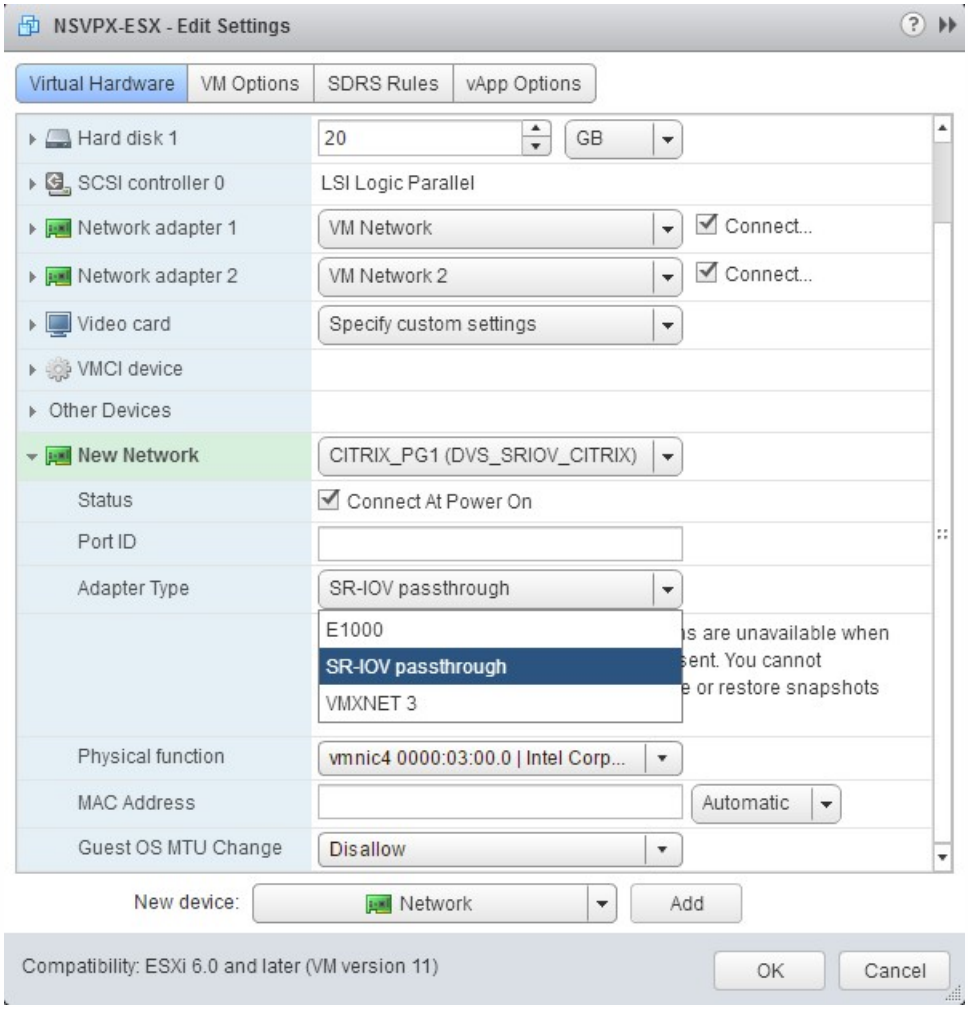
The default E1000 network interface and SR-IOV cannot coexist, make sure that you remove the E1000 network interface and use VMXNET3 (0/1) as the management interface.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



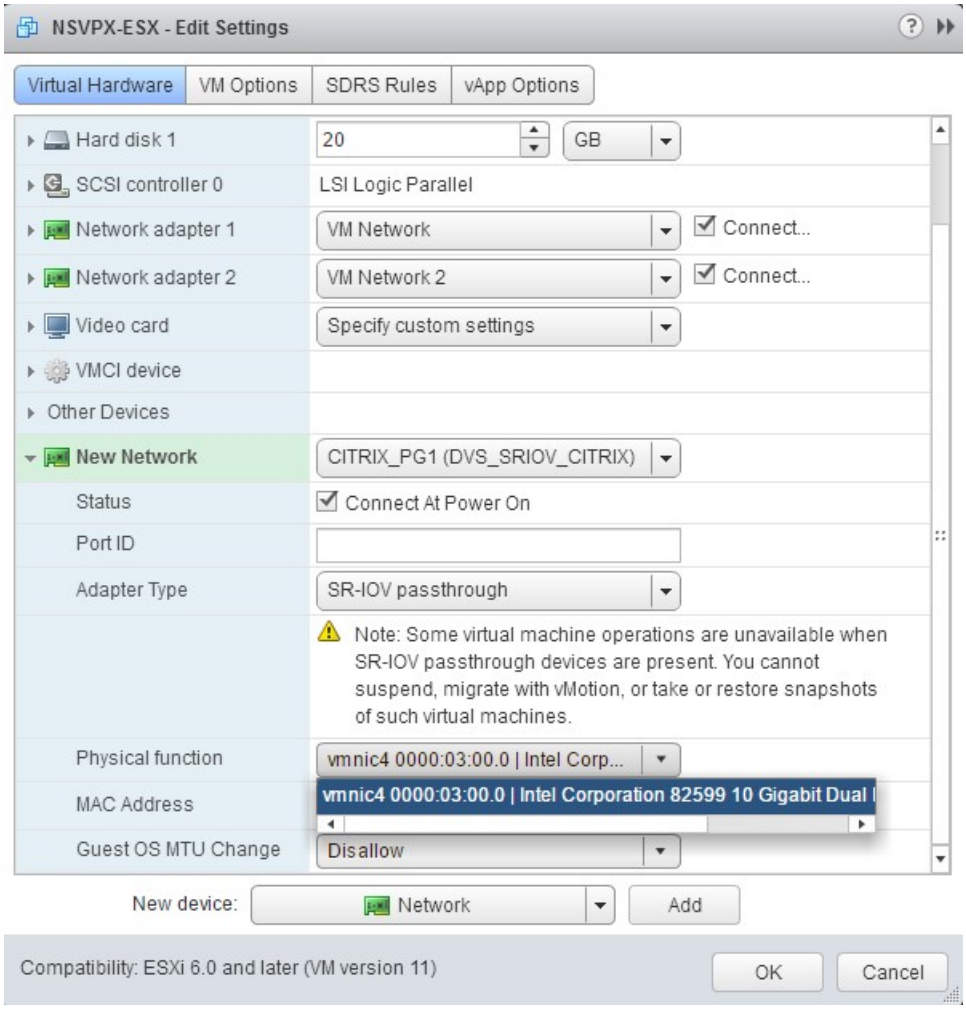
8. In the **New Network** section. From the drop-down list, select the Portgroup that you created, and do the following:
   a. In the **Adapter Type** drop-down list, select **SR-IOV passthrough**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



b. In the **Physical function** drop-down list, select the physical adapter mapped with the Portgroup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

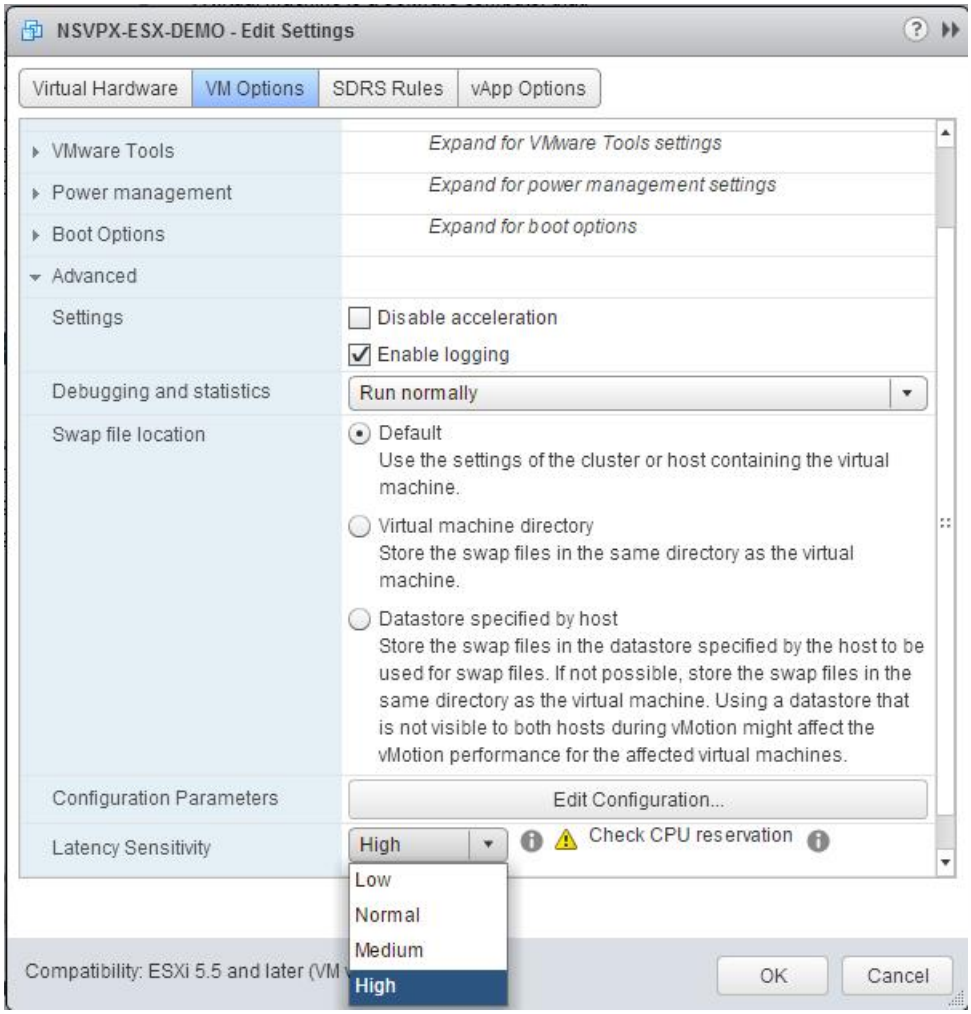


c. In the **Guest OS MTU Change** drop-down list, select **Disallow**.

9. In the **<virtual_appliance> - Edit Settings** dialog box, click the **VM Options** tab.

10. On the **VM Options** tab, select the **Advanced** section. From the **Latency Sensitivity** drop-down list, select **High**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



11. Click **OK**.

12. Power on the NetScaler virtual appliance.

13. Once the NetScaler virtual appliance powers on, you can use the following command to verify the configuration:

Code COPY

```
> show interface summary
```

The output should show all the interfaces that you configured:

Output COPY

```
> show interface summary

--------------------------------------------------------------------------
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Interface  MTU       MAC              Suffix

--------------------------------------------------------------------------------

1    0/1       1500      00:0c:29:1b:81:0b    NetScaler Virtual Interface

2    10/1      1500      00:50:56:9f:0c:6f    Intel 82599 10G VF Interface

3    10/2      1500      00:50:56:9f:5c:1e    Intel 82599 10G VF Interface

4    10/3      1500      00:50:56:9f:02:1b    Intel 82599 10G VF Interface

5    10/4      1500      00:50:56:9f:5a:1d    Intel 82599 10G VF Interface

6    10/5      1500      00:50:56:9f:4e:0b    Intel 82599 10G VF Interface

7    LO/1      1500      00:0c:29:1b:81:0b    Netscaler Loopback interface

 Done

> show inter 10/1

1)     Interface 10/1 (Intel 82599 10G VF Interface) #1

       flags=0xe460 <ENABLED, UP, UP, HAMON, 802.1q>

       MTU=1500, native vlan=55, MAC=00:50:56:9f:0c:6f, uptime 0h21m53s

       Actual: media FIBER, speed 10000, duplex FULL, fctl NONE, throughput 10000

       LLDP Mode: NONE,              LR Priority: 1024



       RX: Pkts(838020742) Bytes(860888485431) Errs(0) Drops(2527) Stalls(0)

       TX: Pkts(838149954) Bytes(860895860507) Errs(0) Drops(0) Stalls(0)

       NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)

       Bandwidth thresholds are not set.
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Migrating the NetScaler VPX from E1000 to SR-IOV or VMXNET3 Network Interfaces

Nov 07, 2016

> **Important**
>
> You can migrate from E1000 to SR-IOV or VMXNET3 network interface only on NetScaler VPX 11.0.65.72 and 11.0 68.x onwards.

You can configure your exiting NetScaler VPX instances that use E1000 network interfaces to use SR-IOV or VMXNET3 network interfaces.

To configure an existing NetScaler VPX instance to use SR-IOV network interfaces, see Configuring NetScaler Virtual Appliances to use Single Root I/O Virtualization (SR-IOV) Network Interface.

To configure an existing NetScaler VPX instance to use VMXNET3 network interfaces, see Configuring NetScaler Virtual Appliances to use VMXNET3 Network Interface.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing Citrix NetScaler Virtual Appliances on Microsoft Hyper-V Servers

May 25, 2017

Note

- The NetScaler virtual appliance is supported on Microsoft Hyper-V Server 2008 R2, Microsoft Hyper-V Server 2012 and 2012 R2.
- Intermediate System-to-Intermediate System (ISIS) protocol is not supported on the NetScaler VPX virtual appliance hosted on the HyperV-2012 platform.

To install Citrix NetScaler virtual appliances on Microsoft Windows Server, you must first install Windows Server, with the Hyper-V role enabled, on a machine with adequate system resources. While installing the Hyper-V role, be sure to specify the network interface cards (NICs) on the server that Hyper-V will use to create the virtual networks. You can reserve some NICs for the host. Use Hyper-V Manager to perform the NetScaler virtual appliance installation.
NetScaler virtual appliance for Hyper-V is delivered in virtual hard disk (VHD) format. It includes the default configuration for elements such as CPU, network interfaces, and hard-disk size and format. After you install NetScaler virtual appliance, you can configure the network adapters on virtual appliance, add virtual NICs, and then assign the NetScaler IP address, subnet mask, and gateway, and complete the basic configuration of the virtual appliance.

Note:
After the initial configuration of the NetScaler appliance, if you want to upgrade the appliance to the latest software release, see "Upgrading or Downgrading the System Software."

Prerequisites for Installing NetScaler Virtual Appliance on Microsoft Servers

Before you begin installing a virtual appliance, do the following:

- Enable the Hyper-V role on Windows Servers . For more information, see http://technet.microsoft.com/en-us/library/ee344837(WS.10).aspx.
- Download the virtual appliance setup files.
- Obtain NetScaler virtual appliance license files. For more information about NetScaler virtual appliance licenses, see the *NetScaler VPX Licensing Guide* at http://support.citrix.com/article/ctx131110.

Microsoft Server Hardware Requirements

The following table describes the minimum system requirements for Microsoft Servers .

Table 1. Minimum System Requirements for Microsoft Servers

Component	Requirement
CPU	1.4 GHz 64-bit processor
RAM	3 GB

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Component	Requirement
Disk Space	32 GB or greater

The following table lists the virtual computing resources for each NetScaler virtual appliance.

Table 2. Minimum Virtual Computing Resources Required for Running NetScaler Virtual Appliance

Component	Requirement
RAM	2 GB
Virtual CPU	2
Disk Space	20 GB
Virtual Network Interfaces	1

Downloading the NetScaler Virtual Appliance Setup Files

NetScaler virtual appliance for Hyper-V is delivered in virtual hard disk (VHD) format. You can download the files from MyCitrix.com. You will need a My Citrix account to log on. If you do not have a My Citrix account, access the home page at http://www.mycitrix.com, click the New Users link, and follow the instructions to create a new My Citrix account.

To download the NetScaler virtual appliance setup files

1. In a Web browser, go to http://www.citrix.com/ and click My Citrix.
2. Type your user name and password.
3. Click Downloads.
4. In Search Downloads by Product, select NetScaler.
5. Under Virtual Appliances, click NetScaler VPX.
6. Copy the compressed file to your server.

Installing NetScaler Virtual Appliance on Microsoft Servers

After you have enabled the Hyper-V role on Microsoft Server and extracted the virtual appliance files, you can use Hyper-V Manager to install NetScaler virtual appliance. After you import the virtual machine, you need to configure the virtual NICs by associating them to the virtual networks created by Hyper-V.

You can configure a maximum of eight virtual NICs. Even if the physical NIC is DOWN, the virtual appliance assumes that the virtual NIC is UP, because it can still communicate with the other virtual appliances on the same host (server).

> **Note**
>
> You cannot change any settings while the virtual appliance is running. Shut down the virtual appliance and then make changes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To install NetScaler Virtual Appliance on Microsoft Server by using Hyper-V Manager:

1. To start Hyper-V Manager, click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In the navigation pane, under **Hyper-V Manage**r, select the server on which you want to install NetScaler virtual appliance.
3. On the **Action** menu, click **Import Virtual Machin**e.
4. In the **Import Virtual Machine** dialog box, in **Location**, specify the path of the folder that contains the NetScaler virtual appliance software files, and then select **Copy the virtual machine (create a new unique ID)**. This folder is the parent folder that contains the Snapshots, Virtual Hard Disks, and Virtual Machines folders.
5. Note: If you received a compressed file, make sure that you extract the files into a folder before you specify the path to the folder.
6. Click **Import**.
7. Verify that the virtual appliance that you imported is listed under **Virtual Machines**.
8. To install another virtual appliance, repeat steps **2** through **6**.

> **Important**
>
> Make sure that you extract the files to a different folder in step **4**.

Auto-provisioning NetScaler Virtual Appliance on Hyper-V

Auto-provisioning of NetScaler virtual appliance is optional. If auto-provisioning is not done, the virtual appliance provides an option to configure the IP address and so on.

To auto-provision NetScaler Virtual Appliance on Hyper-V:

1. Create an ISO9660 compliant ISO image using the xml file as depicted in the example. Make sure that the name of the xml file is **userdata**.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

oe:id=""

xmlns="http://schemas.dmtf.org/ovf/environment/1">

<PlatformSection>

<Kind>HYPER-V</Kind>

<Version>2013.1</Version>

<Vendor>CISCO</Vendor>

<Locale>en</Locale>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
</PlatformSection>

<PropertySection>

<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>

<Property oe:key="com.citrix.netscaler.platform" oe:value="NS1000V"/>

<Property oe:key="com.citrix.netscaler.orch_env" oe:value="cisco-orch-env"/>

<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.102.100.122"/>

<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="255.255.255.128"/>

<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.102.100.67"/></PropertySection>

</Environment>
```


2. Copy the ISO image to hyper-v server.

3. Select the virtual appliance that you imported, and then on the **Action** menu, select **Settings**. You can also select the virtual appliance and then right click and select **Settings**. The **Settings** window for the selected virtual appliance is displayed.

4. In the **Settings** window, under the hardware section, click on **IDE Controller**.

5. In the right window pane, select **DVD Drive** and click on **Add**. The DVD Drive is added under the **IDE Controller** section in the left window pane.

6. Select the **DVD Drive** added in **step 5**. In the right window pane, select the Image file radio button and click on Browse and select the ISO image that you copied on Hyper-V server in step 2.

7. Click **Apply**.

> **Note**
>
> The virtual appliance instance comes up in the default IP address, when:
>
> - The DVD drive is attached and the ISO file is not provided.
> - The ISO file does not inculde the userdata file
> - The userdata file name or format is not correct

To configure virtual NICs on the NetScaler Virtual Appliance:

1. Select the virtual appliance that you imported, and then on the **Action** menu, select **Settings**.
2. In the **Settings for <virtual appliance name>** dialog box, click **Add Hardware** in the left pane.
3. In the right pane, from the list of devices, select **Network Adapter**.
4. Click **Add**.
5. Verify that **Network Adapter (not connected)** appears in the left pane.
6. Select the network adapter in the left pane.
7. In the right pane, from the **Network** drop-down list, select the virtual network to connect the adapter to.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

8. To select the virtual network for additional network adapters that you want to use, repeat steps **6** and **7**.
9. Click **Apply**, and then click **OK**.

To configure NetScaler Virtual Appliance:

1. Right-click the virtual appliance that you previously installed, and then select **Start**.
2. Access the console by double-clicking the virtual appliance.
3. Type the NetScaler IP address, subnet mask, and gateway for your virtual appliance.

You have completed the basic configuration of your virtual appliance. Type the IP address in a Web browser to access the virtual appliance.

> **Note**
>
> You can also use virtual machine (VM) template to provision NetScaler virtual appliance using SCVMM.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing NetScaler Virtual Appliances on Linux-KVM Platform

Dec 27, 2016

To set up NetScaler VPX for the Linux-KVM platform, you can use the graphical Virtual Machine Manager (Virt-Manager) application. If you prefer the Linux-KVM command line, you can use the virsh program.

The host Linux operating system must be installed on suitable hardware by using virtualization tools such as KVM Module and QEMU. The number of virtual machines (VMs) that can be deployed on the hypervisor depends on the application requirement and the chosen hardware.

After you provision a NetScaler virtual appliance, you can add additional interfaces.

Limitations and Usage Guidelines

General Recommendations

To avoid unpredictable behavior, apply the following recommendations:

- Do not change the MTU of the vnet interface associated with the NetScaler VM. Shut down the NetScaler VM before modifying any configuration parameters, such as Interface modes or CPU.
- Do not force a shutdown of the NetScaler VM. That is, do not use the **Force off** command.
- Any configurations done on the host Linux might or might not be persistent, depending on your Linux distribution settings. You can choose to make these configurations persistent to ensure consistent behavior across reboots of host Linux operating system.
- The NetScaler package has to be unique for each of the NetScaler VPX instance provisioned.

Limitations

- Live Migration of the Netscaler VPX running on KVM is not supported.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Prerequisites for Installing NetScaler VPX Virtual Appliances on Linux-KVM Platform

Dec 27, 2016

Hardware Requirements

The following table describes the minimum system requirements for Linux-KVM servers running NetScaler VPX.

Component	Requirement
CPU	• 64-bit x86 processors with the hardware virtualization features included in the AMD-V and Intel VT-X processors. To test whether your CPU supports Linux host, enter the following command at the host Linux shell prompt: **.egrep'^flags.*(vmx|svm)'/proc/cpuinfo** If the BIOS settings for the above extension are disabled, you must enable them in BIOS. • Provide at least 2 CPU cores to Host Linux. • There is no specific recommendation for processor speed, but higher the speed, the better the performance of the VM application.
Memory (RAM)	Minimum 4 GB for the host Linux kernel. Add additional memory as required by the VMs.
Hard Disk	Calculate the space for Host Linux kernel and VM requirements. A single NetScaler VPX VM requires 20 GB of disk space.

Software Requirements

The Host kernel used must be a 64-bit Linux kernel, release 2.6.20 or later, with all virtualization tools. Citrix recommends newer kernels, such as 3.6.11-4 and later.

Many Linux distributions such as Red Hat, Centos, and Fedora, have tested kernel versions and associated virtualization tools.

Guest VM Hardware Requirements

NetScaler VPX supports IDE and virtIO hard disk type. The Hard Disk Type has been configured in the XML file, which is a part of the NetScaler package.

Networking Requirements

NetScaler VPX supports only virtIO para-virtualized network interfaces.

Source Interface and Modes

The source device type can be either Bridge or MacVTap. In case of MacVTap, four modes are possible - VEPA, Bridge,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Private and Pass-through.The following tables list the types of interfaces that you can use and the supported traffic types.

For best performance by the NetScaler instance, make sure that the gro and lro capabilities are switched off on the source interfaces.

Table 1. Interface Types

Interface Type	Considerations
Source: Bridge	• Linux Bridge. • Ebtables and iptables settings on host Linux might filter the traffic on the bridge if you do not choose the correct setting or disable IPtable services.
Source: MacVTap Mode : VEPA	• Better performance than a bridge. • Interfaces from the same lower device can be shared across the VMs. • Inter-VM communication using the same lower device is possible only if upstream or downstream switch supports VEPA mode.
Source: MacVTap Mode : Private	• Better performance than a bridge. • Interfaces from the same lower device can be shared across the VMs. • Inter-VM communication using the same lower device is not possible.
Source: MacVTap Mode : Bridge	• Better as compared to bridge. • Interfaces out of same lower device can be shared across the VMs. • Inter-VM communication using the same lower device is possible, if lower device link is UP.
Source: MacVTap Mode : Pass-through	• Better as compared to bridge. • Interfaces out of same lower device cannot be shared across the VMs. • Only one VM can use the lower device.

Properties Of Source Interfaces

Make sure that you switch off the generic-receive-offload (gro) and large-receive-offload (lro) capabilities of the source interfaces. To switch off the gro and lro capabilities, run the following commands at the host Linux shell prompt.

ethtool -K eth6 gro off

ethool -K eth6 lro off

Example

```
command
```

COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
[root@localhost ~]# ethtool -K eth6
        Offload parameters for eth6:
                rx-checksumming: on
                tx-checksumming: on
        scatter-gather: on
        tcp-segmentation-offload: on
        udp-fragmentation-offload: off
        generic-segmentation-offload: on
        generic-receive-offload: off
        large-receive-offload: off
        rx-vlan-offload: on
        tx-vlan-offload: on
        ntuple-filters: off
        receive-hashing: on
[root@localhost ~]#
```

Example

If the host Linux bridge is used as a source device, as in the following example, gro and lro capabilities must be switched off on the vnet interfaces, which are the virtual interfaces connecting the host to the guest VMs.

```
command
```

COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
[root@localhost ~]# brctl show eth6_br

bridge name     bridge  id              STP enabled interfaces

eth6_br         8000.00e0ed1861ae       no          eth6

                                        vnet0

                                        vnet2

[root@localhost ~]#
```

In the above example, the two virtual interfaces are derived from the eth6_br and are represented as vnet0 and vnet2. Run the following commands to switch off gro and lro capabilities on these interfaces.

command COPY

```
ethtool -K vnet0 gro off

ethtool -K vnet2 gro off

ethtool -K vnet0 lro off

ethtool -K vnet2 lro off
```

Promiscuos Mode

The promiscuos mode has to be enabled for the following features to work:

- L2 mode
- Multicast traffic processing
- Broadcast
- IPV6 traffic
- VMAC
- Dynamic routing

Use the following command to enable the promicuous mode.

command COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
[root@localhost ~]# ifconfig eth6 promisc

[root@localhost ~]# ifconfig eth6

eth6      Link encap:Ethernet  HWaddr 78:2b:cb:51:54:a3

          inet6 addr: fe80::7a2b:cbff:fe51:54a3/64 Scope:Link

          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:9000  Metric:1

          RX packets:142961 errors:0 dropped:0 overruns:0 frame:0

          TX packets:2895843 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:1000

          RX bytes:14330008 (14.3 MB)  TX bytes:1019416071 (1.0 GB)



[root@localhost ~]#
```

Module Required

For better network performance, make sure the vhost_net module is present in the Linux host. To check the existence of vhost_net module, run the following command on the Linux host :

lsmod | grep "vhost_net"

If vhost_net is not yet running, enter the following command to run it:

modprobe vhost_net

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Provisioning the NetScaler Virtual Appliance by using OpenStack

Oct 09, 2016

You can provision a NetScaler vpx instance in an Openstack environment either by using the OpenStack command line interface or the OpenStack dashboard or GUI.

Provisioning a NetScaler instance, optionally involves using data from the config drive. Config drive is a special configuration drive that attaches to the instance when it boots. This configuration drive can be used to pass networking configuration like management IP address, network mask, default gateway etc, which the instance can mount and access before you configure the network settings for the instance. .

When OpenStack provisions a NetScaler instance, it first detects that the instance is booting in an OpenStack environment by reading a specific BIOS string that indicates OpenStack. This string is 'OpenStack Foundation' and for Redhat Linux distributions, the string is stored in /etc/nova/release. This is a standard mechanism that is available in all OpenStack implementations based on KVM hyper-visor platform. The drive should have a specific OpenStack label.

If the config drive is detected, the instance attempts to read the following information from the file name specified in the nova boot command. In the steps mentioned below, the file is referred as userdata:

- Management IP address
- Network mask
- Default gateway

Once the parameters are successfully read, they are populated in the NetScaler stack. This helps in managing the instance remotely. If the parameters are not read successfully or the config drive is not available, the instance transitions to the default behavior, which is:

- The instance attempts to retrieve the IP address information from DHCP
- If DHCP fails or times-out, the instance comes up with default network configuration (192.168.100.1/16)

Provisioning the NetScaler Virtual Appliance by using OpenStack Using Command Line Interface

Updated: 2015-06-24

You can provision a NetScaler appliance in an OpenStack environment. Provisioning a NetScaler Virtual Appliance on OpenStack involves the following three steps:

1. Extracting the .raw file from the .tgz file
2. Building an OpenStack image from the raw image
3. Provisioning a NetScaler instance

To provision a NetScaler instance in an OpenStack environment, complete the following steps:

1. Extract the .raw file from the .tgz file.
   tar xvzf NSVPX-KVM-10.5-50.9_nc.tgz
   NSVPX-KVM.xml
   NSVPX-KVM-10.5-50.9_nc.raw
   checksum.txt
2. Build an OpenStack image using the .raw file extracted in step 1.
   glance image-create --name="NS-VPX-10-5-50-9" --property hw_disk_bus=ide --is-public=

© 1999-2017 Citrix Systems, Inc. All rights reserved.

true --container-format=bare --disk-format=raw < NSVPX-KVM-10.5-50.9_nc.raw

In the above command, **NS-VPX-10-5-50-9** is the name of the OpenStack image that you want to create. **NS-VPX-10-5-50-9.tgz** is the raw file that was extracted from the .tgz file. The raw file is the input for creating the OpenStack image.

The following illustration provides a sample output for the glance image-create command.

```
+-------------------------+--------------------------------------+
| Property                | Value                                |
+-------------------------+--------------------------------------+
| Property 'hw_disk_bus'  | ide                                  |
| checksum                | e405917fcb079f1f02e556db6bb943a4     |
| container_format        | bare                                 |
| created_at              | 2015-06-15T15:24:09                  |
| deleted                 | False                                |
| deleted_at              | None                                 |
| disk_format             | raw                                  |
| id                      | 76b68cd6-2d8c-41ba-8a4c-f1a7ca556904 |
| is_public               | True                                 |
| min_disk                | 0                                    |
| min_ram                 | 0                                    |
| name                    | NS-VPX-10-5-50-9                     |
| owner                   | 8bbe8551955d416fa348664d77c85ee1     |
| protected               | False                                |
| size                    | 21474836480                          |
| status                  | active                               |
| updated_at              | 2015-06-15T15:28:18                  |
| virtual_size            | None                                 |
+-------------------------+--------------------------------------+
```

3. After an OpenStack image is created, provision the NetScaler virtual appliance instance.
   nova boot --image NS-VPX-10-5-50-9 --config-drive=true --user-data ./userdata.txt
   --flavor m1.medium --nic net-id=b8c5acee-36b7-4517-af0e-80f8729aa82e vpx10_5_u
   In the above command, userdata.txt is the file which contains the details like, IP address, netmask, and default gateway for the NetScaler instance. The userdata file is a user customizable file. NS-VPX-10-5-50-9is the name of the virtual appliance that you want to provision.

   The following illustration gives a sample output of the nova boot command.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
+--------------------------------------+-------------------------------------------------------------+
| Property                             | Value                                                       |
+--------------------------------------+-------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                      |
| OS-EXT-AZ:availability_zone          | nova                                                        |
| OS-EXT-SRV-ATTR:host                 | -                                                           |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | -                                                           |
| OS-EXT-SRV-ATTR:instance_name        | instance-000000a7                                           |
| OS-EXT-STS:power_state               | 0                                                           |
| OS-EXT-STS:task_state                | scheduling                                                  |
| OS-EXT-STS:vm_state                  | building                                                    |
| OS-SRV-USG:launched_at               | -                                                           |
| OS-SRV-USG:terminated_at             | -                                                           |
| accessIPv4                           |                                                             |
| accessIPv6                           |                                                             |
| adminPass                            | iN7odsyMybX2                                                |
| config_drive                         | True                                                        |
| created                              | 2015-06-15T15:33:08Z                                        |
| flavor                               | m1.medium (3)                                               |
| hostId                               |                                                             |
| id                                   | c05fe2fe-f844-484c-89a7-80c09d62abe5                        |
| image                                | NS-VPX-10-5-50-9 (76b68cd6-2d8c-41ba-8a4c-f1a7ca556904)     |
| key_name                             | -                                                           |
| metadata                             | {}                                                          |
| name                                 | VPX_10_5                                                    |
| os-extended-volumes:volumes_attached | []                                                          |
| progress                             | 0                                                           |
| security_groups                      | default                                                     |
| status                               | BUILD                                                       |
| tenant_id                            | 8bbe8551955d416fa348664d77c85ee1                            |
| updated                              | 2015-06-15T15:33:08Z                                        |
| user_id                              | 212bbcb674bb45e7897abb1a006028d0                            |
+--------------------------------------+-------------------------------------------------------------+
```

The following illustration shows a sample of the **userdata.txt** file. The values within the <PropertySection> </PropertySection> tags are the values which is user configurable and holds the information like, IP address, netmask, and default gateway.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1">
<PlatformSection>
<Kind>NOVA</Kind>
<Version>2013.1</Version>
<Vendor>Openstack</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
    <Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>
    <Property oe:key="com.citrix.netscaler.platform" oe:value="vpx"/>
    <Property oe:key="com.citrix.netscaler.orch_env" oe:value="openstack-orch-env"/>
    <Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.0.100"/>
    <Property oe:key="com.citrix.netscaler.mgmt.netmask"  oe:value="255.255.0.0"/>
    <Property oe:key="com.citrix.netscaler.mgmt.gateway"  oe:value="10.1.0.1"/>
</PropertySection>
</Environment>
```

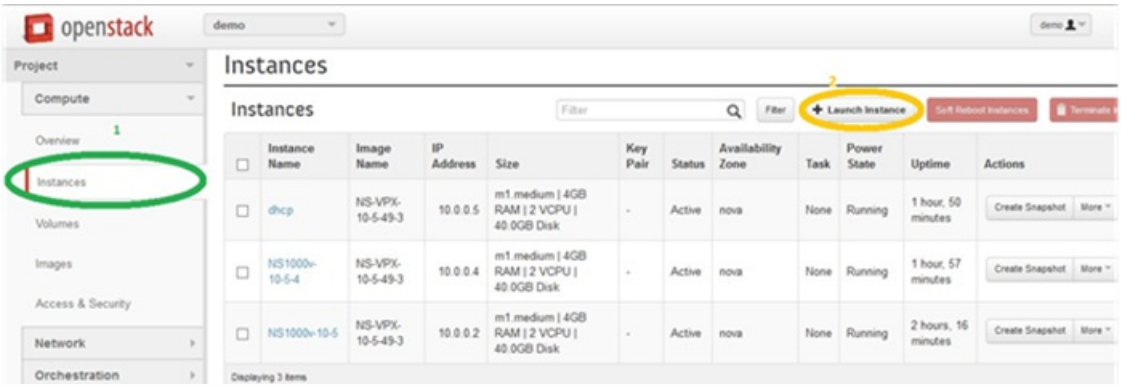
Provisioning the NetScaler Virtual Appliance by using OpenStack Dashboard

Updated: 2015-06-24

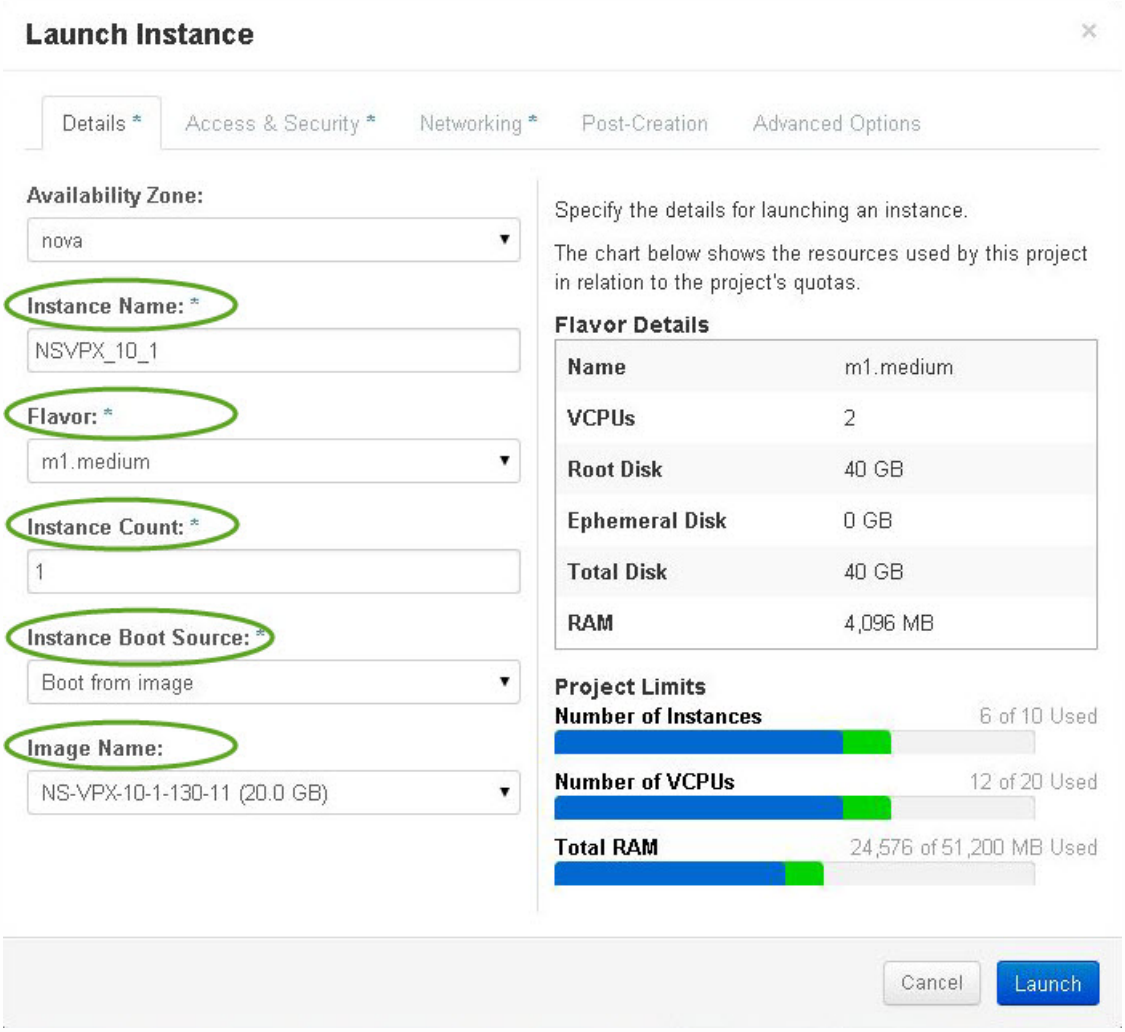
You can provisioning NetScaler in an OpenStack environment using the OpenStack dashboard.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Log in to the OpenStack dashboard.
2. In the Project panel on the left hand side of the dashboard, select Instances.
3. In the Instances panel, click Launch Instance to open the Instance Launching Wizard.



4. In the Launch Instance wizard, fill in the details, like:
   1. Instance Name
   2. Instance Flavor
   3. Instance Count
   4. Instance Boot Source
   5. Image Name



5. Click on the Post Creation tab in the wizard. In the Customization Script, add the content of the userdata file. The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

userdata file contains the IP address, Netmask and Gateway details of the NetScaler instance.

6. Click Launch.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Provisioning the NetScaler Virtual Appliance by using the Virtual Machine Manager

Mar 22, 2016

The Virtual Machine Manager is a desktop tool for managing VM Guests. It enables you to create new VM Guests and various types of storage, and manage virtual networks. You can access the graphical console of VM Guests with the built-in VNC viewer and view performance statistics, either locally or remotely.

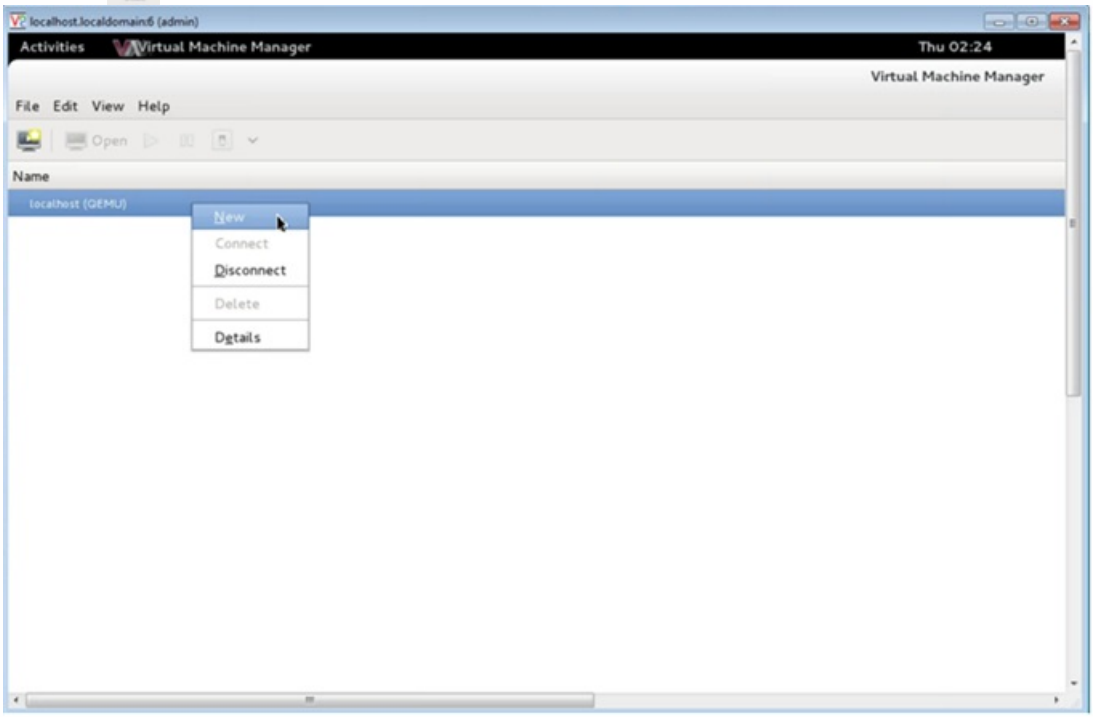
After installing your preferred Linux distribution, with KVM virtualization enabled, you can proceed with provisioning virtual machines.

Provisioning the NetScaler Virtual Appliance using the RAW Image

Using the Virtual Machine Manager, you can provision the NetScaler VPX using the RAW image.

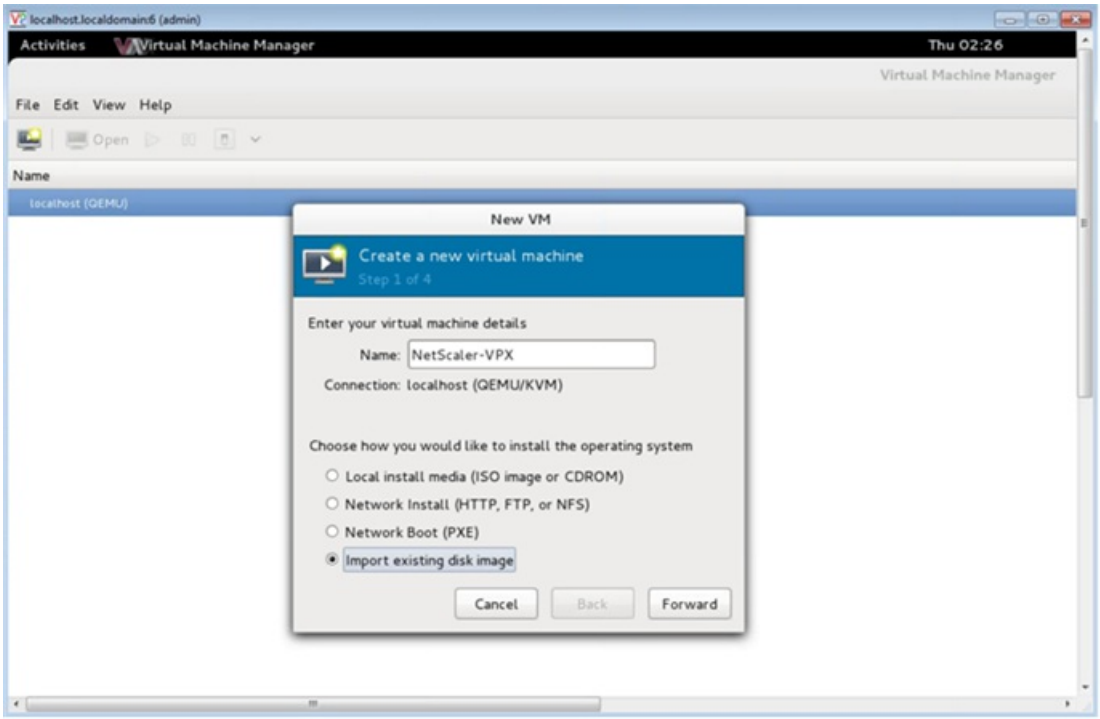
To provision a NetScaler VPX by using Virtual Machine Manager

1. Open the Virtual Machine Manager (**Application** > **System Tools** > **Virtual Machine Manager**) and enter the logon credentials in the **Authenticate** window.
2. Click the icon or right-click **localhost (QEMU)** to create a new NetScaler VPX instance.

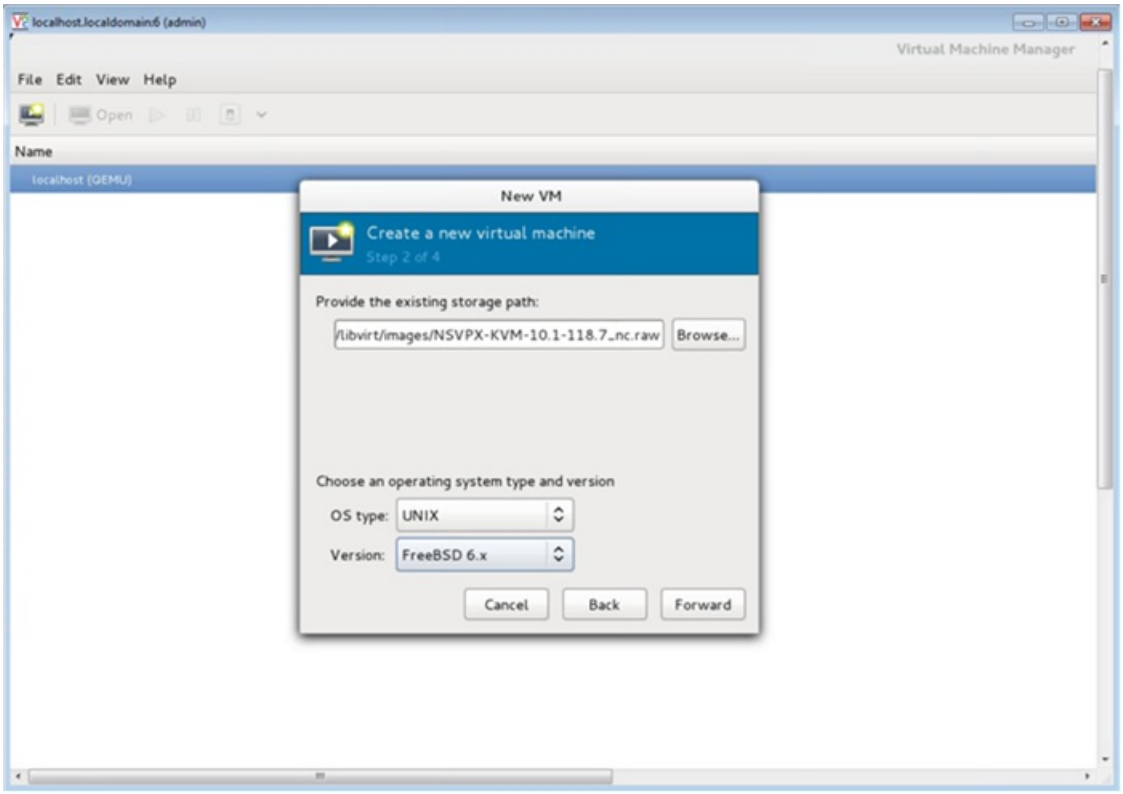


3. In the **Name** text box, enter a name for the new VM (for example, NetScaler-VPX).
4. In the **New VM** window, under "Choose how you would like to install the operating system," select **Import existing disk image**, and then and click **Forward**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

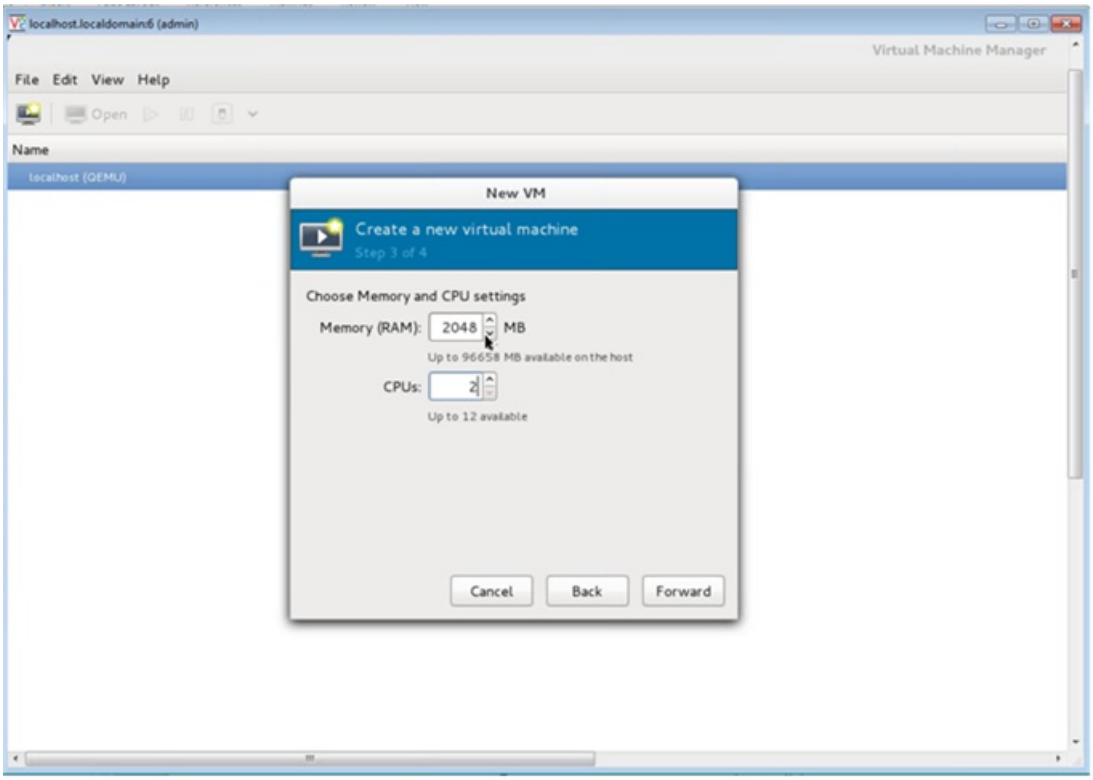


5. In the **Provide the existing storage path** field, navigate the path to the image. Choose the OS type as UNIX and Version as FreeBSD 6.x. Then, click **Forward**.

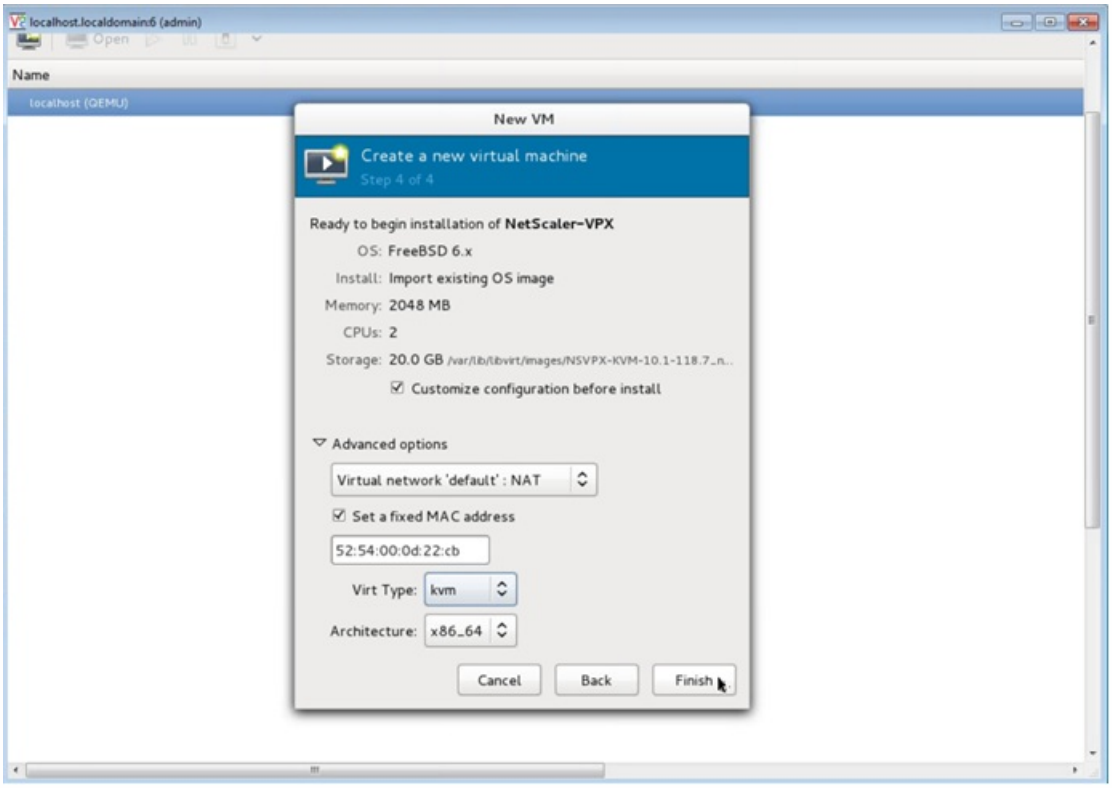


6. Under "Choose Memory and CPU settings," select the following settings, and then click **Forward**:
   - Memory (RAM)— 2048 MB
   - CPUs— 2

© 1999-2017 Citrix Systems, Inc. All rights reserved.



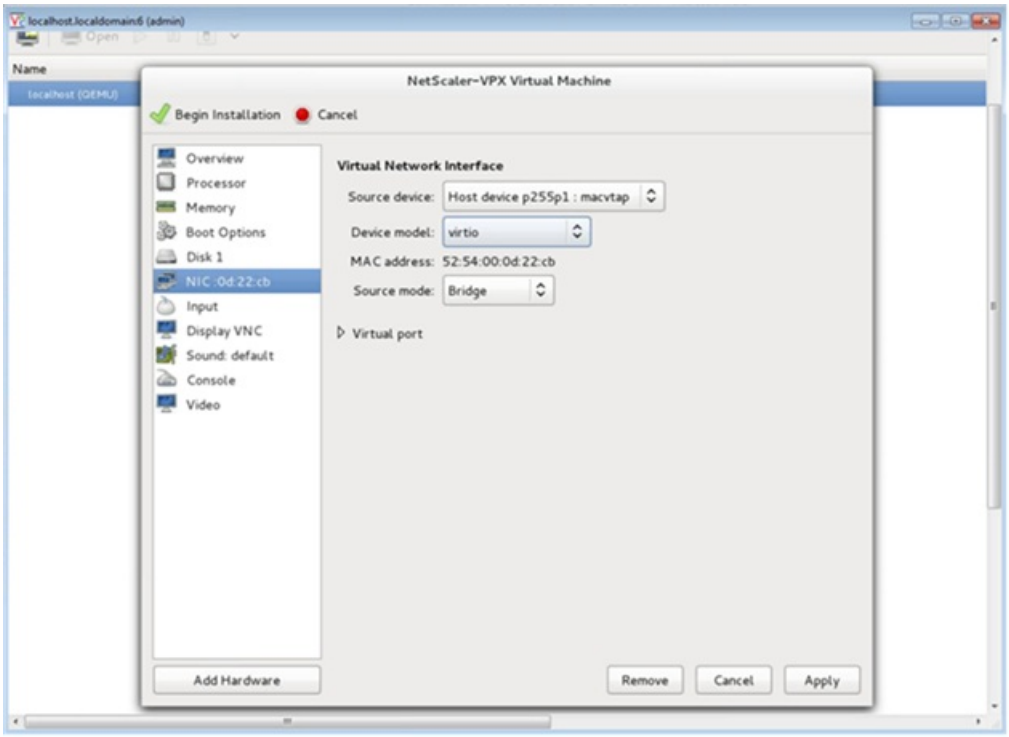
7. Select the **Customize configuration before install** check box. Optionally, under "Advanced options," you can you can customize the MAC address. Make sure the **Virt Type** selected is kvm and the Architecture selected is x86_64. Click **Finish**.



8. Select a NIC and provide the following configuration:
   - Source device— ethX macvtap or Bridge
   - Device model— virtio

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Source mode— Bridge



9. Click **Apply**, and then click **Begin Installation**. After you have provisioned the NetScaler VPX on KVM, you can add additional interfaces

Provisioning the NetScaler Virtual Appliance by using the QCOW2 Image

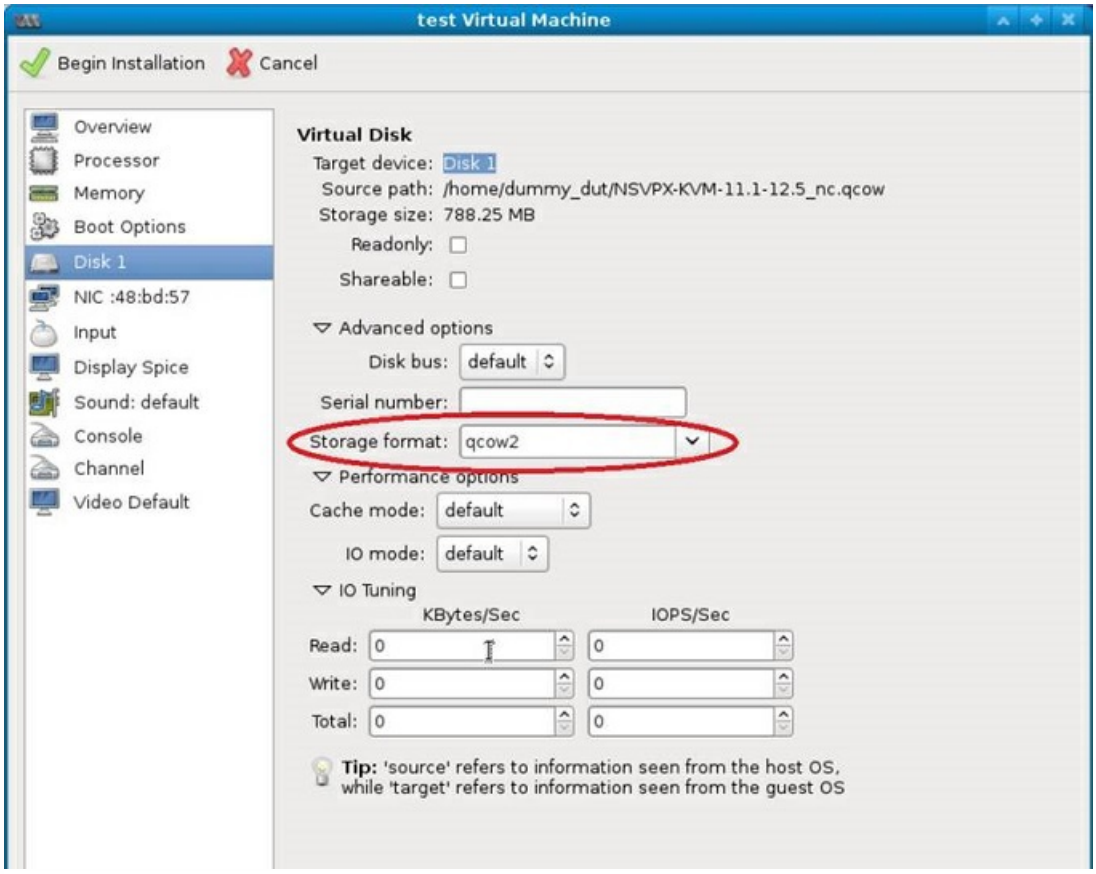
Using the Virtual Machine Manager, you can provision the NetScaler VPX using the QCOW2 image.

Note: You can also convert the NetScaler VPX RAW image to QCOW2 image and provision the NetScaler VPX. For instructions to convert the RAW image to QCOW2, see Converting the RAW Image Format to a QCOW2 Image Format.

To provision the NetScaler VPX using QCOW2 image:

1. Follow **step 1** to **step 8** in Provisioning the NetScaler Virtual Appliance by using the RAW Image.
   Note: Make sure that you select **qcow2** image in **step 5**.
2. Select **Disk 1** and click **Advanced options**. Select **qcow2** from the Storage format drop-down list.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



4. Click **Apply**, and then click **Begin Installation**. After you have provisioned the NetScaler VPX on KVM, you can add additional interfaces.

Adding Additional Interfaces to NetScaler VPX by using Virtual Machine Manager

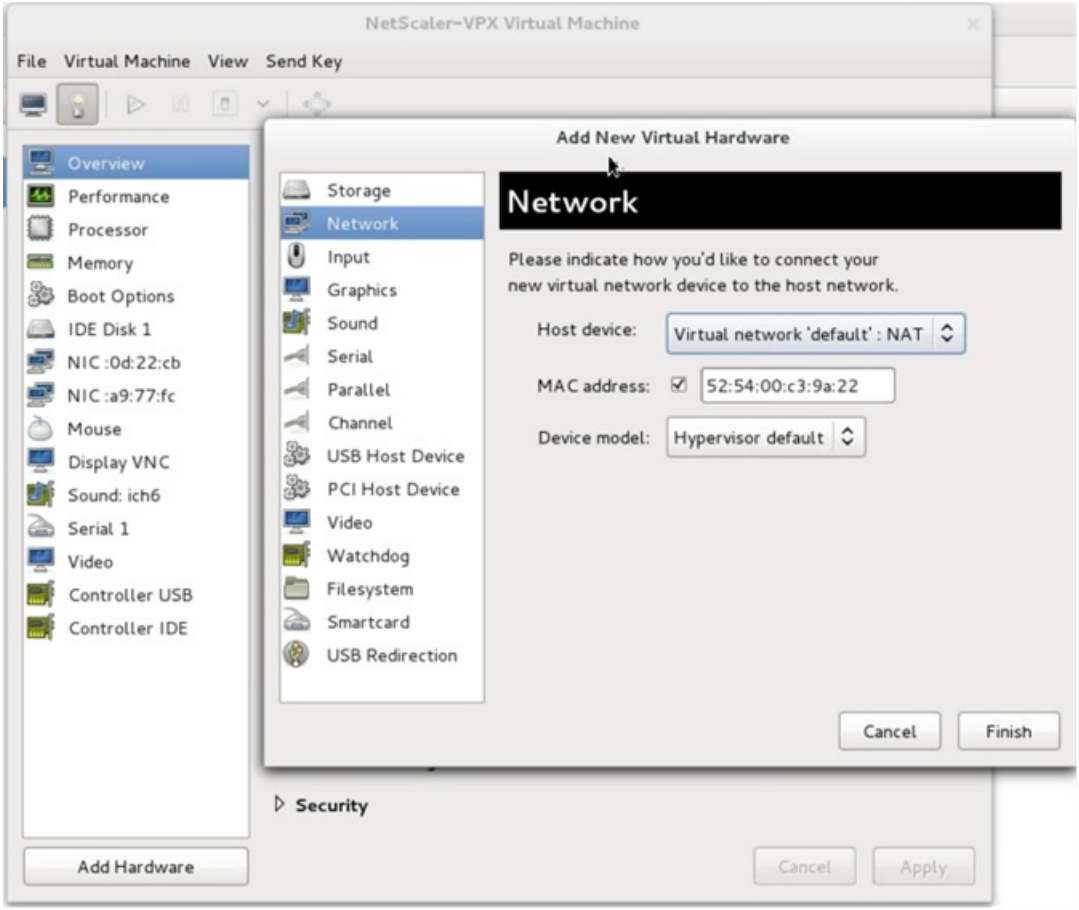
Updated: 2015-03-11

After you have provisioned the NetScaler VPX on KVM, you can add additional interfaces.

To add additional interfaces

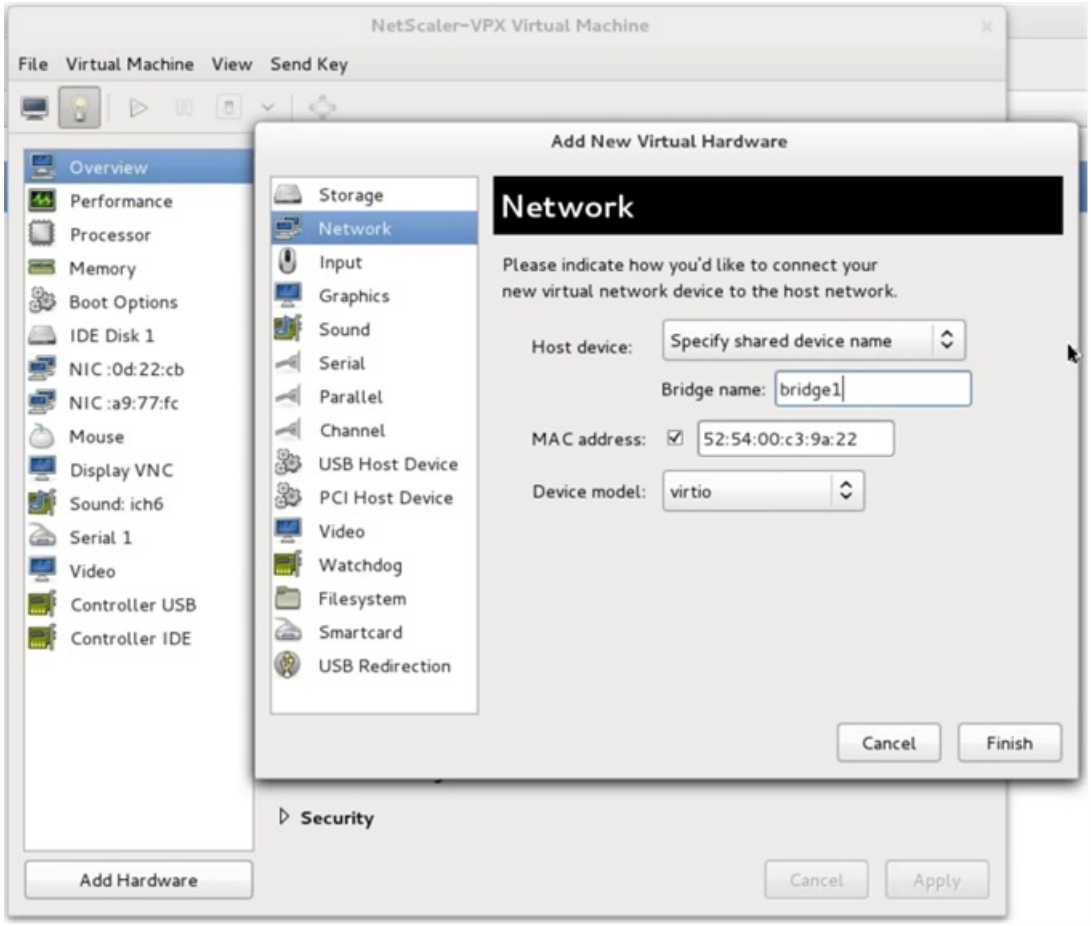
1. Shut down the NetScaler VPX instance running on the KVM.
2. Right-click the VPX instance and choose **Open** from the pop-up menu.
3. Click the icon in the header to view the virtual hardware details.
4. Click **Add Hardware**. In the **Add New Virtual Hardware window**, select **Network** from the navigation menu.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



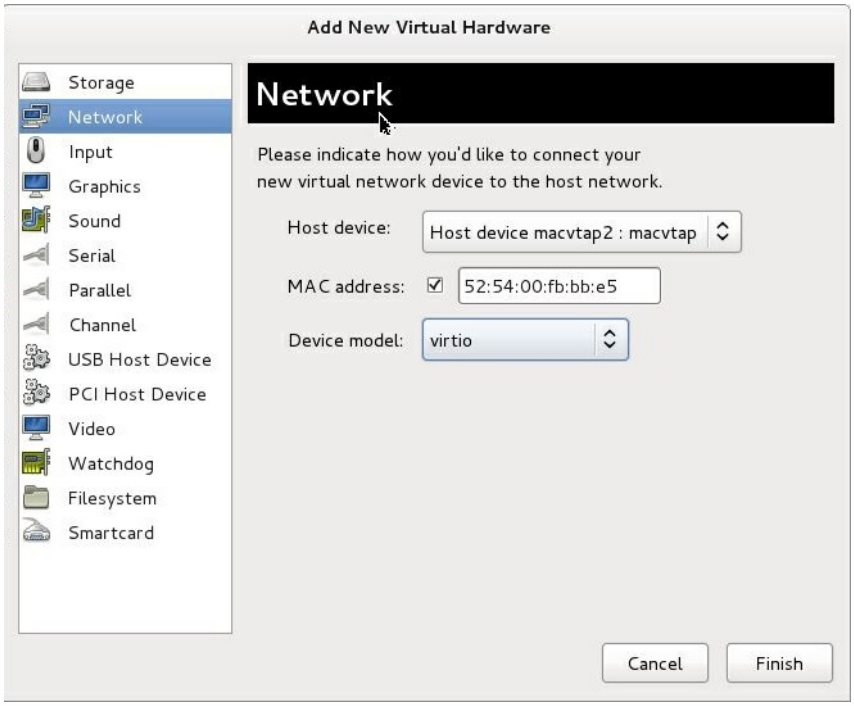
5. In **Host Device** field, select the physical interface type. The host device type can be either Bridge or MacVTap. In case of MacVTap, four modes possible are VEPA, Bridge, Private and Pass-through.
   1. For Bridge
      1. Host device— Select the "Specify shared device name" option.
      2. Provide the Bridge name that is configured in the KVM host.
         Note: Make sure that you have configured a Linux bridge in the KVM host, bound the physical interface to the bridge, and put the bridge in the UP state.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



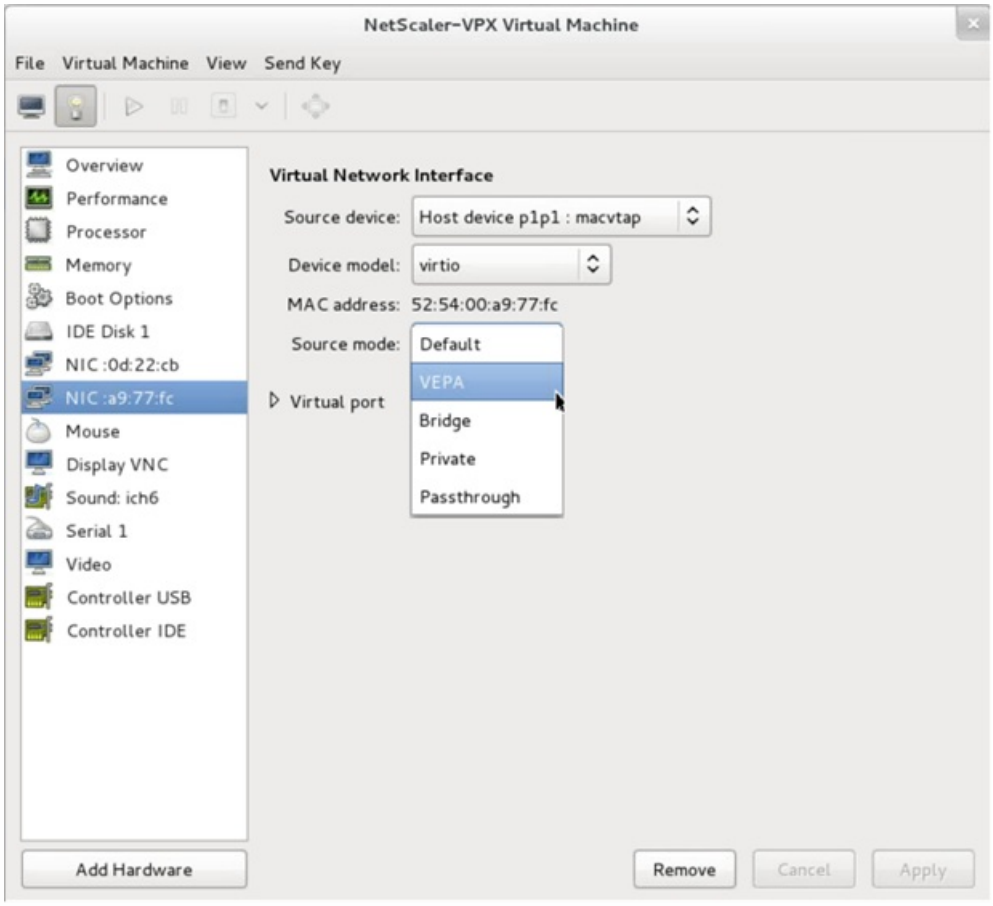
3. Device model—virtio.
4. Click Finish.

2. For MacVTap
   1. Host device—Select the physical interface from the menu.
   2. Device model—virtio.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. Click Finish. You can view the newly added NIC in the navigation pane.



4. Select the newly added NIC and select the Source mode for this NIC. The available modes are VEPA, Bridge, Private, and Passthrough. For more details on the interface and modes, see Source Interface and Modes.
5. Click Apply.

6. Start the NetScaler VPX VM.

Converting the RAW Image Format to a QCOW2 Image Format

You can convert the NetScaler VPX RAW image to QCOW2 image and provision the NetScaler VPX. To convert the RAW image to QCOW2 image. At the command prompt, enter the following command:

qemu-img convert -O qcow2 original-image.raw image-converted.qcow

For example:
qemu-img convert -O qcow2 NSVPX-KVM-11.1-12.5_nc.raw NSVPX-KVM-11.1-12.5_nc.qcow

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Provisioning the NetScaler Virtual Appliance by using the virsh Program

Mar 22, 2016

The virsh program is a command line tool for managing VM Guests. Its functionality is similar to that of Virtual Machine Manager. It enables you to change a VM Guest's status (start, stop, pause, and so on), to set up new Guests and devices, and to edit existing configurations. The virsh program is also useful for scripting VM Guest management operations.

To provision NetScaler VPX by using the virsh program

1. Use the tar command to untar the the NetScaler VPX package. The NSVPX-KVM-*_nc.tgz package contains following components:
   - The Domain XML file specifying VPX attributes [NSVPX-KVM-*_nc.xml]
   - Check sum of NS-VM Disk Image [Checksum.txt]
   - NS-VM Disk Image [NSVPX-KVM-*_nc.raw]

   Example:
   tar -xvzf NSVPX-KVM-10.1-117_nc.tgz
   NSVPX-KVM-10.1-117_nc.xml
   NSVPX-KVM-10.1-117_nc.raw
   checksum.txt
2. Copy the NSVPX-KVM-*_nc.xml XML file to a file named <DomainName>-NSVPX-KVM-*_nc.xml. The <DomainName> is also the name of the virtual machine. Example:
   cp NSVPX-KVM-10.1-117_nc.xml NetScaler-VPX-NSVPX-KVM-10.1-117_nc.xml
3. Edit the <DomainName>-NSVPX-KVM-*_nc.xml file to specify the following parameters:
   - name— Specify the name.
   - mac— Specify the MAC address.
     Note: The domain name and the MAC address have to be unique.
   - sourcefile— Specify the absolute disk-image source path. The file path has to be absolute. You can specify the path of the RAW image file or a QCOW2 image file.
     If you want to specify a RAW image file, specify the disk image source path as shown in the following example:

     Example:
     <name>NetScaler-VPX</name>
     <mac address='52:54:00:29:74:b3'/>
     <source file='/root/NSVPX-KVM-10.1-117_nc.raw'/>

     Specify the absolute QCOW2 disk-image source path and define the driver type as **qcow2**, as shown in the following example:

     Example:
     <name>NetScaler-VPX</name>
     <mac address='52:54:00:29:74:b3'/>
     <driver name ='qemu' type='qcow2'/>
     <source file='/root/NSVPX-KVM-10.1-117_nc.qcow'/>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Edit the <DomainName>-NSVPX-KVM-*_nc.xml file to configure the networking details:
   - source dev— specify the interface.
   - mode— specify the mode. The default interface is **Macvtap Bridge**.

   Example: Mode: MacVTap Bridge Set target interface as ethx and mode as bridge Model type as virtio

```
<interface type='direct'>
  <mac address='52:54:00:29:74:b3'/>
  <source dev='eth0' mode='bridge'/>
  <target dev='macvtap0'/>
  <model type='virtio'/>
  <alias name='net0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
```

   Here, eth0 is the physical interface attached to the VM.
5. Define the VM attributes in the <DomainName>-NSVPX-KVM-*_nc.xml file by using the following command: virsh define <DomainName>-NSVPX-KVM-*_nc.xml Example:
   virsh define NS-VPX-NSVPX-KVM-10.1-117_nc.xml
6. Start the VM by entering following command: virsh start [<DomainName> | <DomainUUID>] Example:
   virsh start NetScaler-VPX
7. Connect the Guest VM through the console virsh console [<DomainName> | <DomainUUID> |<DomainID>] Example:
   virsh console NetScaler-VPX

Adding Additional Interfaces to NetScaler VPX using virsh Program

Updated: 2015-03-09

After you have provisioned the NetScaler VPX on KVM, you can add additional interfaces.

To add additional interfaces

1. Shut down the NetScaler VPX instance running on the KVM.
2. Edit the <DomainName>-NSVPX-KVM-*_nc.xml file using the command: virsh edit [<DomainName> | <DomainUUID>]
3. In the <DomainName>-NSVPX-KVM-*_nc.xml file, append the following parameters:
   1. **For MacVTap**
      - Interface type— Specify the interface type as 'direct'.
      - Mac address— Specify the Mac address and make sure the MAC address is unique across the interfaces.
      - source dev— Specify the interface name.
      - mode— Specify the mode; the modes supported are - Bridge, VEPA, Private, and Pass-through
      - model type— Specify the model type as virtio

      Example:

      Mode: MacVTap Pass-through

      Set target interface as ethx, Mode as bridge, and model type as virtio

```
<interface type='direct'>
    <mac address='52:54:00:29:74:b3'/>
    <source dev='eth1' mode='passthrough'/>
    <model type='virtio'/>
</interface>
```

      Here eth1 is the physical interface attached to the VM.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. **For Bridge Mode**

   Note: Make sure that you have configured a Linux bridge in the KVM host, bound the physical interface to the bridge, and put the bridge in the UP state.

   - Interface type— Specify the interface type as 'bridge'.
   - Mac address— Specify the Mac address and make sure the MAC address is unique across the interfaces.
   - source bridge— Specify the bridge name.
   - model type— Specify the model type as virtio

   Example: Bridge Mode

```
<interface type='bridge'>
    <mac address='52:54:00:2d:43:a4'/>
    <source bridge='br0'/>
    <model type='virtio'/>
 </interface>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing the NetScaler Guest VMs

Sep 03, 2013

You can use the Virtual Machine Manager and the virsh program to perform management tasks such as starting or stopping a VM Guest, setting up new guests and devices, editing existing configurations, and connecting to the graphical console through Virtual Network Computing (VNC).

Managing the NetScaler Guest VMs using Virtual Machine Manager

Updated: 2013-09-04

Listing the VM Guests

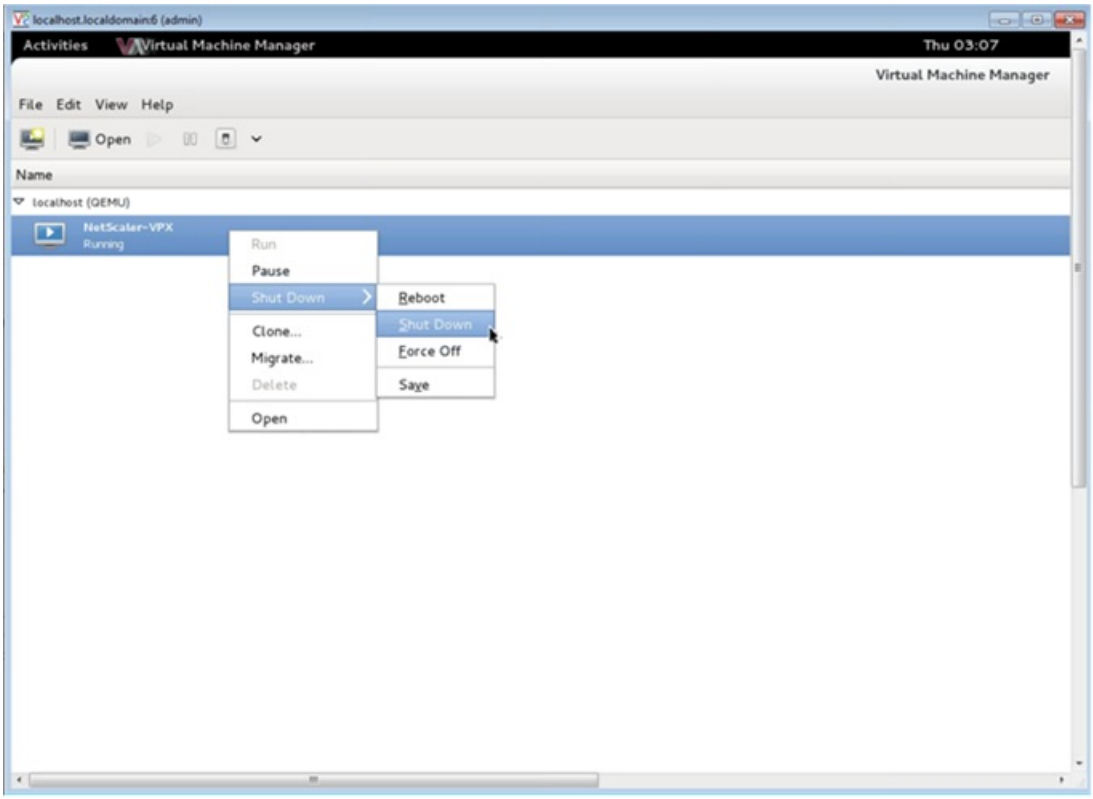
The main Window of the Virtual Machine Manager displays a list of all the VM Guests for each VM host server it is connected to. Each VM Guest entry contains the virtual machine's name, along with its status (Running, Paused, or Shutoff) displayed as icon.

Opening a Graphical Console

Opening a Graphical Console to a VM Guest enables you to interact with the machine like you would with a physical host through a VNC connection. To open the graphical console in the Virtual Machine Manager, right-click the VM Guest entry and select the Open option from the pop-up menu.

Starting and Shutting Down a Guest

You can start or stop a VM Guest from the Virtual Machine Manager. To change the state of the VM, right-click the VM Guest entry and select Run or one of the Shut Down options from the pop-up menu.



Rebooting a Guest

You can reboot a VM Guest from the Virtual Machine Manager. To reboot the VM, right-click the VM Guest entry, and then

© 1999-2017 Citrix Systems, Inc. All rights reserved.

select Shut Down > Reboot from the pop-up menu.

Deleting a Guest

Deleting a VM Guest removes its XML configuration by default. You can also delete a guest's storage files. Doing so completely erases the guest.

1. In the Virtual Machine Manager, right-click the VM Guest entry.
2. Select Delete from the pop-up menu. A confirmation window opens.
   Note: The Delete option is enabled only when the VM Guest is shut down.
3. Click Delete.
4. To completely erase the guest, delete the associated .raw file by selecting the Delete Associated Storage Files check box.

Managing the NetScaler Guest VMs using the virsh Program

Updated: 2013-09-04

Listing the VM Guests and their current states

To use virsh to display information about the Guests

virsh list --all

The command output displays all domains with their states.

Example Output:

```
Id Name          State
----------------------------------
 0 Domain-0      running
 1 Domain-1      paused
 2 Domain-2      inactive
 3 Domain-3      crashed
```

Opening a virsh Console

Connect the Guest VM through the console

virsh console [<DomainID> | <DomainName> | <DomainUUID>]

Example:

virsh console NetScaler-VPX

Starting and Shutting Down a Guest

Guests can be started using the DomainName or Domain-UUID.

virsh start [<DomainName> | <DomainUUID>]

Example

virsh start NetScaler-VPX

To shut down a guest:

virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]

Example:

virsh shutdown NetScaler-VPX

Rebooting a Guest

virsh reboot [<DomainID> | <DomainName> | <DomainUUID>]

Example:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

virsh reboot NetScaler-VPX

Deleting a Guest

To delete a Guest VM you need to shut-down the Guest and un-define the <DomainName>-NSVPX-KVM-*_nc.xml before you run the delete command.

virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]

virsh undefine [<DomainName> | <DomainUUID>]

Example:

virsh shutdown NetScaler-VPX

virsh undefine NetScaler-VPX

Note: The delete command doesn't remove disk image file which needs to be removed manually.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing NetScaler VPX on AWS

May 29, 2017

You can now launch an instance of Citrix® NetScaler VPX within Amazon Web Services (AWS). NetScaler VPX is available as an Amazon Machine Image (AMI) in AWS marketplace. NetScaler VPX on AWS enables customers to leverage AWS Cloud computing capabilities and use NetScaler load balancing and traffic management features for their business needs. NetScaler on AWS supports all the traffic management features of a physical NetScaler appliance. NetScaler instances running in AWS can be deployed as standalone instances or in HA pairs.

How NetScaler VPX on AWS Works

AWS offers different types of web services, such as Amazon Simple Storage Services (S3), Amazon Elastic Cloud Compute (EC2), and Amazon Virtual Private Cloud (VPC). Amazon VPC allows you to run AWS resources (for example, EC2 instances) in a private, virtual network. Amazon EC2 instances are available as instance types that map to hardware archetypes on the basis of factors such as number of EC2 Compute Units (ECU), number of virtual cores, and memory size.

The NetScaler VPX AMI is packaged as an EC2 instance that is launched within an AWS VPC. The VPX AMI instance requires a minimum of 2 virtual CPUs and 2 GB of memory. An EC2 instance launched within an AWS VPC can also provide the multiple interfaces, multiple IP addresses per interface, and public and private IP addresses needed for VPX configuration. Currently, on Amazon AWS, VPX can be launched only within a VPC, because each VPX instance requires at least three IP addresses. (Although VPX on AWS can be implemented with one or two elastic network interfaces, Citrix recommends three network interfaces for a standard VPX on AWS installation.) AWS currently makes multi-IP functionality available only to instances running within an AWS VPC. A VPX instance in a VPC can be used to load balance servers running in EC2 instances.

An Amazon VPC allows you to create and control a virtual networking environment, including your own IP address range, subnets, route tables, and network gateways.

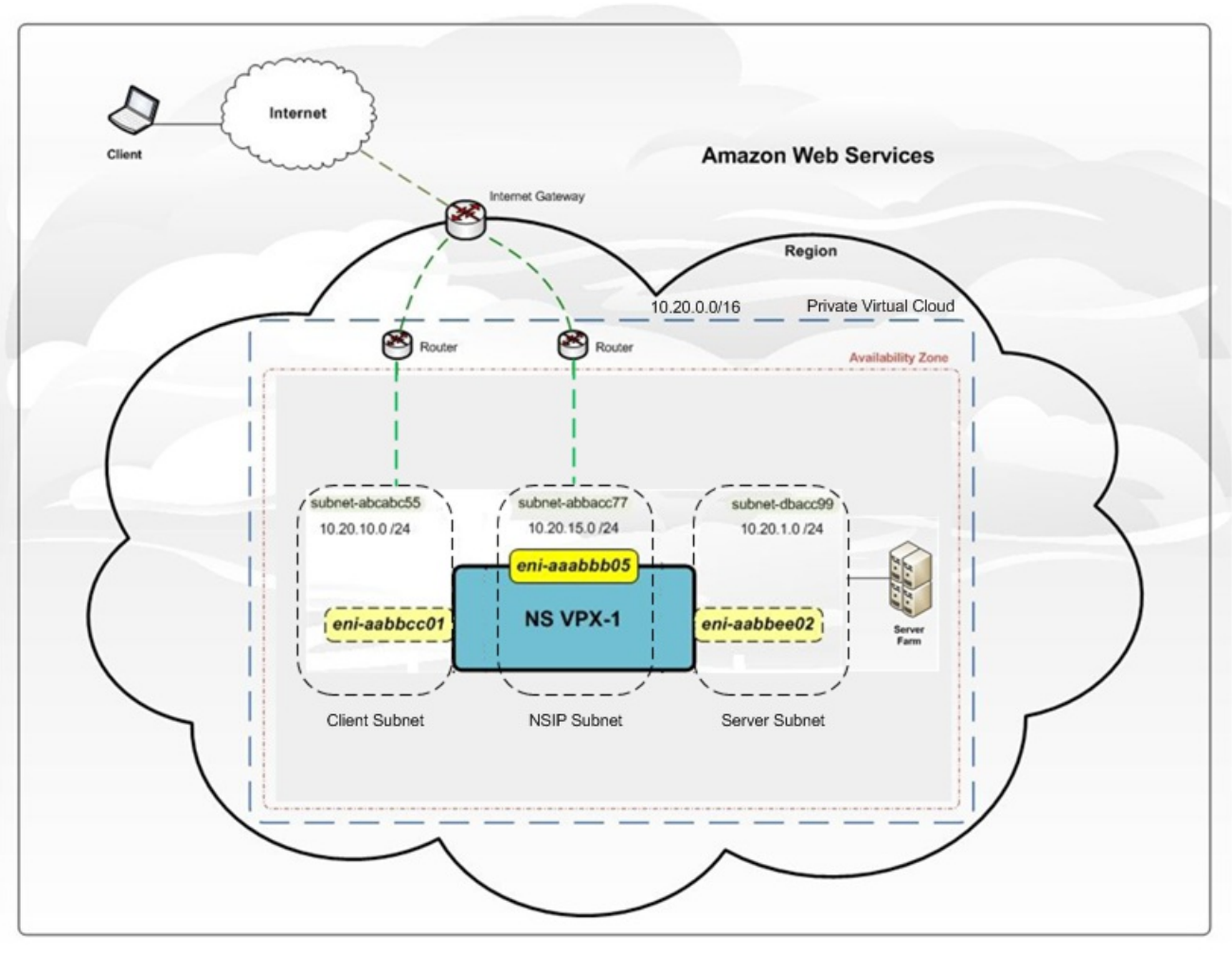
Note: By default, you can create up to 5 VPC instances per AWS region for each AWS account. You can request higher VPC limits by submitting Amazon's request form (http://aws.amazon.com/contact-us/vpc-request/).

VPX on AWS Architecture

An EC2 instance of NetScaler VPX (AMI image) is launched within the AWS VPC. The following figure shows a typical VPX on AWS deployment.

Figure 1. VPX on AWS Architecture

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The figure shows a simple topology of an AWS VPC with a NetScalerVPX deployment. The AWS VPC has:

1. A single Internet gateway to route traffic in and out of the VPC.
2. Network connectivity between the Internet gateway and the Internet.
3. Three subnets, one each for management, client, and server.
4. Network connectivity between the Internet gateway and the two subnets (management and client).
5. A single NetScaler VPX deployed within the VPC. The VPX instance has three Elastic Network Interfaces (ENIs), one attached to each subnet.

Supported EC2 instances

The NetScaler AMI can be launched on any of the following EC2 instance types:

- m3.large
- m3.xlarge
- m3.2xlarge

For more information about Amazon EC2 instances, see:
http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/instance-types.html

ENI Support

The following table lists the EC2 instance types and corresponding number of supported ENIs and number of private IP addresses per ENI.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 1. EC2 Support for ENIs and IP Addresses

Instance Name	Number of ENIs	Private IP Addresses per ENI
m3.large	3	10
m3.xlarge	4	15
m3.2xlarge	4	30

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Limitations and Usage Guidelines

May 13, 2014

- The clustering feature is not supported for VPX.
- For HA to work as expected, associate a dedicated NATing device to management Interface or associate EIP to NSIP. For more information on NAT, in the AWS documentation, see NAT Instances.
- Data traffic and management traffic should be segregated by using ENIs belonging to different subnets.
- Only the NSIP address should be present on the management ENI.
- If a NAT instance is used for security instead of assigning an EIP to the NSIP, appropriate VPC level routing changes are required. For instructions on making VPC level routing changes, in the AWS documentation, see "Scenario 2: VPC with Public and Private Subnets."
- A VPX instance can be moved from one EC2 instance type to another (for example, from m3.large to an m3.xlarge).
- For storage options for VPX on AWS, Citrix recommends EBS, because it is durable and the data is available even after it is detached from instance.
- Dynamic addition of ENIs to VPX is not supported. You have to restart the VPX instance to apply the update. Citrix recommends you to stop the standalone or HA instance, attach the new ENI, and then restart the instance.
- You can assign multiple IP addresses to an ENI. The maximum number of IP addresses per ENI is determined by the EC2 instance type, see EC2 Support for ENIs and IP Addresses.
- Citrix recommends that you avoid using the enable and disable interface commands on NetScaler VPX interfaces.
- Due to Amazon AWS limitations, these features are not supported:
  - IPV6
  - Gratuitous ARP(GARP)
  - L2 mode
  - Tagged VLAN
  - Dynamic Routing
  - Virtual MAC (VMAC)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Launching the NetScaler VPX for AWS AMI

May 21, 2017

You can launch a Citrix NetScaler VPX AMI within an Amazon Web Services (AWS) Virtual Private Cloud (VPC) in one of two ways:

1. Using the Amazon GUI and CLI toolkit.
2. Using a Citrix authored CloudFormation template.
3. Using the Amazon 1-Click launch.

Note: The following are the default administrator credentials to access a NetScaler VPX instance:

- Username—nsroot
- Password—The default password for the nsroot account is set to the AWS instance-ID of the NetScaler VPX instance. For a high availability configuration between two NetScaler VPX instances, the nsroot password of the secondary node is set to that of the primary node after the HA configuration synchronization.

Launching NetScaler VPX for AWS by Using the Amazon GUI and CLI toolkit

To launch a NetScaler VPX AMI within an Amazon Web Services (AWS) Virtual Private Cloud (VPC) by using the Amazon GUI and CLI toolkit, you need:

- An AWS account
- An AWS Virtual Private cloud (VPC)
- The AWS API toolkit (if creating a VPX instance with three or more ENIs).
- An IAM account

Creating an AWS Account

To launch a NetScaler VPX AMI in an Amazon Web Services (AWS) Virtual Private Cloud (VPC), you need an AWS account. You can create an AWS account for free at www.aws.amazon.com.

Creating an AWS Virtual Private Cloud (VPC)

Citrix recommends at least three IP addresses for a NetScaler instance. Currently, the only support that AWS provides for instances with multiple IP addresses is for instances within an AWS VPC.

To create an AWS VPC, first launch the AWS GUI console. For instructions for using the AWS GUI console, see http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide/GetStarted.html?r=2900.

To create an AWS VPC

1. Use the VPC with a Single Public Subnet Only option to create a new AWS VPC in an AWS availability zone.
2. Create additional subnets within the AWS VPC. Citrix recommends that you create at least three subnets, of the following types:
   - One subnet for NetScaler management traffic. You place the NetScaler management IP(NSIP) on this subnet.
   - One or more subnets for client-access (user-to-NetScaler) traffic, through which clients connect to one or more virtual IP (VIP) addresses assigned to NetScaler load balancing virtual servers.
   - One or more subnets for the server-access (NetScaler-to-server) traffic, through which your servers connect to NetScaler-owned subnet IP (SNIP) addresses.
     For more information about NetScaler load balancing and virtual servers, virtual IP addresses (VIPs), and subnet IP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

addresses (SNIPs), see: .

Note:

- All subnets should be in the same availability zone.
- You can launch a NetScaler AMI in an AWS VPC with a single subnet. In this configuration, the management traffic, client-side traffic, and server-side traffic all use the same subnet, and high availability (HA) cannot be configured.
- You can launch the NetScaler AMI into an AWS VPC with two subnets. In this configuration, one subnet is used for management traffic, and the other subnet is used for both client-side and server-side traffic. This topology supports NetScaler HA.

3. Create an Internet gateway and attach it to the VPC instance.
4. Create routing tables for all traffic flowing into or out of the VPC. You need routes for access to the NSIP and to any client-facing VIP addresses. Traffic leaving the VPC must be routed through the Internet Gateway of the AWS VPC.
   Note:
   - Make sure that you associate management and client subnets with the routing table.
   - Add a default route to the routing table for the traffic flowing out of the VPC. Set the Destination to 0.0.0.0/0, and the Target as the Internet gateway address.
5. Create a security group and open the required ports.

Setting-up the AWS API Toolkit

The AWS GUI console does not allow you to launch instances with more than two ENIs. For a standard deployment, you have to create at least three ENIs for a VPC instance (though it is possible to launch a NetScaler AMI with one or two ENIs). To create three or more ENIs for a NetScaler instance, you must use the AWS CLI. To use the AWS CLI, you must install the AWS API toolkit.

The AWS API toolkit is available for download at http://aws.amazon.com/developertools/351/. To install the AWS API toolkit, complete the following tasks on a Windows or Linux machine:

1. Download the AWS API Toolkit.
2. Download X.509 certificate files and X.509 private key file.
3. Download the private key.
4. Convert the downloaded private key (.pem file) for SSH connectivity.
5. Configure the AWS API Toolkit environment on your Windows or Linux computer.

To download the AWS API toolkit

1. In a web browser, open the following website: http://aws.amazon.com/developertools/351/.
2. On the Amazon EC2 API Tools page, in the Download section, click Download the Amazon EC2 API Tools.
3. Save the file, ec2-api-tools.zip, to a local disk and use a file compression utility (for example, WinZip) to extract the files.

To download the X.509 certificate file and X.509 private key file

1. In your browser, open the following website: http://aws.amazon.com/.
2. Click My Account/Console, and then click Security Credentials.
3. On the Amazon Web Services Sign in page, use your Amazon account credentials to sign in.
4. On the Security Credentials page, in the Access Credentials section, on the X.509 Certificates tab, click Create a New Certificate.
5. In the X509 Certificate Created dialog box, Click Download Private Key File and save the private key file to a secure folder on your local drive.
6. Click Download X.509 Certificate and save the certificate to a secure folder on your local drive.
7. Click Close.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: The Private Key File can be downloaded only at the time of creating a certificate. However, you can download the certificate at any time after creating it.

To download private key for SSH connectivity

1. In your browser, open the following website: http://aws.amazon.com/ .
2. Click **My Account/Console.**
3. On the **Amazon Web Services Sign in** page, use your Amazon account credentials to sign in.
4. In the **Service** pane, in **Amazon Web Services**, click **EC2**.
5. In the **Navigation** section, in **Network and Security**, click **Key Pairs**.
6. In the **Key Pairs** pane, click **Create Key Pair**.
7. In the **Create Key Pair** dialog box, type the name for key pair and click **Create**.
8. Download the Key Pair to the local disk and click **Close**.

To convert the downloaded private key for SSH connectivity

For SSH connections from a management machine using Putty, you must convert the .pem file (Private Key) into .ppk file. The .ppk file is the private key for SSH connections to the NetScaler VPX instance hosted in the AWS environment. To convert the .pem file to a .ppk file, use the Putty application's PuttyGen utility. Make sure that the key pairs and certificate files are stored in an unshared and secured directory. After the conversion, you can use SSH to securely connect to the management address of the VPX on AWS instance.

To configure the AWS API Toolkit environment on a Windows machine

1. Move the certificate files to an unshared folder (for example, aws-ec2-api-tools).
2. Move the extracted AWS API toolkit folder to the unshared folder (for example, the aws-ec2-api-tools folder created in example in Step 1).
3. Create a batch file to configure the specific AWS environment in the unshared folder (aws-ec2-api-tools if you used the example in the preceding two steps). Following is an example of the batch file. The file location used in this example is C:\aws-vpc-config\ and the file name is set-aws-environment.bat.

```
rem Setup Amazon EC2 Command-Line Tools

set JAVA_HOME="C:\Program Files\Java\jre7\"

set EC2_HOME="C:\aws-ec2-api-tools\"

set PATH=%PATH%;%EC2_HOME%\bin

set EC2_PRIVATE_KEY=C:\aws-ec2-security-files\pk-3T6ACCLBEDGD3O3SMAM7YDI76VP5HXSU.pem

set EC2_CERT=C:\aws-ec2-security-files\cert-3T6ACCLBEDGD3O3SMAM7YDI76VP5HXSU.pem

set EC2_URL=https://<aws-region>.ec2.amazonaws.com
```

4. Open the command prompt and run the batch file. For the file in the above example, type:
   C:\aws-vpc-config> set-aws-environment.bat
5. Run the ec2ver command to verify that the AWS toolkit is installed properly. For example:
   C:\aws-vpc-config>ec2ver 1.5.6.1 2012-06-15

To configure the AWS API Toolkit on a Linux machine

1. Move the certificate files to an unshared folder (for example, aws-ec2-api-tools).
2. Move the extracted AWS API toolkit folder to the unshared folder (for example, the aws-ec2-api-tools folder created in

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

example in Step 1).

3. Create a shell script to configure the specific AWS environment in the unshared folder (aws-ec2-api-tools if you used the example in the preceding two steps). Following is an example of the batch file. In this example, the file location used is C:\aws-vpc-config\ and the file name used is set-aws-environment.bat.

```
# Setup Amazon EC2 Command-Line Tools

export EC2_HOME=~/ec2-api-tools-1.5.6.0

export EC2_URL= https://us-east-1.ec2.amazonaws.com

export PATH=$EC2_HOME/bin:/usr/bin:$/usr/sbin:/usr/local/sbin:/sbin

export EC2_PRIVATE_KEY=~/pk-XOX3NS2UPZL6BGLFO7PM5OGLYBDPBUCB.pem

export EC2_CERT=~/cert-XOX3NS2UPZL6BGLFO7PM5OGLYBDPBUCB.pem

export JAVA_HOME=/usr

export PS1="AWS PROMPT >"
```

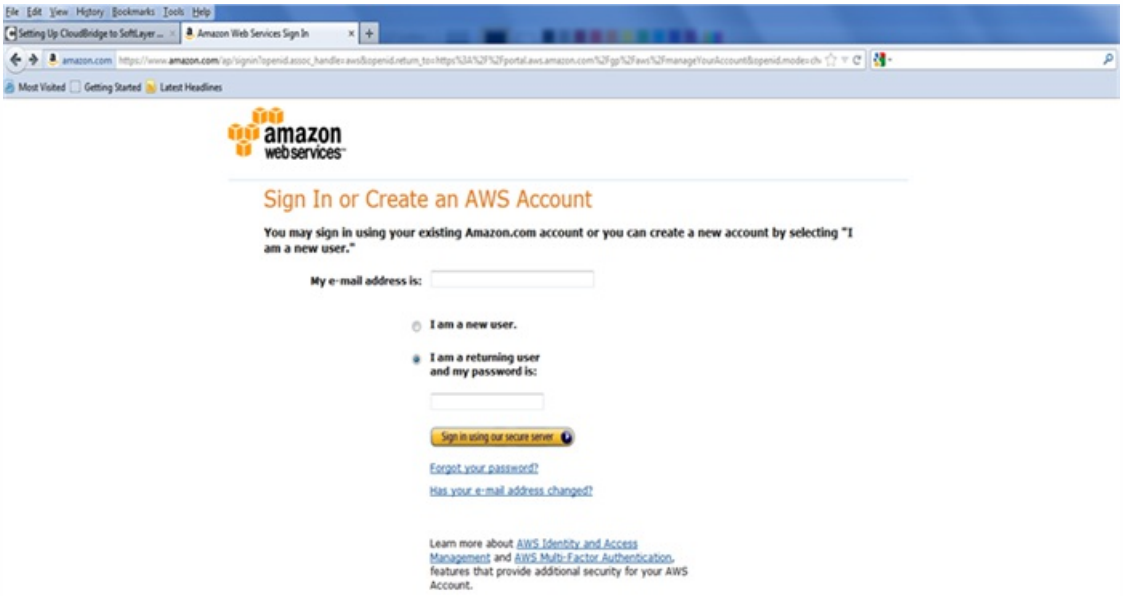
4. Run the ec2ver command to verify that the AWS toolkit is installed properly. For example:

```
AWS PROMPT >ec2ver

1.5.6.1 2012-06-15
```

Creating an IAM Account

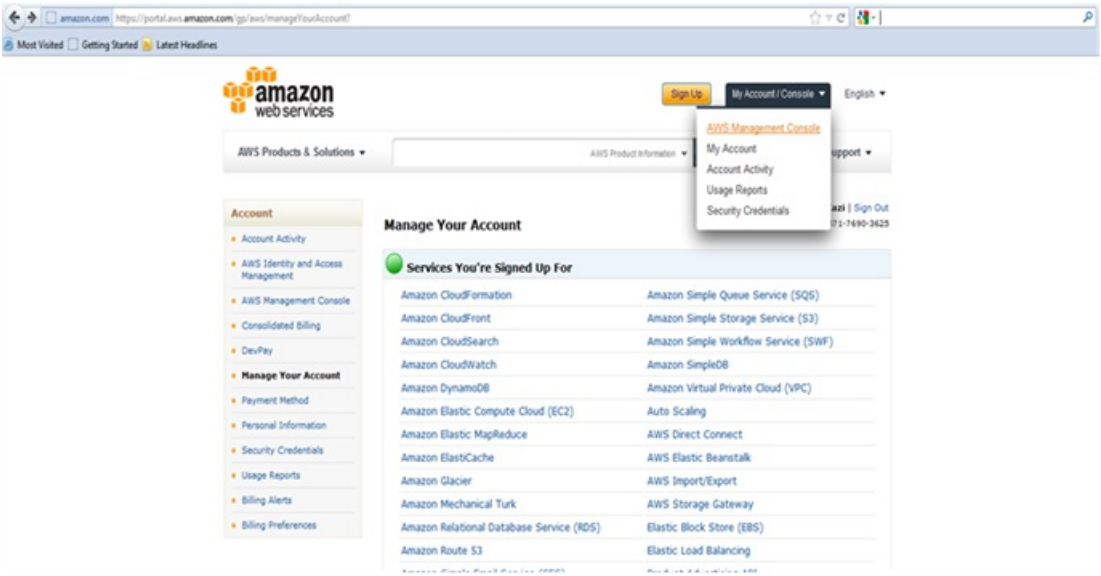
Before you launch the VPX AMI instance, you have to create a new IAM user account with the Access and Secret keys. The Access and Secret key credentials from the new IAM user are required for launching the NetScaler AMI instance. To create a new IAM user for NetScaler, complete the following steps.

1. In a web browser, open the website at www.aws.amazon.com and log on with AWS credentials.

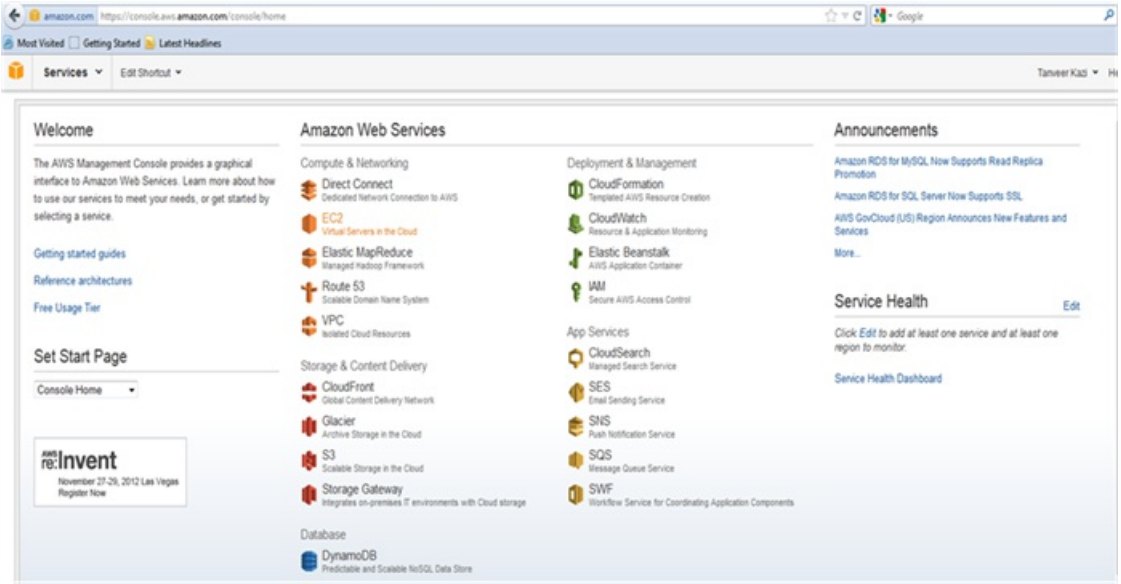


2. Click My Account/Console, and then click AWS Management Console.

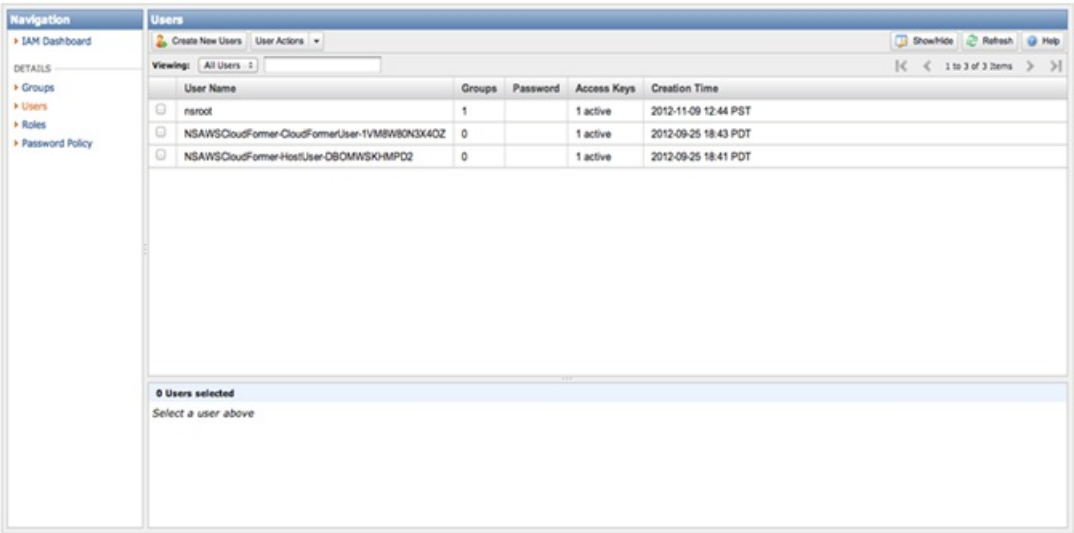
© 1999-2017 Citrix Systems, Inc. All rights reserved.



3. On the Amazon Web Services page, click IAM.



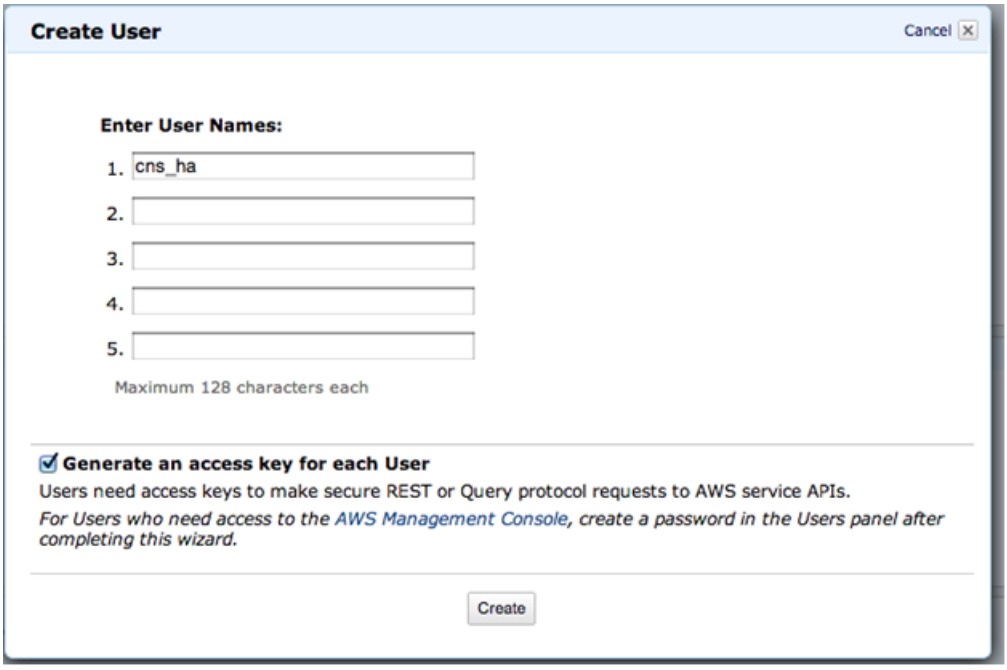
4. In the Navigation pane, click Users, and then click Create New Users.



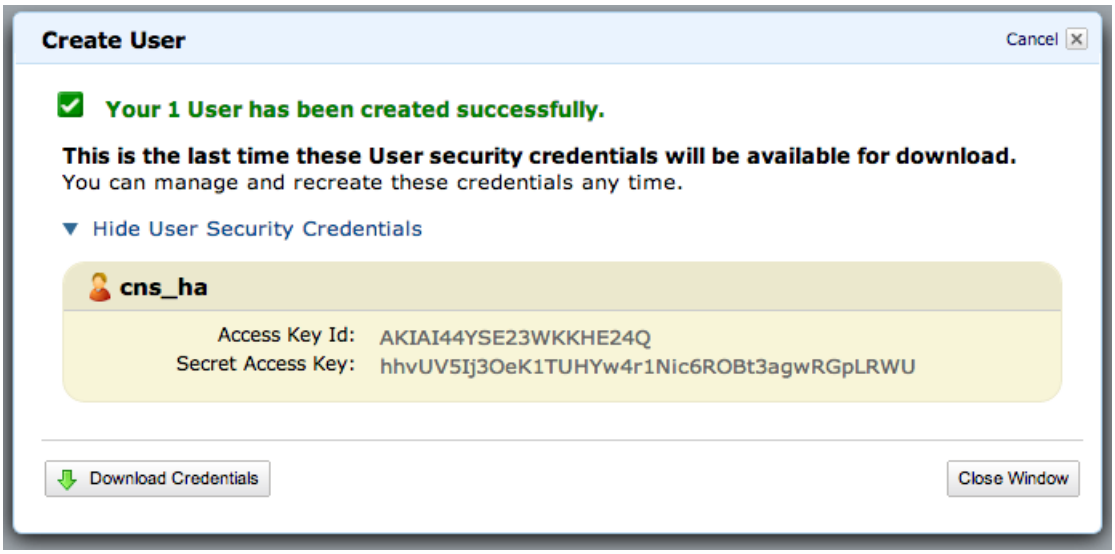
5. In the Create User dialog box, in one of the Enter User Names text boxes, type a user name (for example, cns_ha). Also

© 1999-2017 Citrix Systems, Inc. All rights reserved.

select the Generate an access key for each User check box, and then click Create.

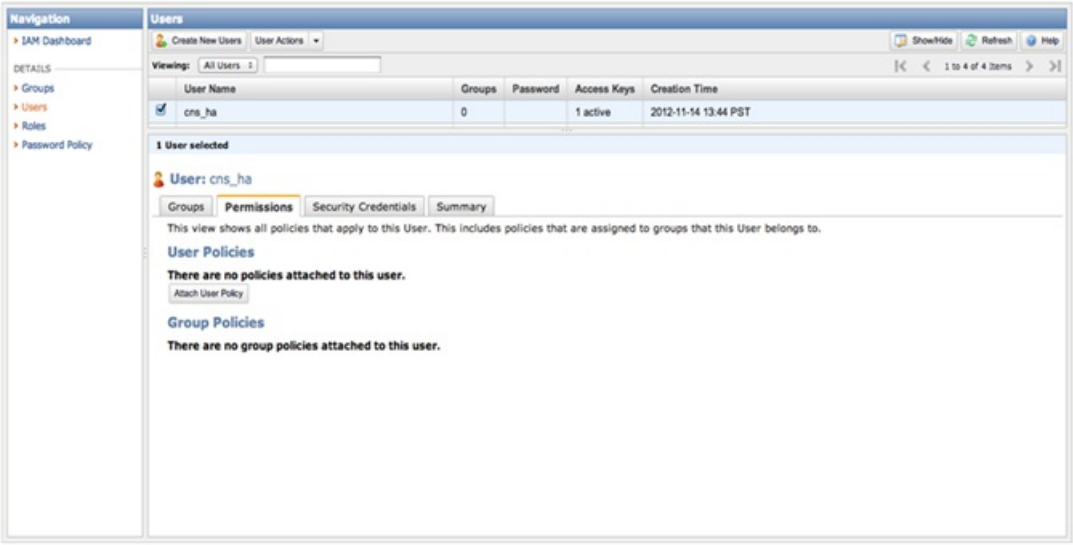


6. After a new IAM user is created, click Download Credentials to download the Access and Secret Keys to a safe location. These keys are required for launching NetScaler AMI. Click Close.
   Note: The Access Key ID and Secret Access Key values are used to create the key-pair file and to launch an instance.

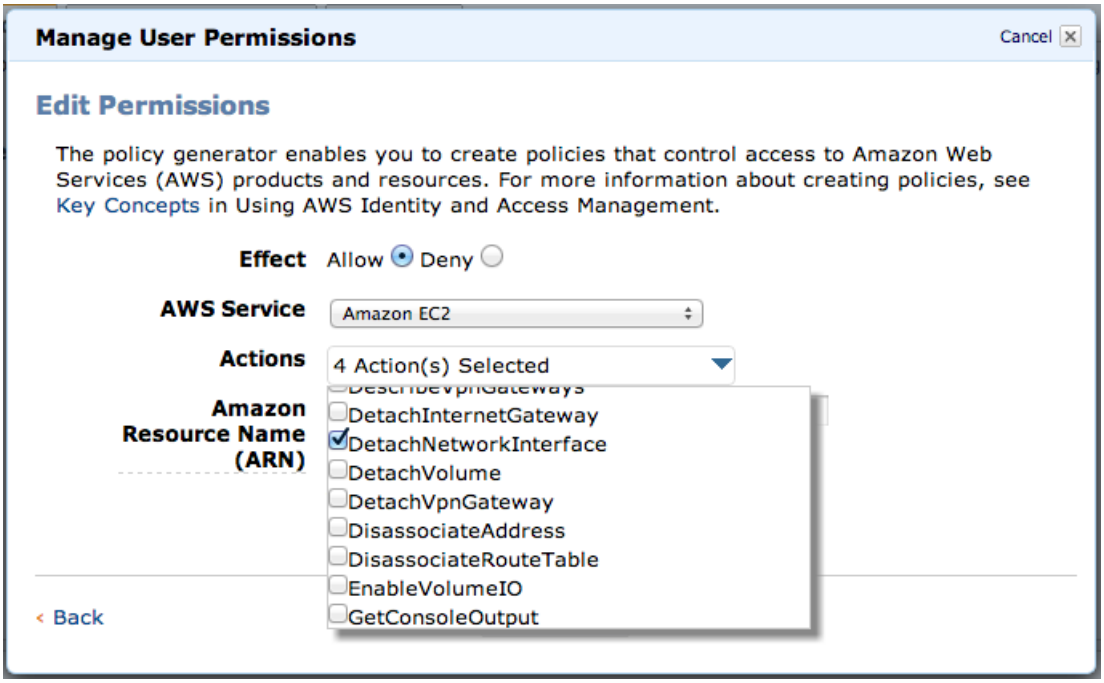


7. In the Users pane, select the newly created IAM user and click the Permissions tab. Then, click Attach User Policy to set policies for the user.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

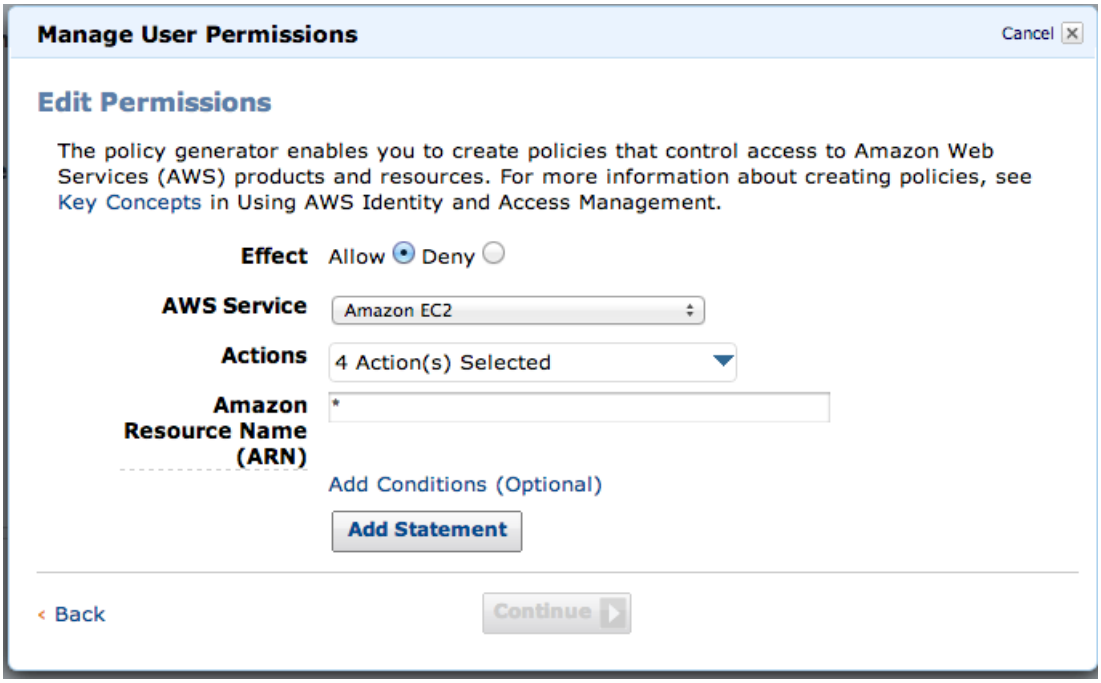


8. In the Manage User Permissions dialog box, next to Effect, select the Allow option. For AWS Service, select Amazon EC2. From the Actions drop-down list, select the following four actions:
   - AttachNetworkInterface
   - DescribeInstances
   - DescribeNetworkInterfaces
   - DetachNetworkInterface

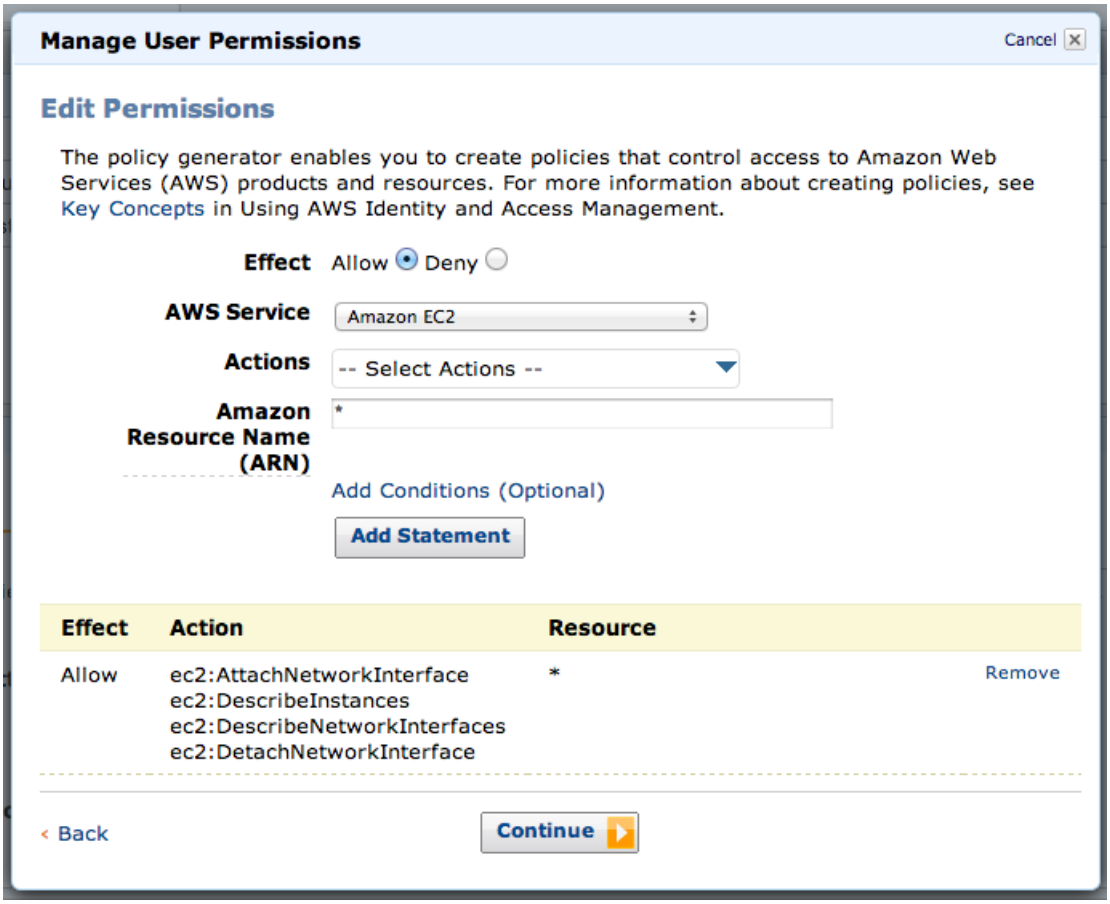


9. Click Add Statement.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

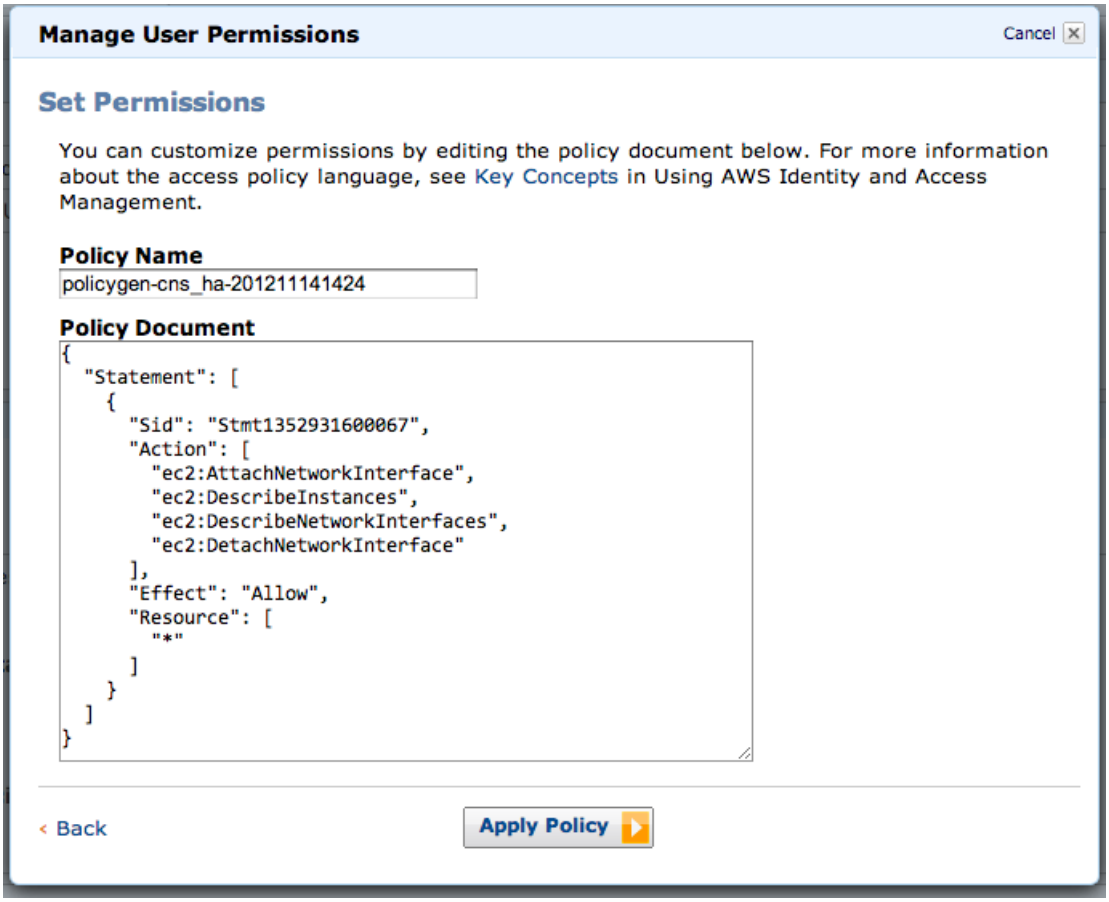


10. Click Continue.



11. Click Apply Policy to set the new permissions for the selected user.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Launching the NetScaler AMI

Use the AWS CLI to launch the NetScaler AMI in an AWS VPC. Use the ec2-run-instances command. For information about the ec2-run-instances command, see http://docs.amazonwebservices.com/AWSEC2/latest/CommandLineReference/ApiReference-cmd-RunInstances.html.

Following are Windows and Linux examples of running the command to launch a single NetScaler instance. The EC2 instance type is m3.large. It is configured with the following entities:

- NetScaler AMI named ami-bd2986d4.
- Three ENIs (named NSIP, CLIENT-SIDE, and SERVER-SIDE) associated with the three subnets (15fa057e, 1547ba7e, and 1547ba7e) within the VPC.
- A single IP address for the NSIP ENI.
- Multiple private IP addresses (for multiple VIPs) on the CLIENT-SIDE ENI.
- Multiple private IPs (for multiple SNIPs) on the SERVER-SIDE ENI.

On a Windows platform:

C:\aws-vpc-config>ec2-run-instances ami-bd2986d4 -n 1 -t m1.large -k keyPairName -f access-secret-key-file -a :0:subnet-15fa057e:"NSIP":10.20.15.21 -a :1:subnet-1547ba7e:"CLIENT-SIDE":10.20.10.21::::"10.20.10.22,10.20.10.23,10.20.10.24,10.20.10.25,10.20.10.26,10.20.10.27,10.20.10.28,10.20.10.29,10.20.10.30" -a :2:subnet-cc47baa7:"SERVER-SIDE":10.20.1.21::::"10.20.1.22,10.20.1.23,10.20.1.24,10.20.1.25,10.20.1.26,10.20.1.27,10.20.1.28,10.20.1.29,10.20.1.30"
Note: The access-secret-key-file file contains the access and secret keys.

On a Linux platform:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

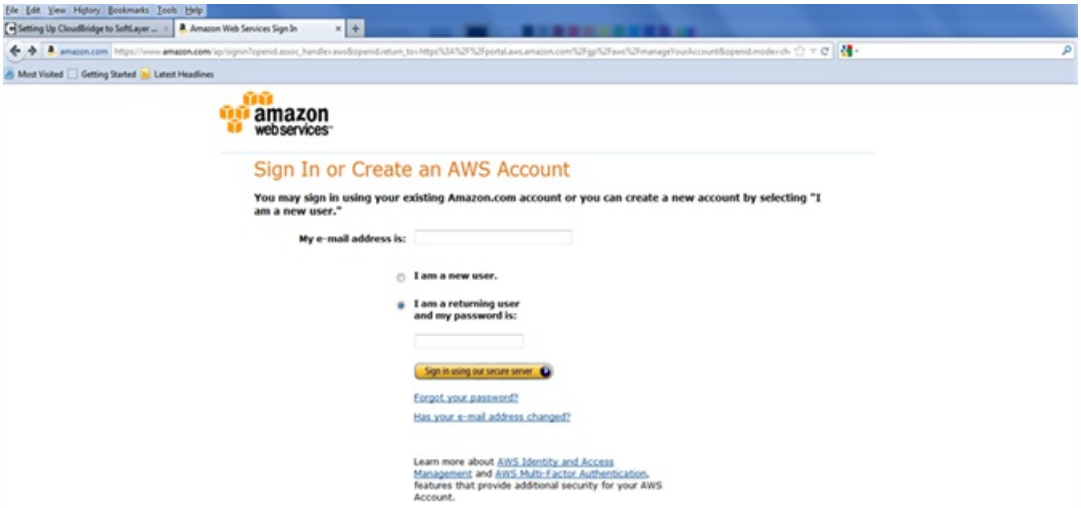
AWS PROMPT > ec2-run-instances ami-bd2986d4 -n 1 -t m1.large -k keyPairName -f access-secret-key-file -a :0:subnet-15fa057e:"NSIP":10.20.15.21 -a :1:subnet-1547ba7e:"CLIENT-SIDE":10.20.10.21::::10.20.10.22,10.20.10.23,10.20.10.24,10.20.10.25,10.20.10.26,10.20.10.27,10.20.10.28,10.20.10.29,10.20.10.30 -a :2:subnet-cc47baa7:"SERVER-SIDE":10.20.1.21::::10.20.1.22,10.20.1.23,10.20.1.24,10.20.1.25,10.20.1.26,10.20.1.27,10.20.1.28,10.20.1.29,10.20.1.30
Note: The access-secret-key-file file contains the access and secret keys.

The command returns the instance ID and the associated information. You can see the instance running within your AWS GUI Console.

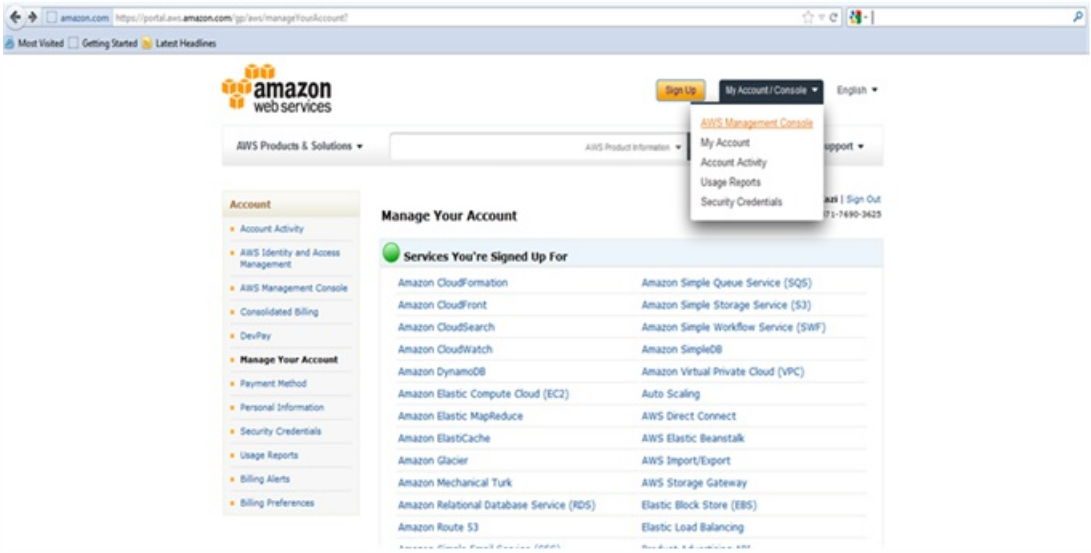
Note: Make sure that the environment variable EC2_URL points to the region where you want to launch the VPX instance.

To access the EC2 instance

1. In a web browser, open the website at www.aws.amazon.com and log on with AWS credentials.

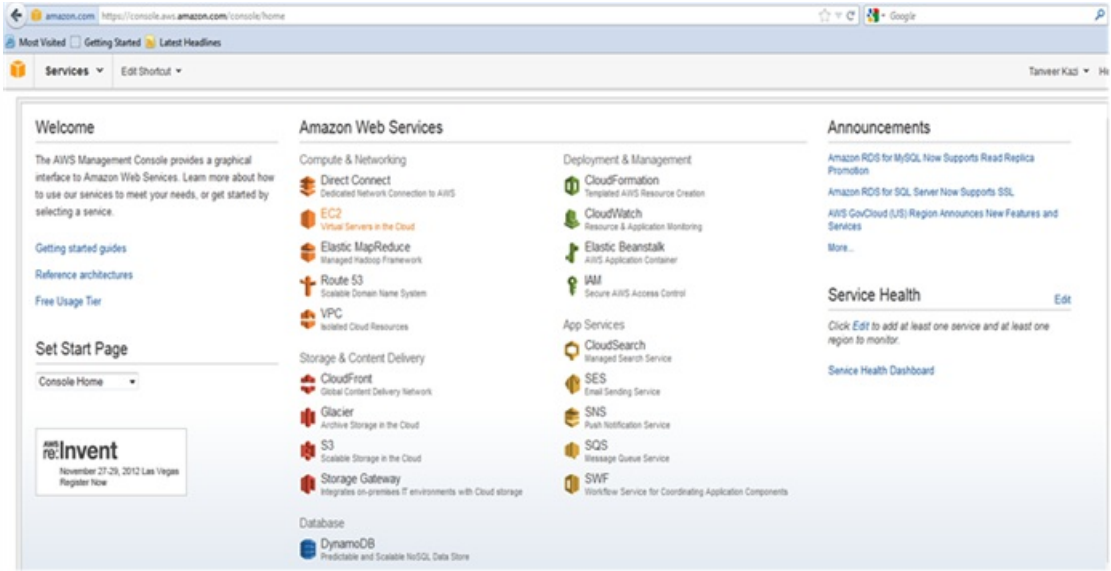


2. Click My Account/Console, and then click AWS Management Console.

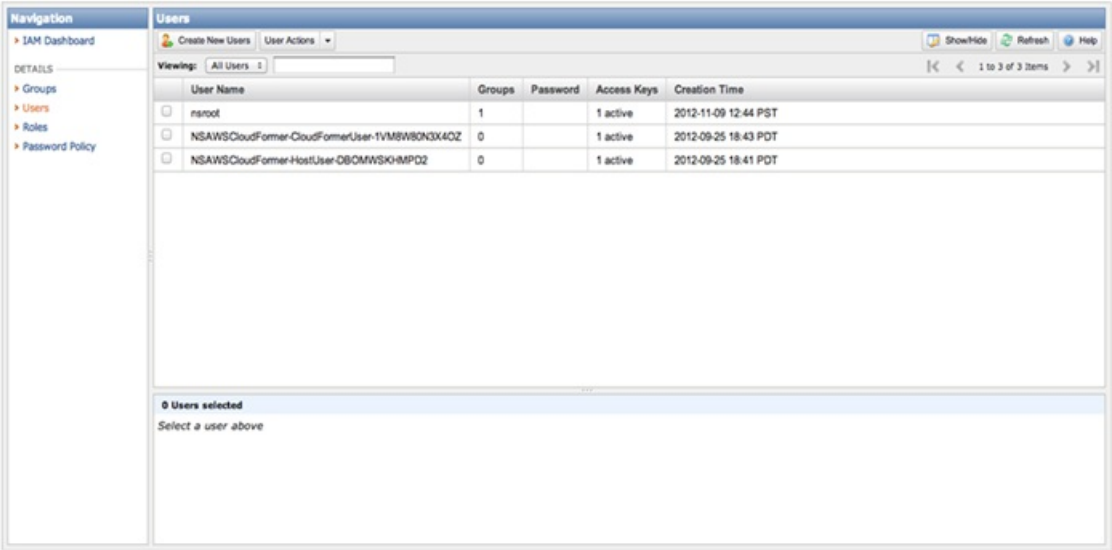


3. On the Amazon Web Services page, click EC2.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



4. On the Amazon EC2 Console Dashboard page, in the Navigation pane, click Instances and verify that all of the NetScaler VPX instances are configured with the IP addresses that you specified when you used the ec2-run-instances command. Note: The VPX instance or instances can take from five to ten minutes to start running.



The ec2-run-instances command does not allow associating AWS elastic IP with an ENI. To associate one or more EIPs with an ENI in the Navigation pane, in the NETWORK & SECURITY area, click Elastic IPs and associate EIPs with Private IP addresses for any of the VIPs that need to be externally routable.

You must also associate the instance ENIs with appropriate security groups. Go to the Network Interfaces section, right-click on the individual ENI, and select the Change Security Groups option. You can then associate a proper VPC security group.

Using the Citrix CloudFormation Template to launch NetScaler VPX for AWS

Citrix also provides a CloudFormation template that can be used to automate NetScaler instance launch. The tool requires an existing VPC environment. It launches a NetScaler instance with three ENIs. Therefore, to use the CloudFormation template, make sure that you have the following:

1. AWS account

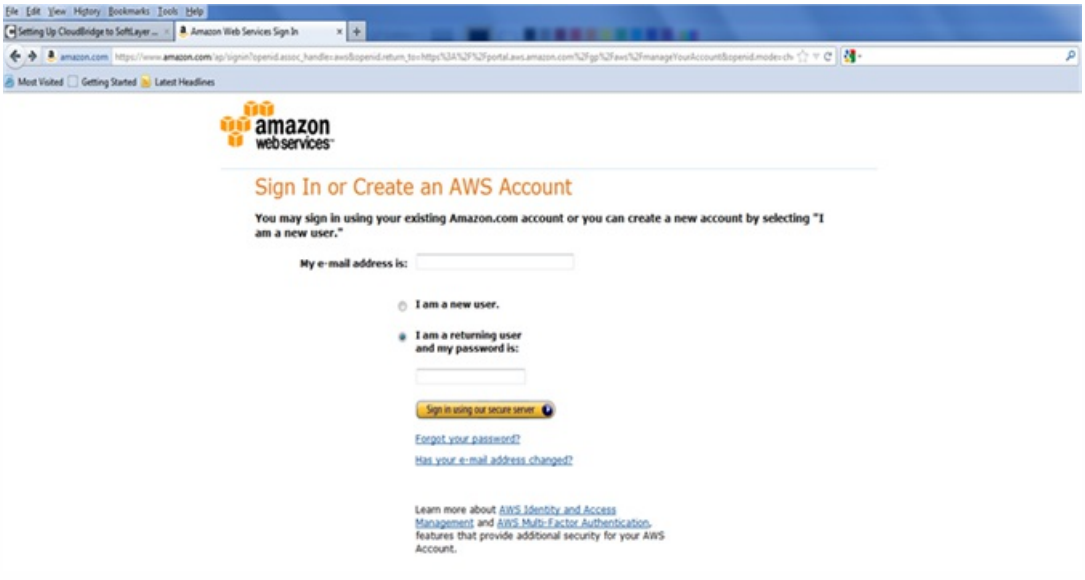
© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. AWS VPC
3. Three subnets within the VPC
4. A security group to use for the NetScaler instances ENIs

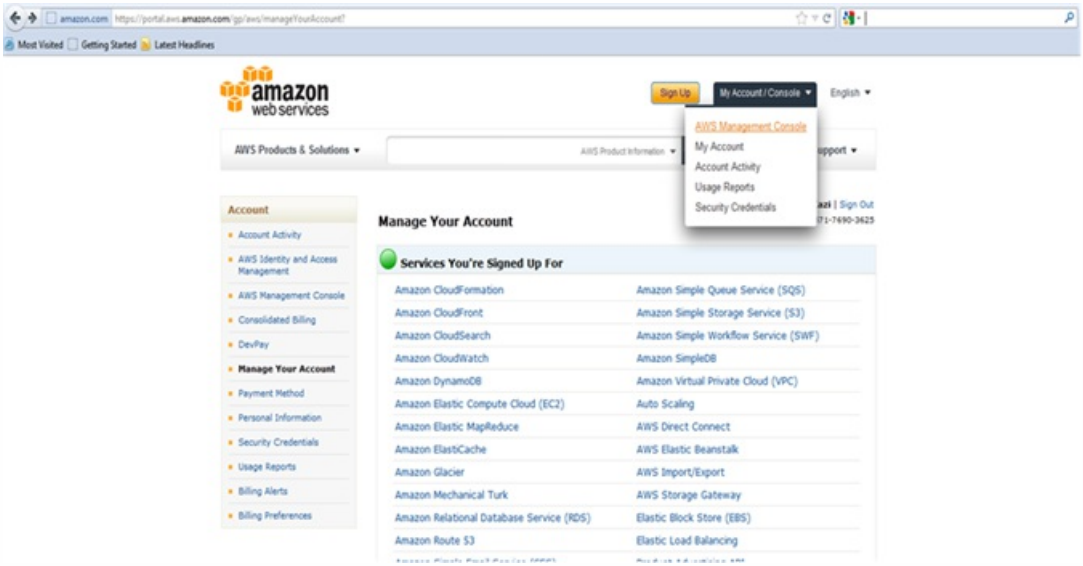
Refer to Creating an AWS Virtual Private Cloud (VPC) for information about how to configure subnets and security groups within a VPC. After configuring the required subnets and security groups, you can launch the NetScaler VPX AMI in AWS VPC. The CloudFormation tool provides functionality to launch a single NetScaler VPX instance or, to create a high availability environment, a pair of NetScaler VPX instances.

Launching a single NetScaler VPX instance in AWS

1. In a web browser, open the website at www.aws.amazon.comand log on with AWS credentials.

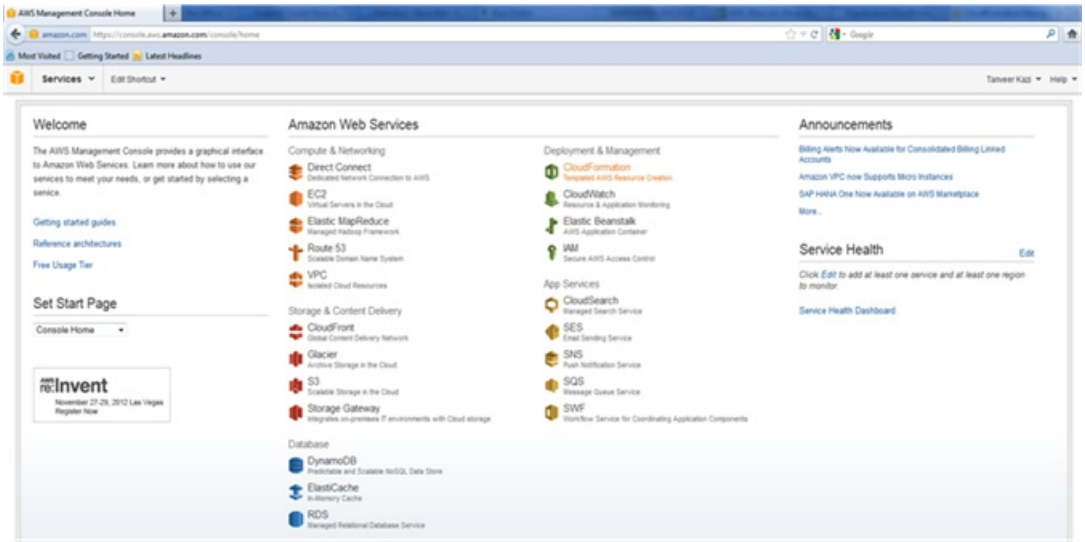


2. Click My Account/Console, and then click AWS Management Console.

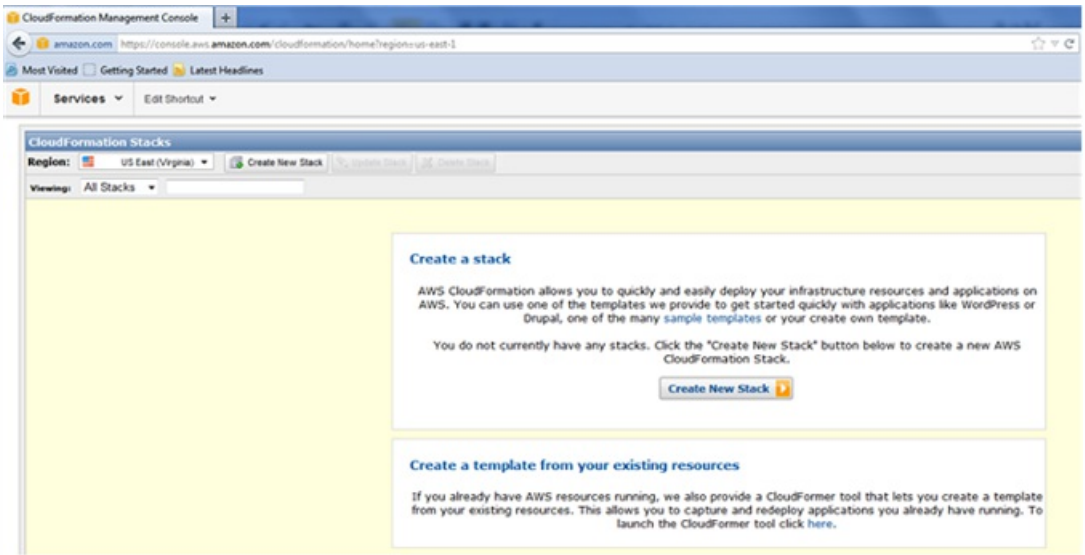


3. On the Amazon Web Services page, click Cloud Formation in the Deployment & Management section.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



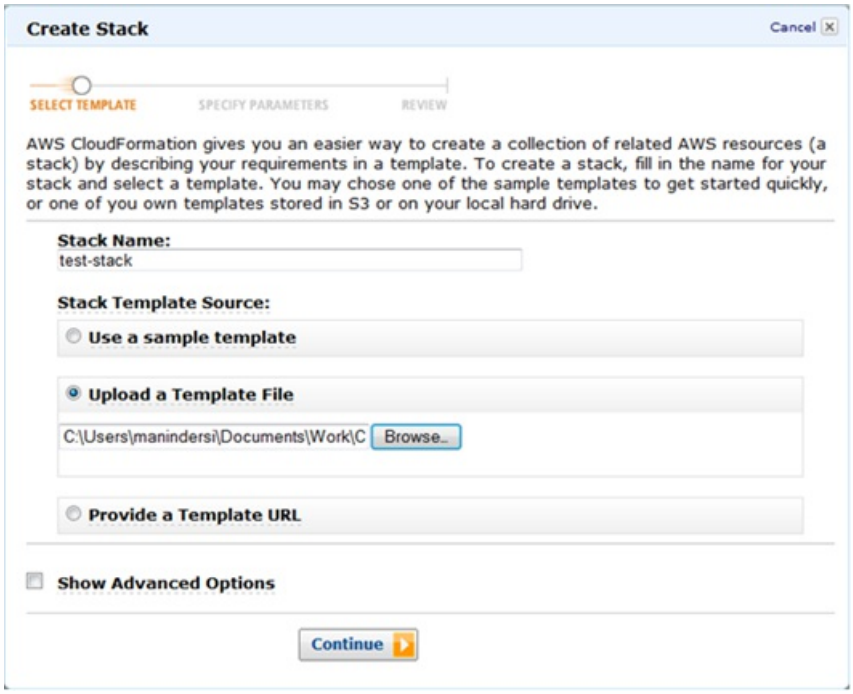
4. On the CloudFormation Stacks page, select the Region in which you plan to deploy the NetScaler VPX instance, and then click Create New Stack.



5. In the Create Stack dialog box, specify a value for Stack Name, select the Upload a Template File option, and then click Browse. Select the template for a standalone NetScaler VPX from the local drive, and then click Continue.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

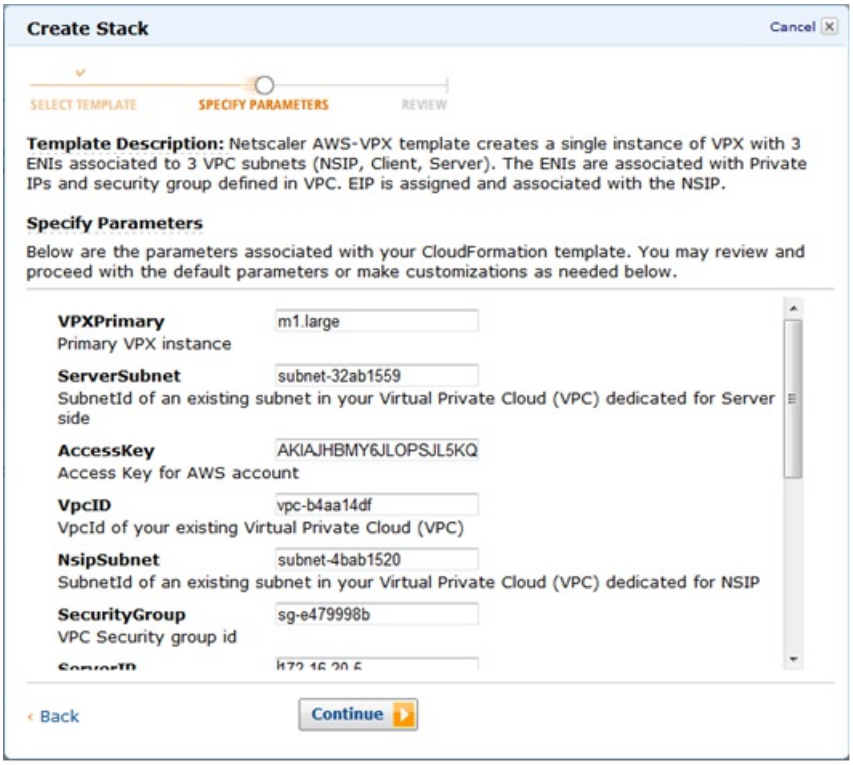
Note:



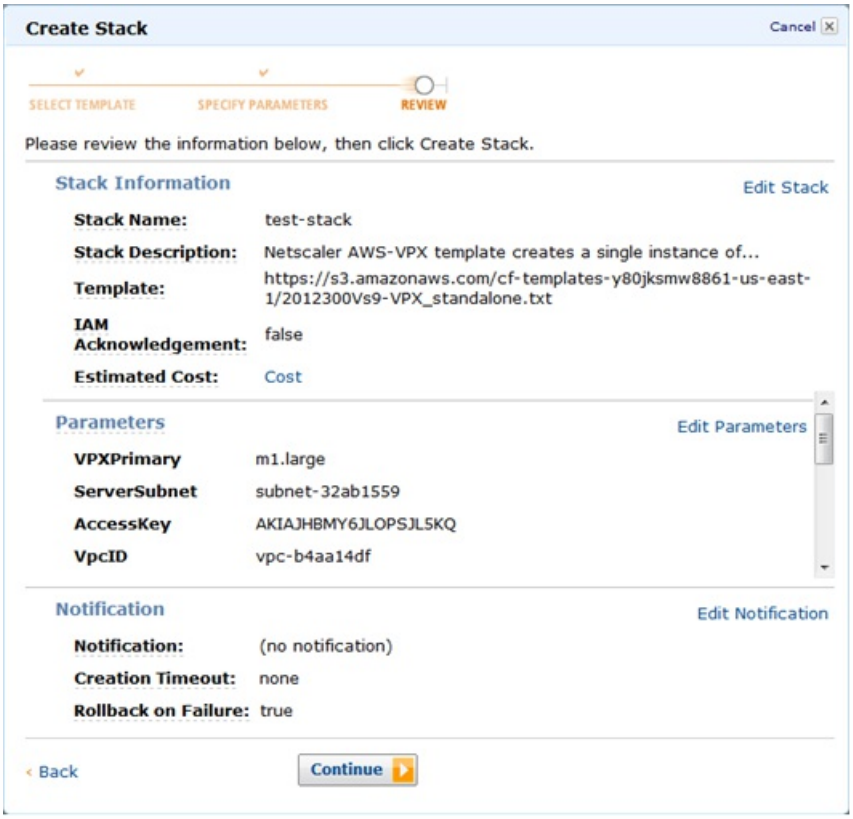
6. In the next pane, specify values for:
   - **VpcID** : An identifier to assign to the Virtual Private Cloud (VPC).
   - **NsipSubnet** : Subnet in which the NSIP is configured in the VPC
   - **ServerSubnet**: Subnet in which the server farm is configured in the VPC
   - **ClientSubnet**: SubnetId in which the client side is configured in the VPC
   - **SecurityGroup**: VPC Security group ID
   - **VPXPrimary**: Name of the primary VPX instance type
   - **AccessKey**: Access Key for IAM user account
   - **SecretKey**: Secret Key for IAM user account
   - **TenancyType**: Instance tenancy type, can be default or dedicated
   - **NsIP**: Private IP assigned to the NSIP ENI. The last octet of NSIP should be between 5 and 254.
   - **ServerIP**: Private IP assigned to the Server ENI. The last octet should be between 5 and 254.
   - **ClientIP**: Private IP assigned to the Client ENI. The last octet should be between 5 and 254.
   - **KeyName**: Name of an existing EC2 KeyPair to enable SSH access to the instances.

   Note: Make sure that the VPC, subnets, security groups, routes and gateway associations are already configured.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

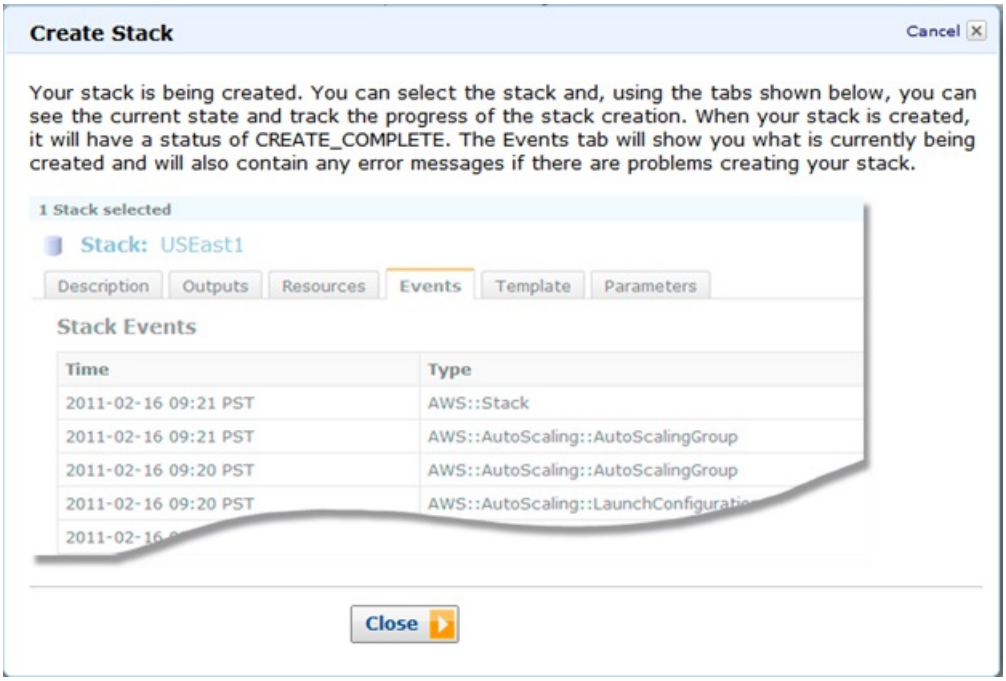


7. Click Continue.
8. Review the values in the Create Stack dialog box.

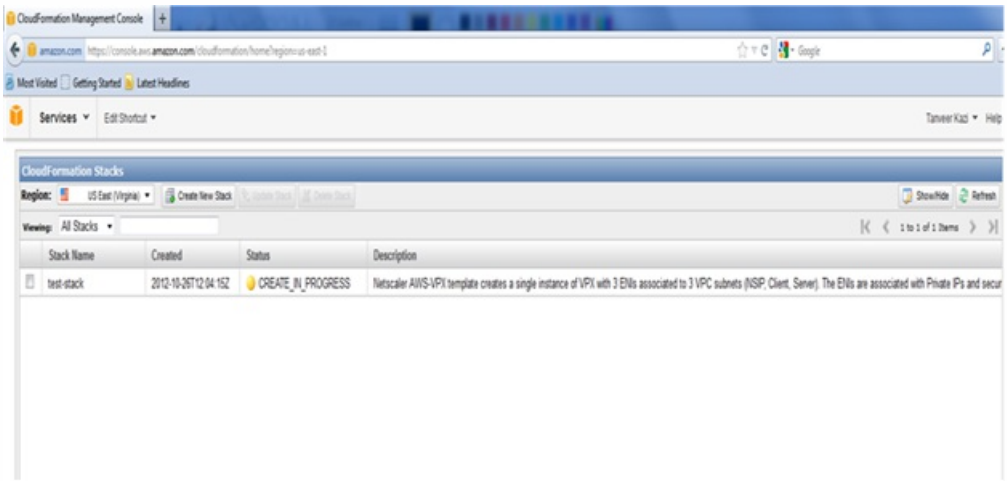


9. Click Continue to create a Stack.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



10. Click Close to close the Create Stack dialog box.
11. The new stack that you created appears on the CloudFormation Stacks page.



Note:

- Currently, the CloudFormation utility does not provide the functionality to add secondary IP addresses. Use the AWS console, after deploying a NetScaler VPX instance, to add the secondary IP addresses to the ENIs.
- The CloudFormation scripts for the standalone and HA pair VPX instances have the latest AMIs for the five supported regions. You have to update the scripts to synchronize with the latest AMIs.
- The script automatically selects the correct AMI for the region in which the VPX instance is being deployed.
- By default, all the ENIs are attached to one security group, use the AWS console to attach an ENI to a different security group.
- EIPs are automatically allocated and assigned to an instance. If the EIP limit exceeds the threshold for the region, the CloudFormation script fails and displays an error message.

Collaborating to Deliver High-Quality Products and Content Launching NetScaler VPX by using the AWS 1-Click

Updated: 2015-01-29

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1-Click helps you to launch an instance of NetScaler VPX on AWS, quickly as compared to other launching methods, with the default options. After the instance is launched on AWS, you can modify these options by using either the AWS CLI or the AWS GUI.

The default options include the following elastic network interfaces (ENIs) for the NetScaler instance:

- **Management Interface**—Associates a subnet for management related traffic. You add the NetScaler management IP (NSIP) address to this subnet.
- **Public Interface**—Associates a subnet for the client-access (user-to-NetScaler) traffic. You add one or more virtual IP (VIP) addresses on this subnet.
- **Private Interface**—Associates a subnet for server-access (NetScaler-to-server) traffic. You add subnet IP (SNIP) addresses on this subnet.

Before you begin launching an instance of NetScaler VPX on AWS, consider the following points :

- For security reasons, none of the elastic IP addresses are attached to the ENIs of the NetScaler VPX instance launched by using 1-Click. This means that the NetScaler VPX instance (including the management IP address) is not reachable from outside the AWS Virtual Private Cloud (VPC). If your VPC uses a Virtual Gateway or other method to provide a VPN access to the VPC, you can administer the instance by using the IP address of the network interface in the management subnet. If you do not have VPN access to your VPC, Citrix recommends that you set up a jump box instance within the VPC, and then use this as the source for accessing or managing other instances within the VPC. For instructions to create an SSH jump box, see https://s3.amazonaws.com/awsmp-usageinstructions/Creating_and_using_VPC.txt.
- Three default security policies are created. A policy each is attached to the management, public and private interfaces, respectively.
  - The security policy for the management interface allows traffic from a set of ports.
  - The security policies for the public and private interfaces block all the traffic to or from these interfaces. You can later modify these security groups to filter the desired traffic.
- High Availability configuration is not supported for a NetScaler VPX instance launched by using AWS 1-click.

Before you begin launching an instance of NetScaler VPX on AWS, make sure that you have the following:

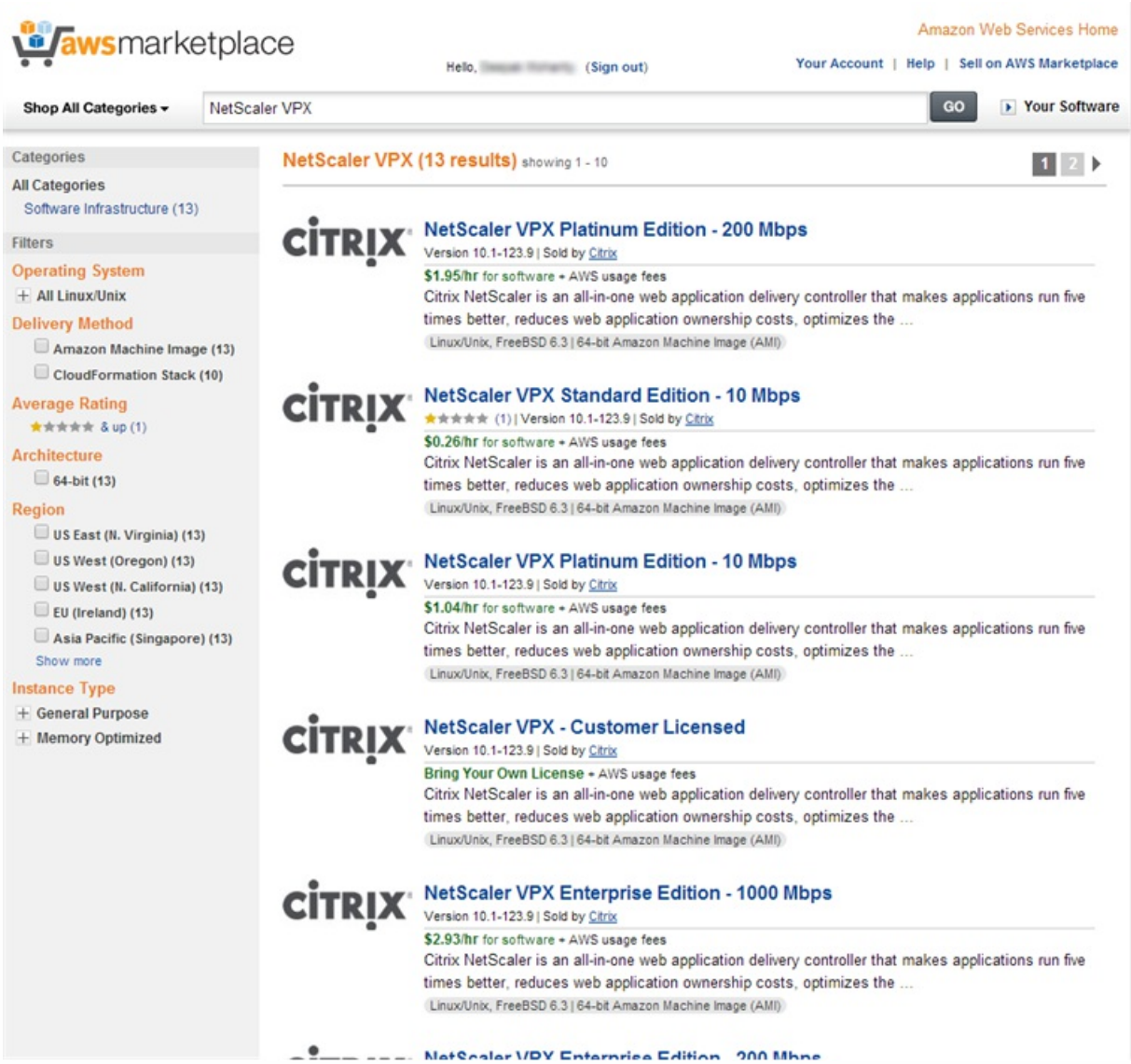
- An AWS account
- An AWS Virtual Private Cloud (VPC)
- Three subnets within the AWS VPC (one each for management interface, public interface, and private interface of the NetScaler instance)
- An IAM key pair

For information about creating an AWS account, a VPC, subnets in a VPC, and an IAM key pair, see Launching NetScaler VPX for AWS by Using the Amazon GUI and CLI toolkit.

To launch an instance of NetScaler VPX on AWS by using 1-Click

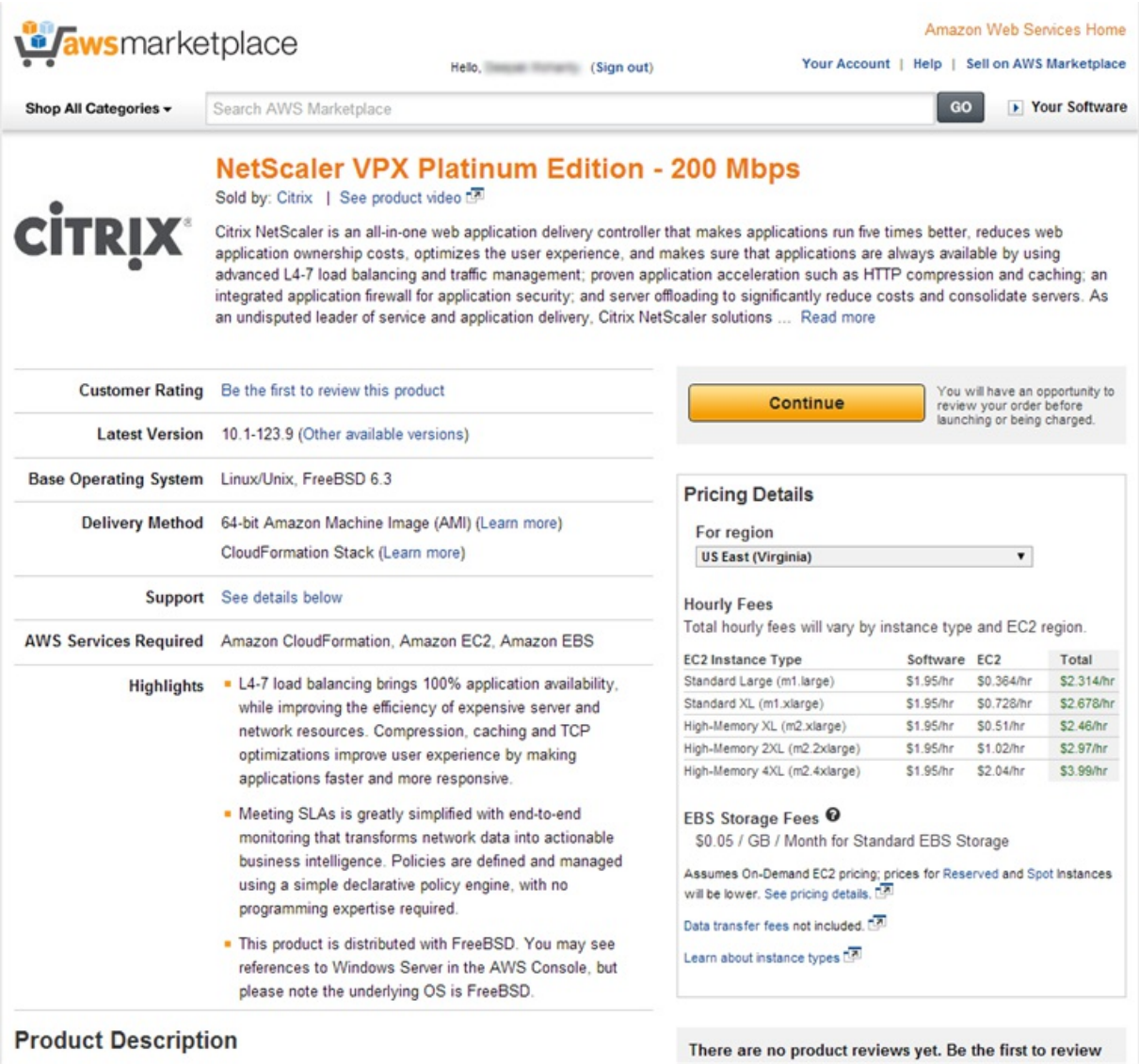
1. Log on to the AWS marketplace (https://aws.amazon.com/marketplace) by using your Amazon AWS credentials.
2. In the search field, type NetScaler VPX to search for the NetScaler AMI, and click Go.
3. On the search result page, click the desired Citrix NetScaler VPX offering.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



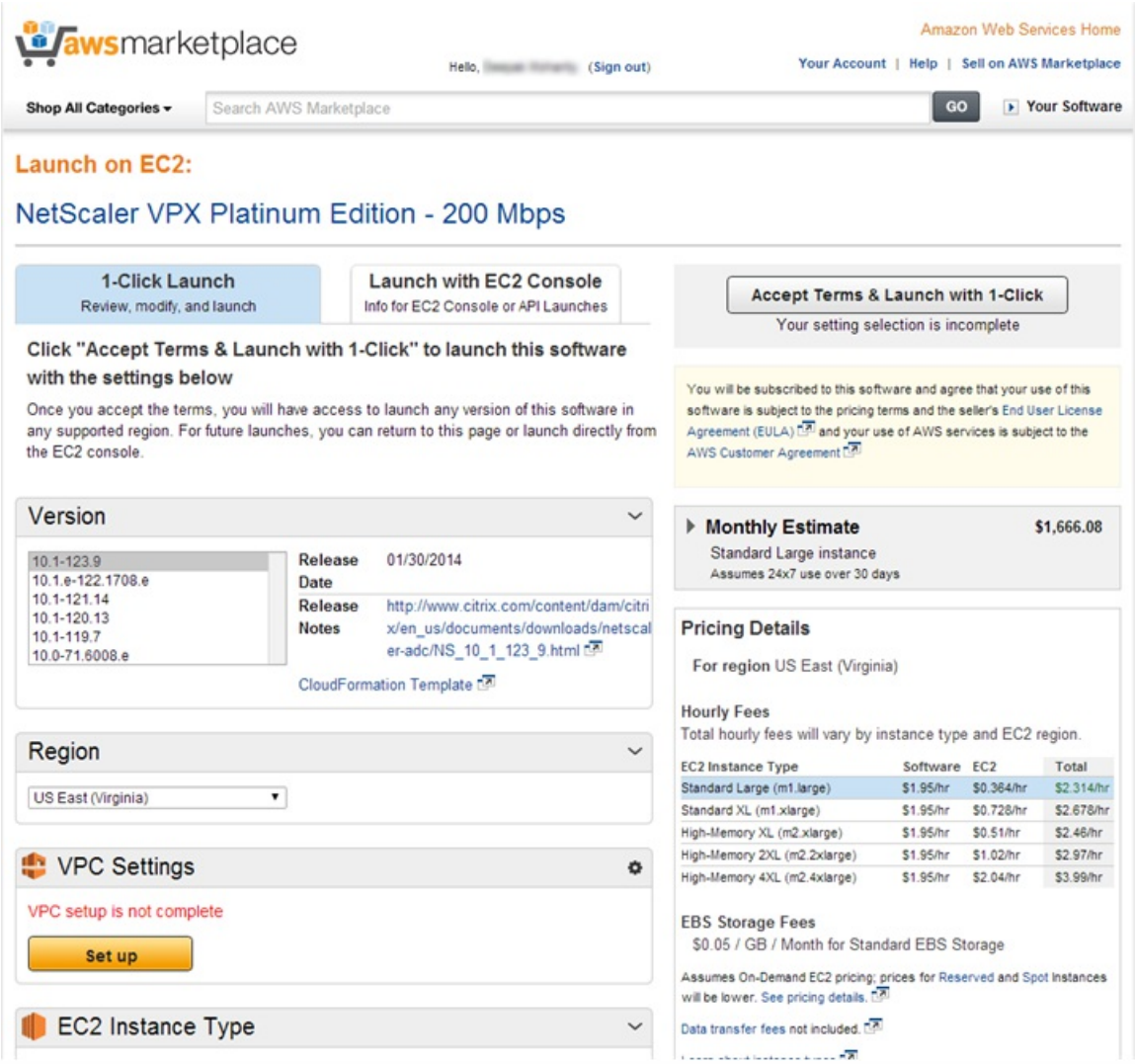
4. On the Citrix NetScaler VPX page, click Continue.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



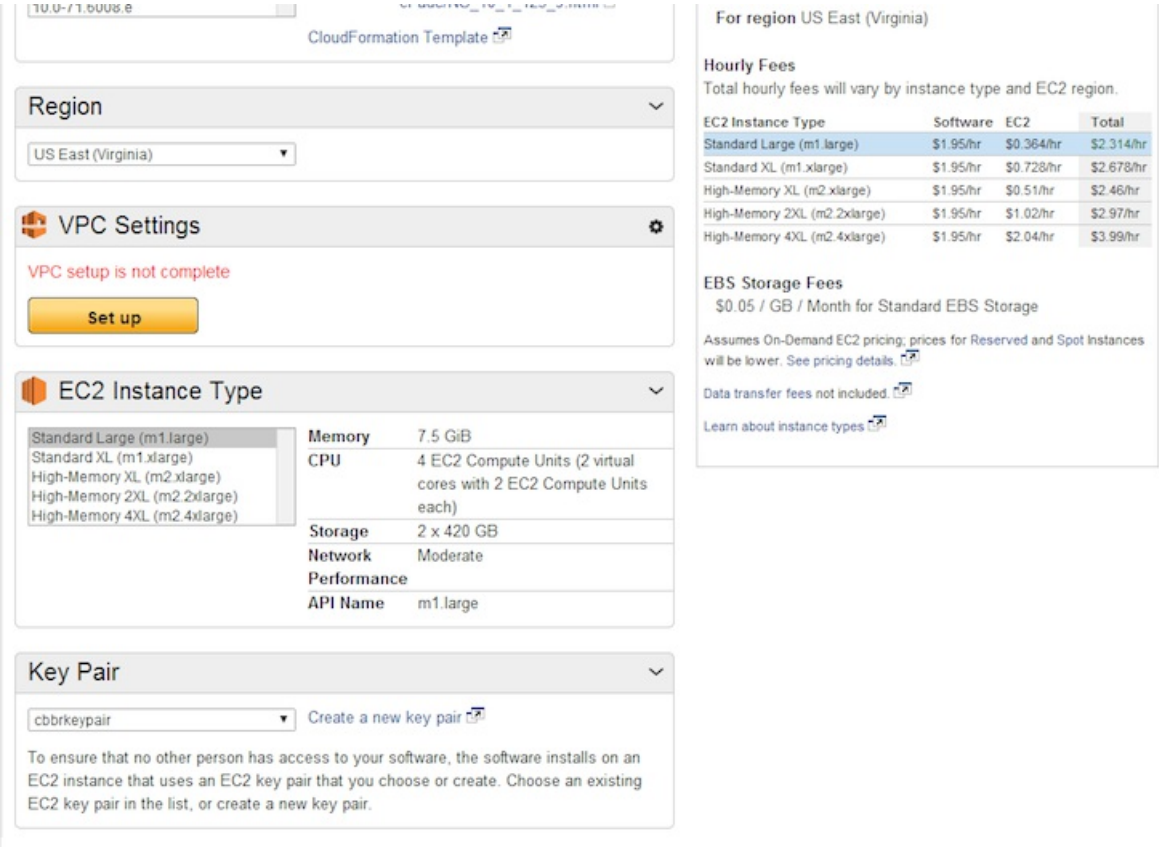
5. Click the 1-Click Launch tab. On the 1-Click Launch tab, specify values for the following fields:
   - Version
   - Region
   - EC2 Instance type
   - Key Pair

© 1999-2017 Citrix Systems, Inc. All rights reserved.



6. On the VPC Settings pane, click Setup.

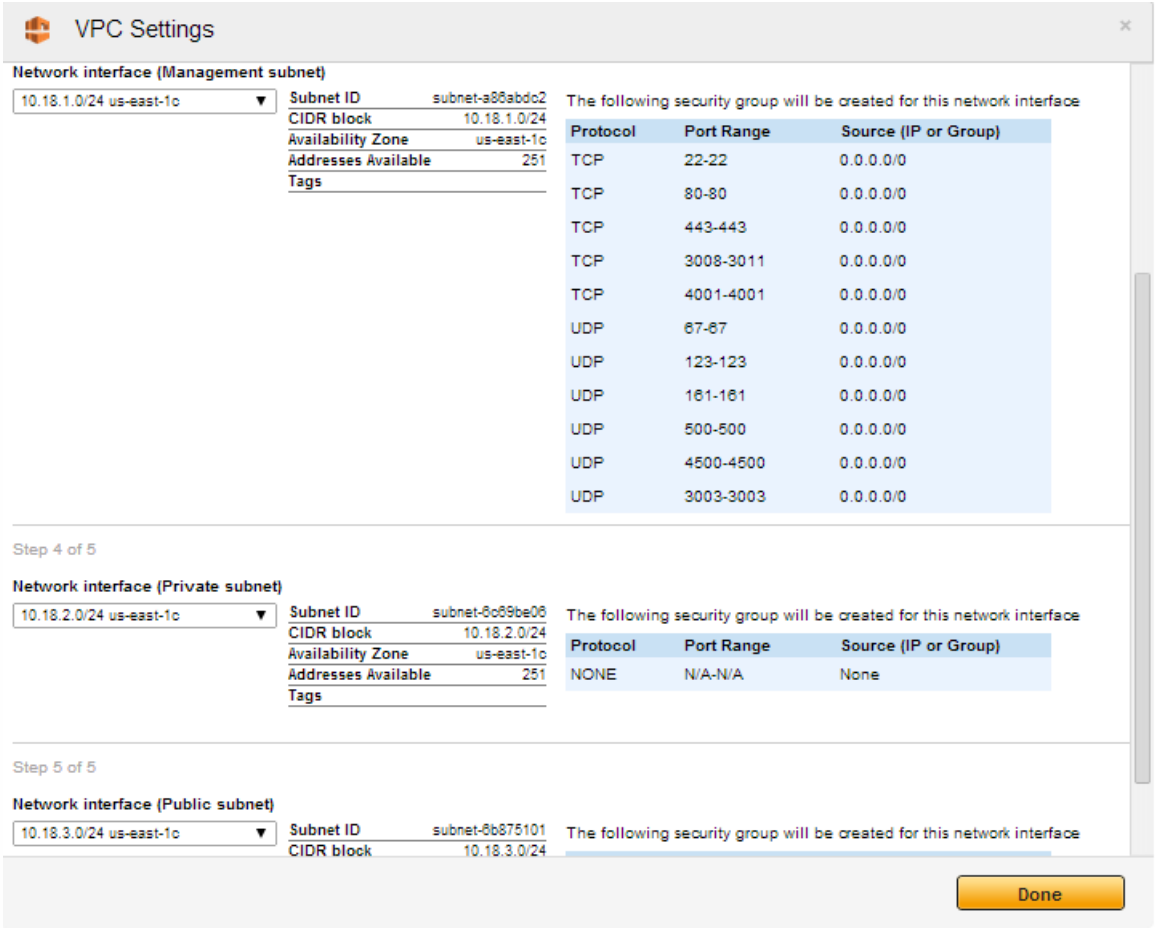
© 1999-2017 Citrix Systems, Inc. All rights reserved.



7. On the VPC Settings page, specify values for the following fields, and then click Done:
   - VPC
   - Network Interface (Management subnet)
   - Network Interface (Private subnet)
   - Network Interface (Public subnet)

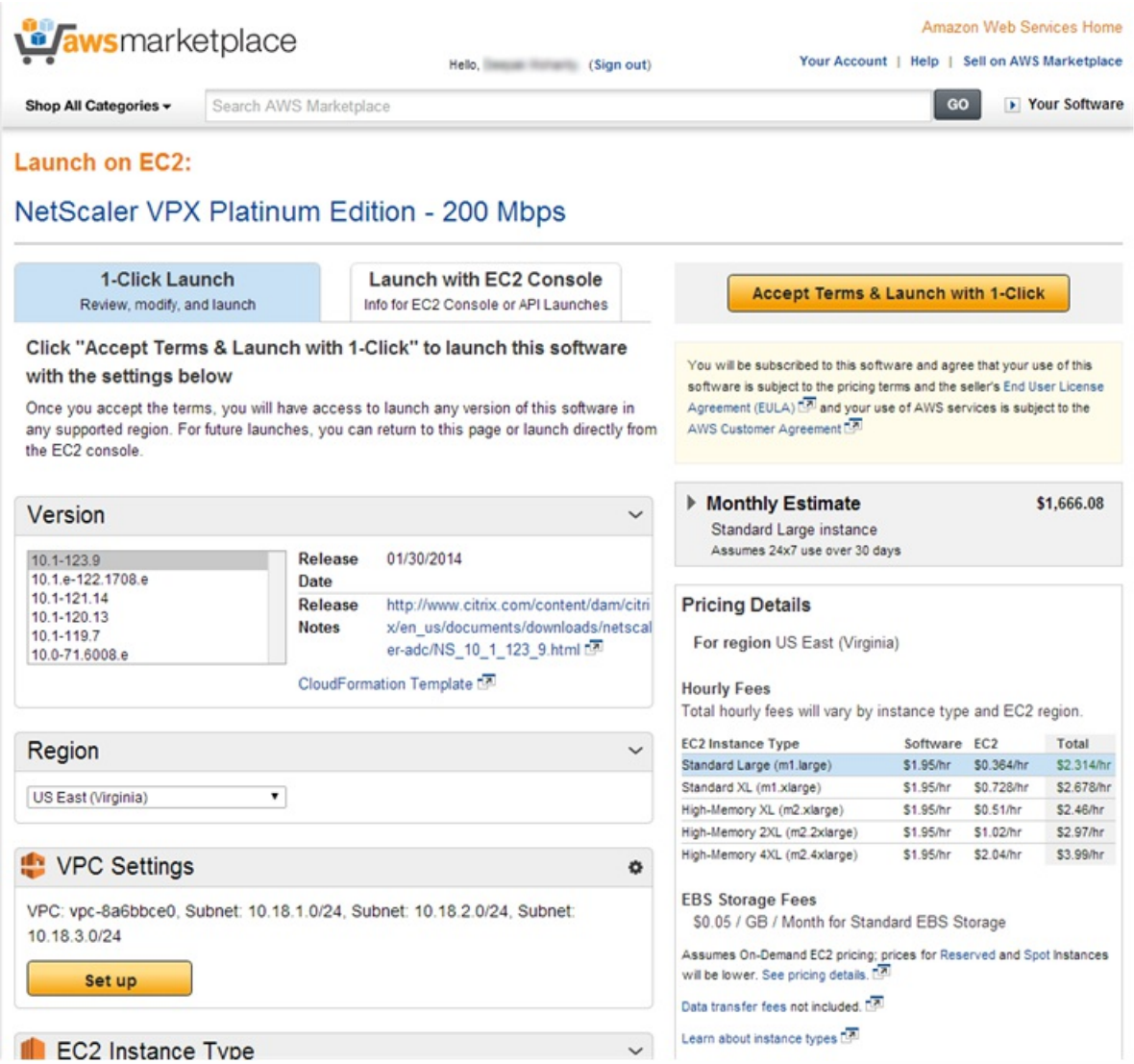
   Note: You need to make sure that the subnets attached to these ENIs are different from each other. Attaching the same subnet to more than one ENI might cause routing issues.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



8. Click Accept Terms & Launch with 1-Click.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



After few minutes, the NetScaler instance is launched with three ENIs. You can now connect to the NSIP address (the IP address on the management ENI) of the instance by using the NetScaler CLI or NetScaler GUI and start configuring the NetScaler features, for example, load balancing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Verifying the NetScaler VPX on AWS Installation

May 14, 2014

When the NetScaler instance is running, you can access the instance through the NetScaler GUI or the NetScaler CLI by connecting to the EIP associated with the management ENI (NSIP). For example, use the following addressing notation in a web browser:

http://<Elastic_IP> (unsecured access)

or

https://<Elastic_IP> (secured access)

Note:

- To access a NetScaler instance through SSH, provide the .pem file.
- You can use the AWS GUI console to manually add the private IP addresses for SNIPs on server subnets and VIPs on client subnets.
- If you want to access the NSIP from the Internet, you must assign an EIP to the NSIP address of each NetScaler instance. Also, make sure that the NSIP subnet is associated with a routing table that has a default route set to the Internet gateway.
- If you want VIP addresses to be accessible through the Internet, you must associate an EIP with each VIP address that is defined in the configuration.
- The following are the default administrator credentials to access a NetScaler VPX instance:
  - Username—nsroot
  - Password—The default password for the nsroot account is set to the AWS instance-ID of the NetScaler VPX instance. For a high availability configuration between two NetScaler VPX instances, the nsroot password of the secondary node is set to that of the primary node after the HA configuration synchronization.

- You can find the private key file from the AWS console. To view the private key file:
  1. Log on to the AWS marketplace (https://aws.amazon.com/marketplace) by using your Amazon AWS credentials.
  2. Click **Amazon Web Services Home**.
  3. Click **My Account/Console**, and then click **Security Credentials**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Attaching Additional IP Addresses to an Instance

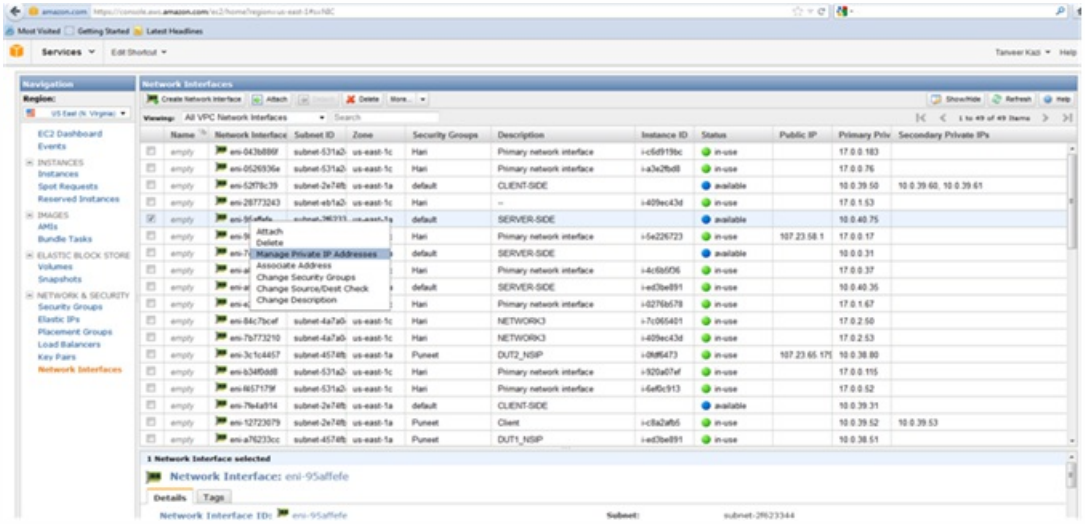
Aug 23, 2013

You can attach additional IP addresses to an instance as follows:

1. Add a secondary IP address to an ENI.
2. Associate an EIP with the secondary IP address that you created.

To add a secondary IP address to the ENI

1. In a web browser, open the website at www.aws.amazon.com and log on with AWS credentials.
2. Click My Account/Console, and then click AWS Management Console.
3. On the Amazon Web Services page, click EC2.
4. On the Amazon EC2 Console's Dashboard page, in the Navigation pane, in NETWORK & SECURITY, click Network Interfaces.
5. In the Network Interfaces pane, right-click the ENI attached to the subnet, and then select the Manage Private IP Addresses option from the pop-up menu.



6. In the Manage Private IP Addresses dialog box, click Assign a secondary private IP address and either let AWS automatically assign an IP address or type an IP address in the auto-assign text-field. Click Yes, Update.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Associating an EIP with the secondary IP

Complete the following steps to associate an EIP with a secondary IP address:

1. On the Amazon EC2 Console Dashboard page, in the Navigation pane, in NETWORK & SECURITY, click Elastic IPs.
2. In the Addresses pane, click Allocate New Address.
3. In the Allocate New Address dialog box, select VPC from the EIP used in drop-down list and click Yes, Allocate.
4. Select the newly allocated EIP, and click Associate Address.
5. In the Associate Address dialog box, select, from the **Instance** and the **Private IP address** drop-down lists, the instance and private address that you want to associate with the EIP. Then, click Yes, Associate.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Downloading a NetScaler VPX License

Jun 24, 2014

After the initial instance launch, NetScaler VPX for AWS requires a license. If you are bringing your own license (BYOL), see the *VPX Licensing Guide* at http://support.citrix.com/article/CTX122426

You have to:

1. Use the licensing portal within MyCitrix to generate a valid license.
2. Upload the license to the instance.

If this is a **paid** marketplace instance, then you do not need to install a license. The correct feature set and performance will activate automatically.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing Servers in different Availability Zones

Nov 27, 2012

A NetScaler instance can be used to load balance servers running in the same availability zone, or in:

- A different availability zone (AZ) in the same AWS VPC
- A different AWS region
- AWS EC2 in a VPC

To enable NetScaler to load balance servers running outside the AWS VPC that the NetScaler instance is in, configure the NetScaler to use EIPs to route traffic through the Internet gateway, as follows:

1. Configure a SNIP on the NetScaler by using the NetScaler CLI or the NetScaler GUI
2. Enable traffic to be routed out of the AZ, by creating a public facing subnet for the server-side traffic.
3. Add an Internet gateway route to the routing table, using the AWS GUI console.
4. Associate the routing table you just updated with the server-side subnet.
5. Associate an EIP with the server-side private IP address that is mapped to a NetScaler SNIP address.

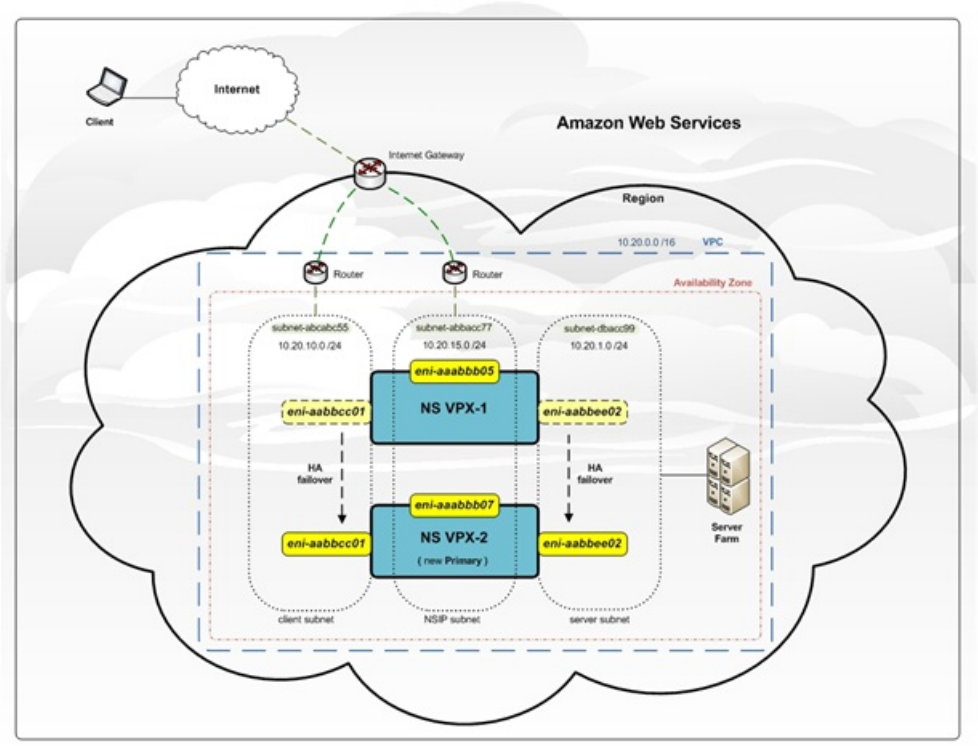
© 1999-2017 Citrix Systems, Inc. All rights reserved.

High Availability

May 13, 2014

Two Citrix NetScaler VPX instances in AWS can be configured as a high availability (HA) pair. With one instance configured as the primary node and the other as the secondary node, the primary node accepts connections and manages servers while the secondary node monitors the primary. If, for any reason, the primary node is unable to accept connections, the secondary node takes over.

The following figure shows an example of the HA deployment architecture for NetScaler VPX on AWS.
Figure 1. NetScaler VPX on AWS - HA Deployment



To deploy HA for VPX on AWS, you must configure at least two ENIs on the primary instance and a single ENI on the secondary instance. On each instance, configure the NetScaler IP (NSIP) address (the management address) on the default ENI. On the primary instance, use the additional ENIs for client and server connections.

For instructions on obtaining access and secret keys, in the AWS documentation, see "How Do I Get Security Credentials?" and "Creating, Modifying, and Viewing User Access Keys (AWS Management Console)." For instructions to create an IAM user and set permissions, see "Creating an IAM Account."

Example format for a key file is:
ACCESS_KEY="AKIAJPBBBBBBBVA2PR2OHJNA"
SECRET_KEY="d75KxU7ukd44444NNNNtrrAOgynwBdJoSiooP"
Note: For HA failover to work:

1. The NSIP addresses for each NetScaler instance in an HA pair must be configured on the default ENI of the instance.
2. Both the primary and secondary instances must have EIPs associated with the NSIP or NAT configured to handle outgoing traffic in order to have access to the AWS API servers.
3. Client and server traffic (data-plane traffic) must not be configured on the default ENI.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Access and secret keys associated with the user's AWS Identity and Access Management (IAM) account. If the correct key information is not used when creating VPX instances, the HA deployment will fail. The access and secret keys are required for sending Query APIs to the AWS server.
5. Nameservers/DNS servers are configured at VPC level using DHCP options.

Notes on HA:

- Because Amazon does not allow any broadcast/multicast packets in AWS, HA is implemented by migrating data-plane ENIs from the primary to the secondary (new primary) VPX instance when the primary VPX instance fails.
- To deploy HA for VPX on AWS, you must configure at least two ENIs on the primary instance and a single ENI on the secondary instance.
- Because the default ENI cannot be moved to another VPX instance, you should not use the default ENI for data.
- The message AWSCONFIG IOCTL NSAPI_HOTPLUG_INTF success output 0 indicates that the two data ENI's have successfully attached to the secondary instance (the new primary).
- Failover might take up to 20 seconds due to the AWS detach/attach ENI mechanism.
- Upon failover, the failed instance always restarts.
- The secondary node always has one ENI interface (for management) and the primary node can have up to four ENIs.
- The heartbeat packets are received only on the management interface.
- The configuration file of the primary and secondary NetScaler appliances is synchronized, including the nsroot password. The nsroot password of the secondary node is set to that of the primary node after the HA configuration synchronization.
- **The AWS debug messages are available in the log file, /var/log/ns.log, on the VPX instance**.

Configuring High Availability for VPX on AWS

To deploy HA for two VPX instances on AWS, you must create the primary NetScaler VPX instances with three ENIs and the secondary NetScaler VPX with a single ENI.

Following is an example of launching a primary VPX instance with three ENIs:

C:\aws-vpc-config>ec2-run-instances ami-bd2986d4 -n 1 -t m1.large -k keyPairName -f ./access-secret-key-file -a :0:subnet-15fa057e:"NSIP":10.20.15.21 -a :1:subnet-1547ba7e:"CLIENT-SIDE":10.20.10.21::::"10.20.10.22,10.20.10.23,10.20.10.24,10.20.10.25,10.20.10.26,10.20.10.27,10.20.10.28,10.20.10.29,10.20.10.30" -a :2:subnet-cc47baa7:"SERVER-SIDE":10.20.1.21::::"10.20.1.22,10.20.1.23,10.20.1.24,10.20.1.25,10.20.1.26,10.20.1.27,10.20.1.28,10.20.1.29,10.20.1.30"
Following is an example of launching a secondary VPX instance with a single ENI:

C:\aws-vpc-config>ec2-run-instances ami-bd2986d4 -n 1 -t m1.large -k keyPairName -f access-secret-key-file -a :0:subnet-15fa057e:"NSIP":10.20.15.31
Note: The access-secret-key-file argument contains the access and secret key. (You cannot change the access-secret-key-file associated with a VPC instance after it is created.)
After the two NetScaler instances are UP, configure the HA pairing on both the instances. You have to configure the instance with two or more ENIs before configuring HA on the instance with one ENI. Use the add HA node command, from within the NetScaler CLI, or from the NetScaler GUI. For example:

On the VPX instance with two or more ENIs:

add HA node 1 10.20.15.31

On the VPX instance with one ENI:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

add HA node 1 10.20.15.21

After you enter add HA node commands, the two nodes form an HA pair, and configuration information is synchronized between the two VPX instances.

To remove HA from NetScaler VPX pair

You can remove HA configuration from the NetScaler VPX pair by using the remove ha node command. You have to remove the HA configuration from the secondary NetScaler VPX before removing the HA configuration from the primary NetScaler VPX.
For example, on the Secondary NetScaler VPX instance, at the NetScaler command line, type:

remove ha node

save config

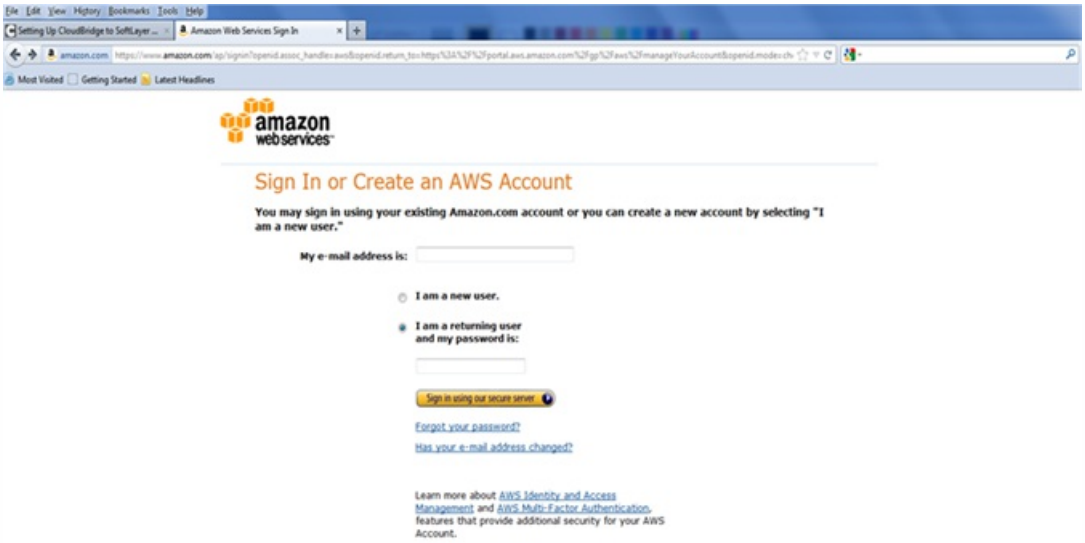
On the Primary NetScaler VPX instance, at the NetScaler command line, type:

remove ha node

save config

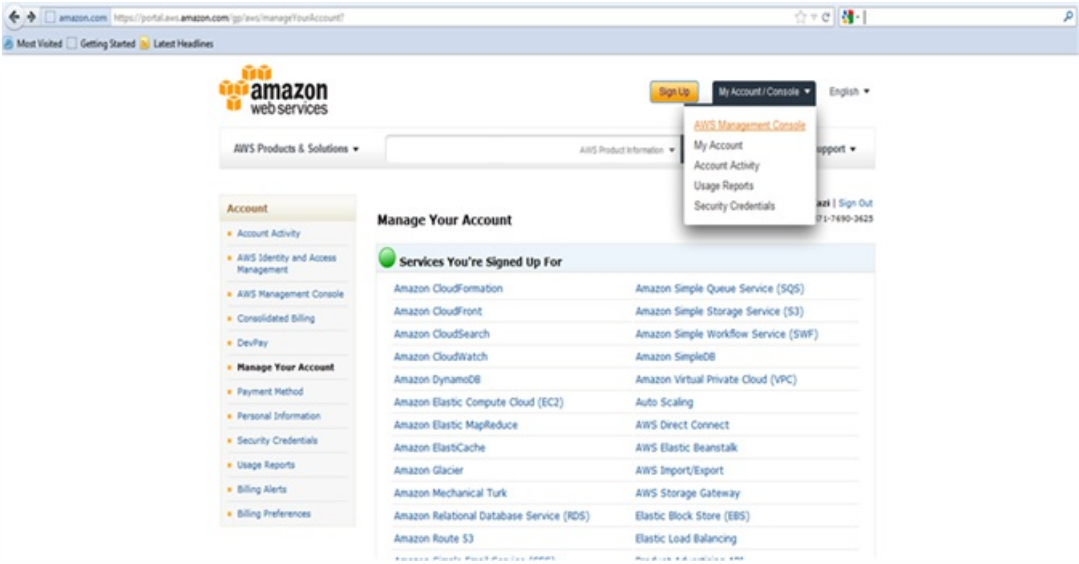
Launching NetScaler VPX pairs for HA by using Citrix CloudFormation

1. In a web browser, open the website at www.aws.amazon.com and log on with AWS credentials.

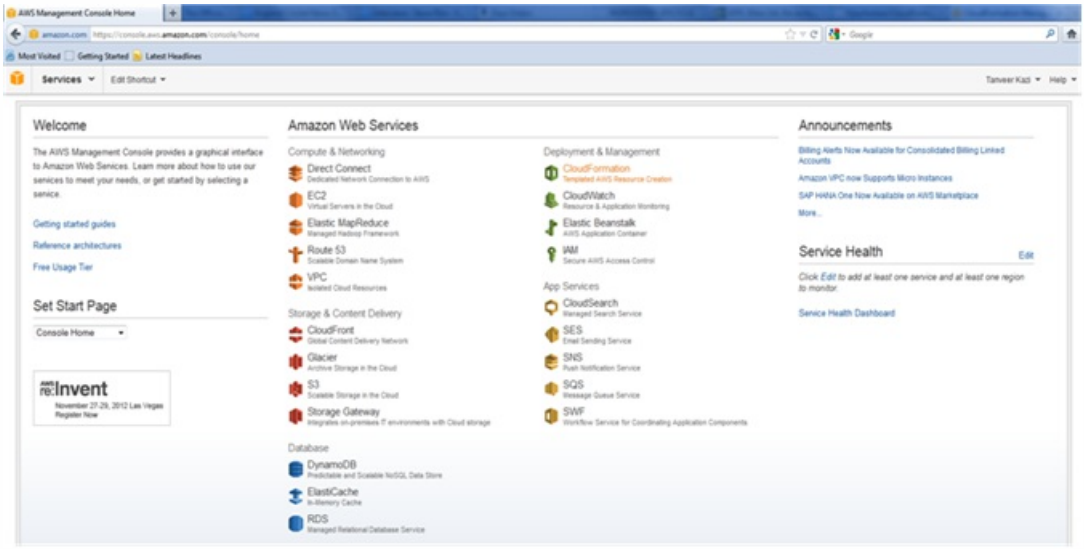


2. Click My Account/Console, and then click AWS Management Console.

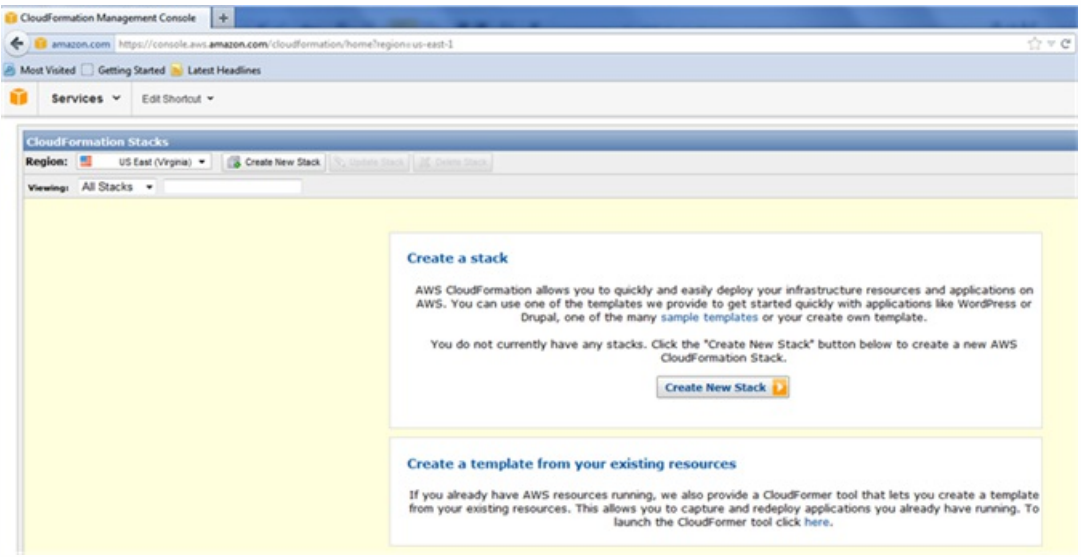
© 1999-2017 Citrix Systems, Inc. All rights reserved.



3. On the Amazon Web Services page, in the Deployment & Management section, click Cloud Formation.

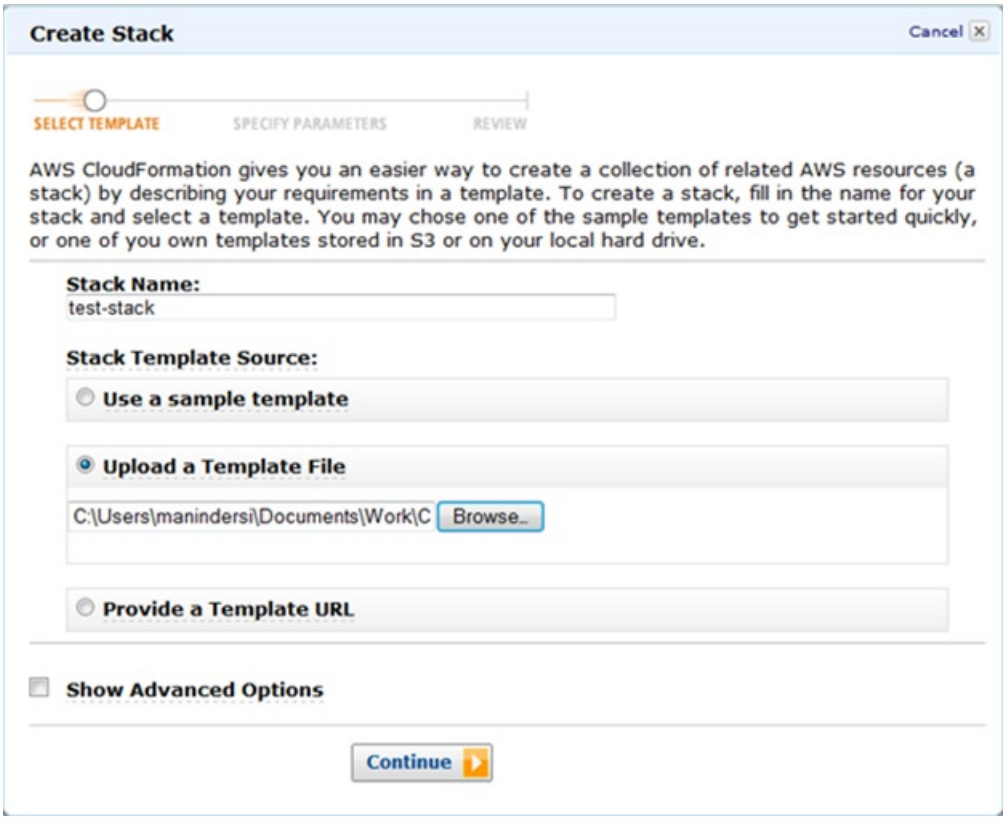


4. On the CloudFormation Stacks page, select the Region in which you plan to deploy the NetScaler VPX instance, and then click Create New Stack.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

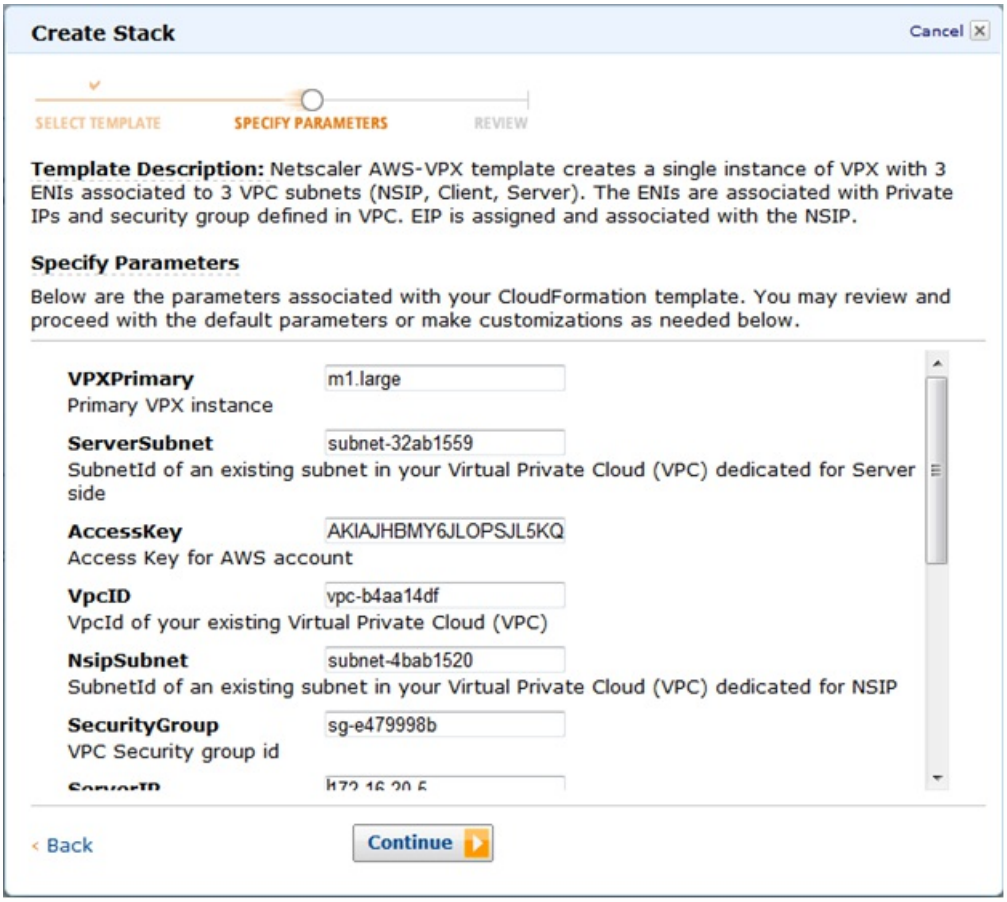
5. In the Create Stack dialog box, specify value for Stack Name, select the Upload a Template File option, and then click Browse. Select the template for HA NetScaler VPX from the local drive, and then click **Continue**.



6. In the next pane, specify values for:
   - **VpcID**: An identifier to assign to the Virtual Private Cloud (VPC).
   - **NsipSubnet**: Subnet in which the NSIP is configured in VPC.
   - **ServerSubnet**: Subnet in which the server farm is configured in VPC.
   - **ClientSubnet**: SubnetId in which the client side is configured in VPC.
   - **SecurityGroup**: VPC Security group id.
   - **VPXPrimary**: Name of Primary VPX instance type.
   - **AccessKey**: Access Key for IAM user account.
   - **SecretKey**: Secret Key for IAM user account.
   - **TenancyType**: Instance tenancy type, can be default or dedicated.
   - **NsIP**: Private IP assigned to the NSIP ENI. The last octet of NSIP should be between 5 and 254.
   - **NsIPSec**: Private IP assigned to the NSIP ENI of Secondary. last octet has to be between 5 and 254.
   - **ServerIP**: Private IP assigned to the Server ENI. The last octet should be between 5 and 254.
   - **ClientIP**: Private IP assigned to the Client ENI. The last octet should be between 5 and 254.
   - **KeyName**: Name of an existing EC2 KeyPair to enable SSH access to the instances.

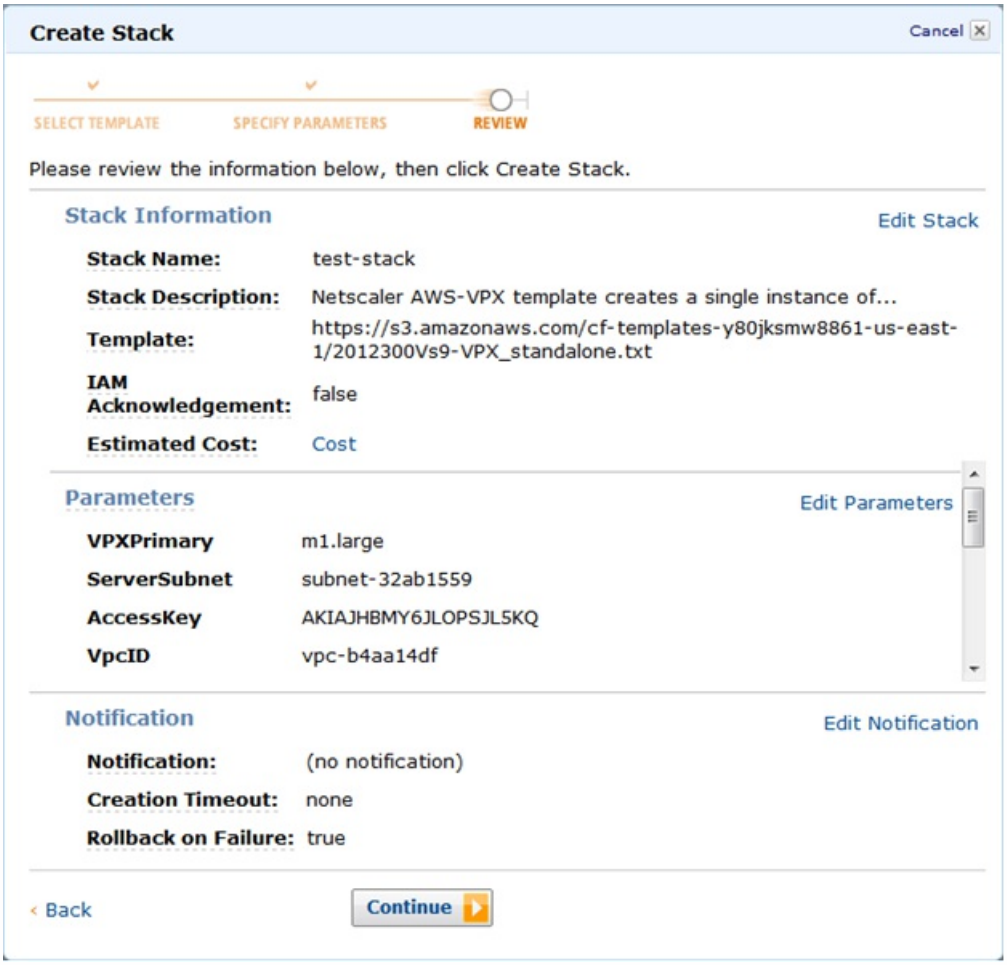
   Note: Make sure that the VPC, subnets, security groups, routes associations, gateway associations are already configured.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

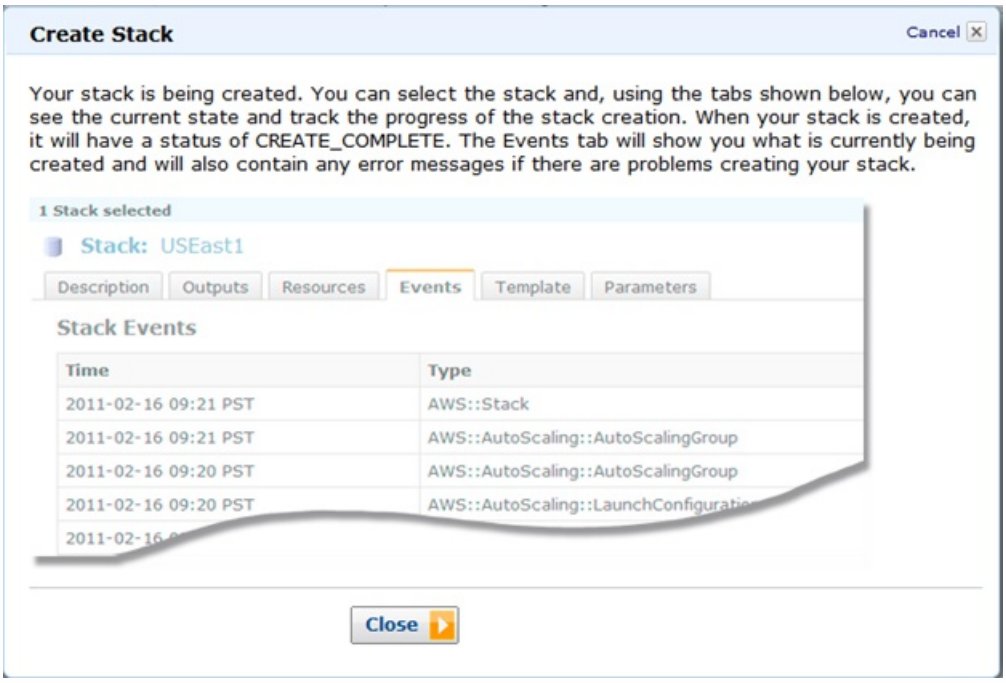


7. Click Continue.
8. Review the specified values in the Create Stack dialog box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

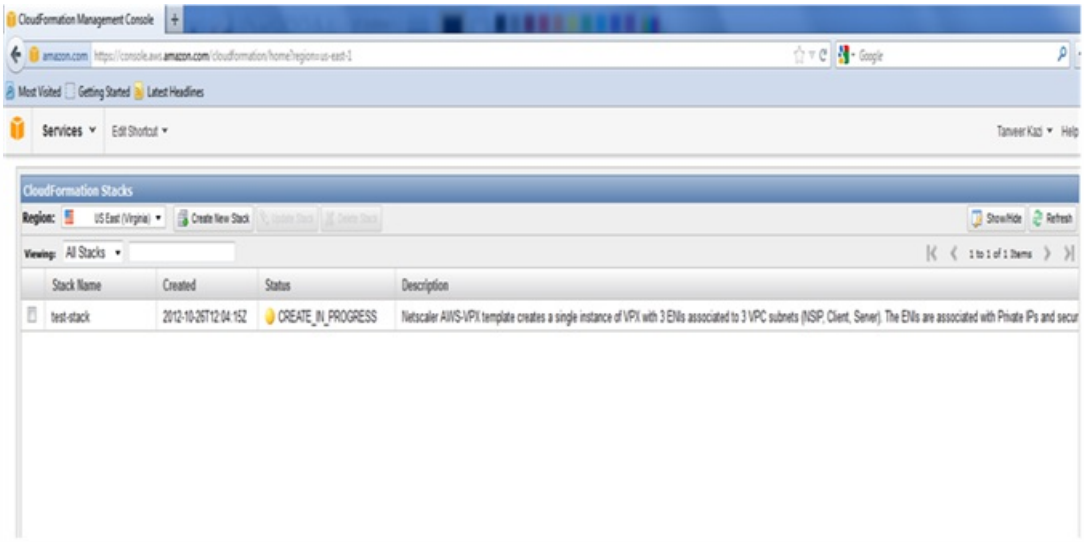


9. Click Continue to create a Stack.



10. Click Close to close the Create Stack dialog box.
11. The new stack that you created appears on the CloudFormation Stacks page.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Upgrading a NetScaler VPX instance on AWS

Apr 22, 2014

You can upgrade the EC2 instance type, throughput, software edition, and the system software of a NetScaler VPX running on AWS. For certain types of upgrades, Citrix recommends using the High Availability Configuration method to minimize downtime.

Note:

- NetScaler software release 10.1.e-124.1308.e or later for a NetScaler VPX AMI (including both utility license and customer license) does not support the M1 and M2 instance families.
- Because of changes in NetScaler instance support, downgrading from 10.1.e-124 or a later release to 10.1.123.x or an earlier release is not supported.
- Most of the upgrades do not require the launch of a new AMI, and the upgrade can be done on the current NetScaler AMI instance. If you do want to upgrade to a new NetScaler AMI instance, use the high availability configuration method.

Changing the EC2 Instance Type of a NetScaler VPX Instance on AWS

Updated: 2014-04-22

If your NetScaler VPX instances are running release 10.1.e-124.1308.e or later, you can change the EC2 instance type from the AWS console as follows:

1. Stop the VPX instance.
2. Change the EC2 instance type from the AWS console.
3. Start the instance.

You can also use the above procedure to change the EC2 instance type for a release, earlier than 10.1.e-124.1308.e, unless you want to change the instance type to M3. In that case, you must first follow the standard NetScaler upgrade procedure, at , to upgrade the NetScaler software to 10.1.e-124 or a later release, and then follow the above steps.

Upgrading the Throughput or Software Edition for a NetScaler VPX Instance on AWS

Updated: 2014-04-22

To upgrade the software edition (for example, to upgrade from standard to platinum edition) or throughput (for example, to upgrade from 200 mbps to 1000mbps), the method depends on the instance's license.

Using a customer license (Bring-Your-Own-License)

If you are using a customer license, you can purchase and download the new license from the Citrix Licensing portal (MyCitrix), and then install the license on the VPX instance. For more information about downloading and installing a license from the MyCitrix portal, see the VPX Licensing Guide.

Using a utility license (Utility license with hourly fee)

AWS does not support direct upgrades for fee-based instances. To upgrade the software edition or throughput of a fee based NetScaler VPX instance, launch a new AMI with the desired license and capacity and migrate the older instance configuration to the new instance. This can be achieved by using a NetScaler high availability configuration as described in "Upgrading to a New NetScaler AMI Instance by Using a NetScaler High Availability Configuration."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Upgrading the System Software of a NetScaler VPX Instance on AWS

Updated: 2014-04-22

If you need to upgrade a NetScaler instance running 10.1.e-124.1308.e or a later release, follow the standard NetScaler upgrade procedure at .

If you need to upgrade a NetScaler instance running a release older than 10.1.e-124.1308.e to 10.1.e-124.1308.e or a later release, first upgrade the system software, and then change the instance type to M3 as follows:

1. Stop the VPX instance.
2. Change the EC2 instance type from the AWS console.
3. Start the instance.

Upgrading to a New NetScaler AMI Instance by Using a NetScaler High Availability Configuration

Updated: 2014-04-22

To use the high availability method of upgrading to a new NetScaler AMI instance, perform the following tasks:

- Create a new instance with the desired EC2 instance type, software edition, throughput, or software release from the AWS marketplace.
- Configure high availability between the old instance (to be upgraded) and the new instance. After high availability is configured between the old and the new instance, configuration from the old instance is synchronized to the new instance.
- Force an HA failover from the old instance to the new instance. As a result, the new instance becomes primary and starts receiving traffic.
- Stop, and reconfigure or remove the old instance from AWS.

Prerequisites and Points to Consider

- Make sure you understand how high availability works between two NetScaler VPX instances on AWS. For more information about high availability configuration between two NetScaler VPX instances on AWS, see High Availability.
- You must create the new instance in the same availability zone as the old instance, having the exact same security group and subnet.
- High availability setup requires access and secret keys associated with the user's AWS Identity and Access Management (IAM) account for both instances. If the correct key information is not used when creating VPX instances, the HA setup fails. For more information about creating an IAM account for a VPX instance, see [Creating an IAM Account].
- You must use the EC2 console to create the new instance. You cannot use the AWS 1-click launch, because it does not accept the access and secret keys as the input.
- The new instance should have only one ENI interface.

To upgrade a NetScaler VPX Instance by using a high availability configuration

1. Configure high availability between the old and the new instance. To configure high availability between two NetScaler VPX instances, at the NetScaler command prompt of each intance, type:
   - add ha node <nodeID> <IPaddress of the node to be added>
   - save config

   Example

   At the NetScaler command prompt of the old instance, type:

   > add ha node 30 192.0.2.30

   Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the NetScaler command prompt of the new instance, type:

```
> add ha node 10 192.0.2.10
Done
```

Note the following:

- In the HA setup, the old instance is the primary node and the new instance is the secondary node.
- The NSIP IP address is not copied from the old instance to the new instance. Therefore, after the upgrade, your new instance has a different management IP address from the previous one.
- The nsroot account password of the new instance is set to that of the old instance after HA synchronization.

For more information about high availability configuration between two NetScaler VPX instances on AWS, see High Availability.

2. Force an HA failover. To force a failover in a high availability configuration, at the NetScaler command prompt of either of the instances, type:
   - force HA failover

   As the result of forcing a failover, the ENIs of the old instance are migrated to the new instance and traffic flows through the new instance (the new primary node). The old instance (the new secondary node) restarts.

   If the following warning message appears, type N to abort the operation:

   ```
   WARNING]:Force Failover may cause configuration loss, peer health not optimum. Reason(s):
   HA version mismatch
   HA heartbeats not seen on some interfaces
   Please confirm whether you want force-failover (Y/N)?
   ```

   The warning message appears because the system software of the two VPX instances is not HA compatible. As a result, the configuration of the old instance cannot be automatically synced to the new instance during a forced failover.

   Following is the workaround for this issue:

   1. At the NetScaler shell prompt of the old instance, type the following command to create a backup of the configuration file (ns.conf):
      - copy /nsconfig/ns.conf to /nsconfig/ns.conf.bkp
   2. Remove the following line from the backup configuration file (ns.conf.bkp):
      - set ns config -IPAddress <IP> -netmask <MASK>
        For example, set ns config -IPAddress 192.0.2.10 -netmask 255.255.255.0
   3. Copy the old instance's backup configuration file (ns.conf.bkp) to the /nsconfig directory of the the new instance.
   4. At the NetScaler shell prompt of the new instance, type the following command to load the old instance's configuration file (ns.conf.bkp) on the new instance:
      - batch -f /nsconfig/ns.conf.bkp
   5. Save the configuration on the new instance.
      - Save conifg
   6. At the NetScaler command prompt of either of the nodes, type the following command to force a failover, and then type Y for the warning message to confirm the force failover operation:
      - force ha failover

      Example

      ```
      > force ha failover

      WARNING]:Force Failover may cause configuration loss, peer health not optimum.
      Reason(s):
      ```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

HA version mismatch
HA heartbeats not seen on some interfaces
Please confirm whether you want force-failover (Y/N)? Y

3. Remove the HA configuration, so that the two instances are no longer in an HA configuration. First remove the HA configuration from the secondary node and then remove the HA configuration from the primary node.
   To remove an HA configuration between two NetScaler VPX instances, at the command prompt of each instance, type:

   - remove ha node <nodeID>
   - save config

   For more information about high availability configuration between two NetScaler instances on AWS, see High Availability.

   Example

   At the NetScaler command prompt of the old instance (new secondary node), type:
   > remove ha node 30
   Done
   > save config
   Done
   At the NetScaler command prompt of the new instance (new primary node), type:
   > remove ha node 10
   Done
   > save config
   Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

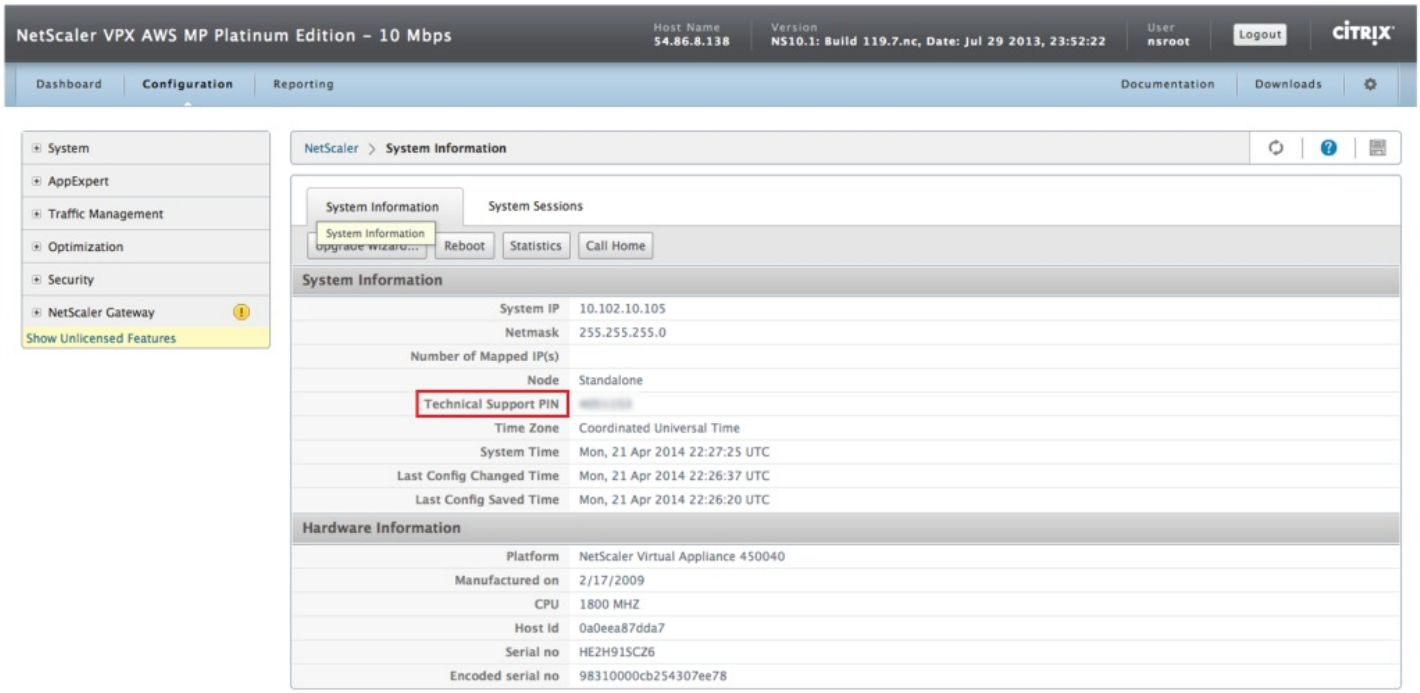
Troubleshooting the NetScaler VPX on AWS

Apr 28, 2014

Amazon does not provide console access to a NetScaler VPX virtual instance. To troubleshoot, you have to use the AWS GUI to view the activity log. You can debug only if the network is connected. To view an instance's system log, right-click the instance and select system log.

Citrix provides support for fee based NetScaler VPX instances (utility license with hourly fee) on AWS. To file a support case, find your AWS account number and support PIN code, and call Citrix support. You will also be asked for your name and email address. To find the support PIN, log on to the NetScaler configuration utility and navigate to the System page.

Here is an example of a system page showing the support PIN.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Deploying Citrix NetScaler VPX on Microsoft Azure

Mar 07, 2017

The NetScaler VPX virtual appliance is available as an image in the Microsoft Azure Marketplace. NetScaler VPX on Microsoft Azure Resource Manager (ARM) enables customers to leverage Azure cloud computing capabilities and use NetScaler load balancing and traffic management features for their business needs. You can deploy NetScaler VPX instances on Azure Resource Manager either as standalone instances or as high availability pairs in active-active or active-standby modes.

This document assumes that you are familiar with Microsoft Azure terminology and network details. For information about Microsoft Azure, see the Microsoft Azure documentation at http://azure.microsoft.com/en-us/documentation/.

This document also assumes that you have basic knowledge of a NetScaler appliance. For detailed information about NetScaler appliances, see:

- http://docs.citrix.com/en-us/netscaler/11.html
- http://docs.citrix.com/en-us/netscaler-gateway/11.html

This document provides information about:

- Limitations and Usage Guidelines
- Supported Features
- Network Architecture
- How NetScaler VPX Works on Azure
- Traffic Flow through Port Address Translation
- Traffic Flow through Network Address Translation
- Port Usage Guidelines

For information about NetScaler Gateway configuration for Citrix XenApp and XenDesktop in Azure cloud, see https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/netscaler-vpx-deployment-with-xendesktop-and-xenapp-on-microsoft-azure.pdf.

> **Note**
>
> This document describes how NetScaler VPX works when deployed with Azure Resource Manager (ARM). For information about NetScaler VPX deployment and architecture in Azure cloud services, see the details provided in NetScaler 10.5 http://docs.citrix.com/en-us/netscaler/10-5/vpx/deploy-vpx-on-azure.html.

Limitations and Usage Guidelines

Running the NetScaler VPX load balancing solution on ARM imposes the following limitations:

1. Although you can provision a NetScaler VPX instance in Azure with multiple NICs, the multiple NIC feature has a few constraints as documented in the following Microsoft article: http://azure.microsoft.com/blog/2014/10/30/multiple-vm-nics-and-network-virtual-appliances-in-azure/.
   One limitation is described as “The order of the NICs inside the virtual machine (VM) will be random." If you provision a NetScaler VPX instance with multiple NICs, the instance might become unresponsive if an Azure process or your use of Azure tools shuts down and restarts the virtual machine.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To avoid this issue, Citrix recommends that you deploy NetScaler VPX instances in a single IP mode, which is available only in Azure deployments. For more information about the single IP mode and the traffic flow in a NetScaler VPX on Azure deployment, see How NetScaler VPX Works on Azure.

2. The Azure architecture does not accommodate support for the following NetScaler features:
   Clustering

   IPv6

   Gratuitous ARP (GARP)

   L2 Mode

   Tagged VLAN

   Dynamic Routing

   Virtual MAC (VMAC)

   USIP

   GSLB

   CloudBridge Connector

3. The Intranet IP (IIP) feature is not supported, because Azure does not provide the pool of IP addresses required for this feature. IIP is frequently used in VOIP, SIP, or server-initiated-connection deployment.
4. If you expect that you might have to shut down and temporarily deallocate the NetScaler VPX virtual machine at any time, assign a static Internal IP address while creating the virtual machine. If you do not assign a static internal IP address, Azure might assign the virtual machine a different IP address each time it restarts, and the virtual machine might become inaccessible.
5. In an Azure deployment, only the following NetScaler VPX models are supported: VPX 10, VPX 200, and VPX 1000. These virtual appliances can be deployed on any instance type that has two or more cores and more than 2 GB memory. See the NetScaler VPX datasheet: https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/citrix-netscaler-vpx-data-sheet.pdf
6. The 'deployment ID' that is generated by Azure during virtual machine provisioning is not visible to the user in ARM. You cannot use the deployment ID to deploy NetScaler VPX appliance on ARM.

Supported Features

At initial startup, with the default license, the following features are available on a NetScaler VPX instance running in Azure:

- Web Logging
- Content Switching
- Load Balancing
- SSL Offloading
- Content Filtering
- SSL VPN (Maximum users = 5) (Maximum ICA users = unlimited)
- Rewrite
- Responder
- HTML Injection

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Web Interface on NS
- Appflow
- vPath
- Strong encryption

The Model Number ID shows the overall NetScaler VPX throughput, which is 5 Mbps of throughput at the initial startup. Additional NetScaler features and throughput are enabled by applying a NetScaler VPX license obtained from Citrix.
For a XenApp and XenDesktop deployment, a VPN virtual server on a NetScaler appliance can be configured in the following modes:

- Basic mode, where the ICAOnly VPN virtual server parameter is set to ON. The Basic mode works fully on an unlicensed NetScaler VPX instance.
- Smart-Access mode, where the ICAOnly VPN virtual server parameter is set to OFF. The Smart-Access mode works for only 5 AAA session users on an unlicensed NetScaler VPX instance.

> **Note**
>
> To configure the Smart Control feature, you must apply a platinum license to the NetScaler VPX instance.

Network Architecture

In ARM, a NetScaler VPX virtual machine (VM) resides in a virtual network. A virtual Network Interface Card (NIC) is created on each NetScaler VM. The network security group (NSG) configured in the virtual network is bound to the NIC, and together they control the traffic flowing into the VM and out of the VM.

The NSG forwards the requests to the NetScaler VPX instance, and the VPX instance sends them to the servers. The response from a server follows the same path in reverse. The NSG can be configured to control a single VPX VM, or, with subnets and virtual networks, can control traffic in multiple VPX VM deployment.

The NIC contains network configuration details such as the virtual network, subnets, internal IP address, and Public IP address.

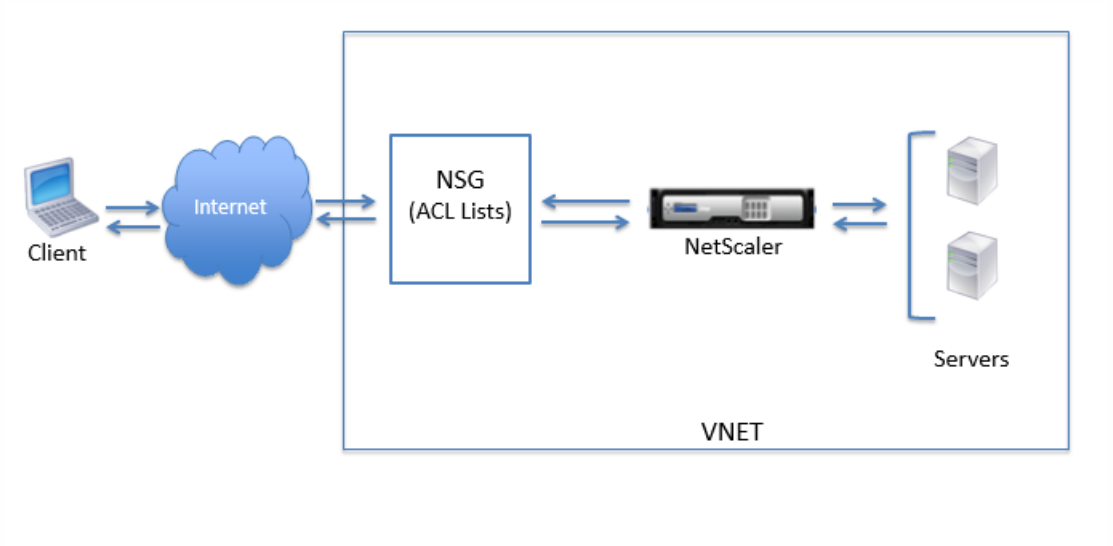
While on ARM, it is good to know the following IP addresses used to access the VMs:

- Public IP (PIP) address is the Internet-facing IP address configured directly on the virtual NIC of the NetScaler VM. This allows you to directly access a VM from the external network without the need to configure inbound and outbound rules on the NSG.
- NetScaler IP (NSIP) address is internal IP address configured on the VM. It is non-routable.
- Virtual IP address (VIP) is configured by using the NSIP and a port number. Clients access NetScaler services through the PIP address, and when the request reaches the NIC of the NetScaler VPX VM or the Azure load balancer, the VIP gets translated to internal IP (NSIP) and internal port number.
- Internal IP address is the private internal IP address of the VM from the virtual network's address space pool. This IP address cannot be reached from the external network. This IP address is by default dynamic unless you set it to static. Traffic from the internet is routed to this address according to the rules created on the NSG. The NSG works with the NIC to selectively send the right type of traffic to the right port on the NIC, which depends on the services configured on the VM.

In this document, PIP, VIP, and Instance Level PIP (ILPIP) mean the same thing and are used interchangeably.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following figure shows how traffic flows from a client to a server through a NetScaler VPX instance provisioned in ARM.



How NetScaler VPX Works on Azure

In an on-premises deployment, a NetScaler VPX instance requires, at least three IP addresses:

- Management IP address, called the NetScaler IP (NSIP) address
- Subnet IP (SNIP) address for communicating with the server farm
- Virtual server IP (VIP) address for accepting client requests

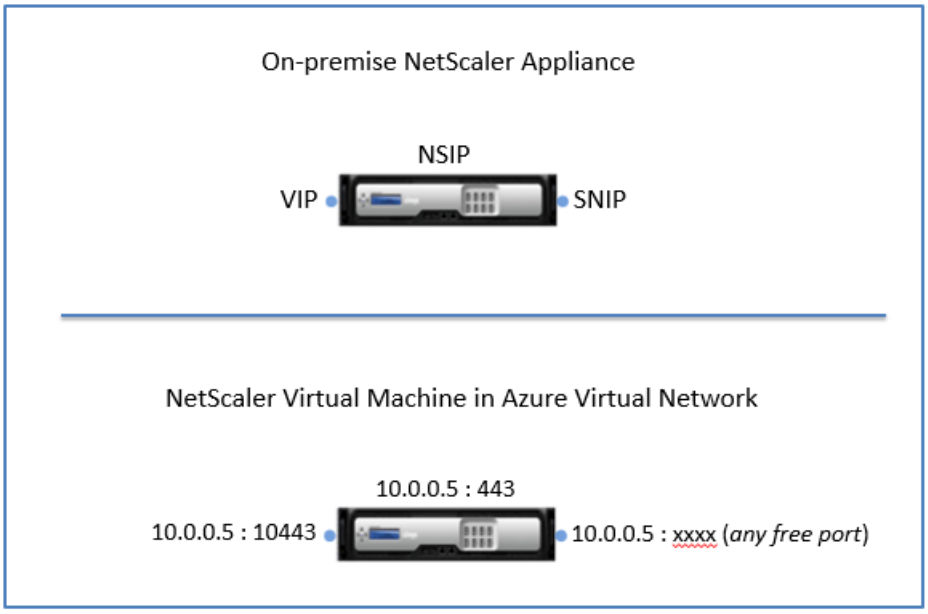
In an Azure deployment, only one IP address is assigned to an instance during provisioning through DHCP, and it is a private (internal) address.

Although you can provision a NetScaler VPX instance in Azure with multiple NICs, the multiple NIC feature has a few constraints that might make the instance unresponsive. For details about this constraint, see Limitations and Usage Guidelines.

To avoid this limitation, you can deploy NetScaler VPX instance in Azure with a single IP architecture, where the three IP functions of a NetScaler appliance are multiplexed onto one IP address. This single IP address uses different port numbers to function as the NSIP, SNIP, and VIP.

The following image illustrates how a single IP address is used to perform the functions of NSIP, SNIP, and VIP.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Note

The single IP mode is available only in Azure deployments. This mode is not available for a NetScaler VPX instance on your premises, on AWS, or in in other type of deployment.

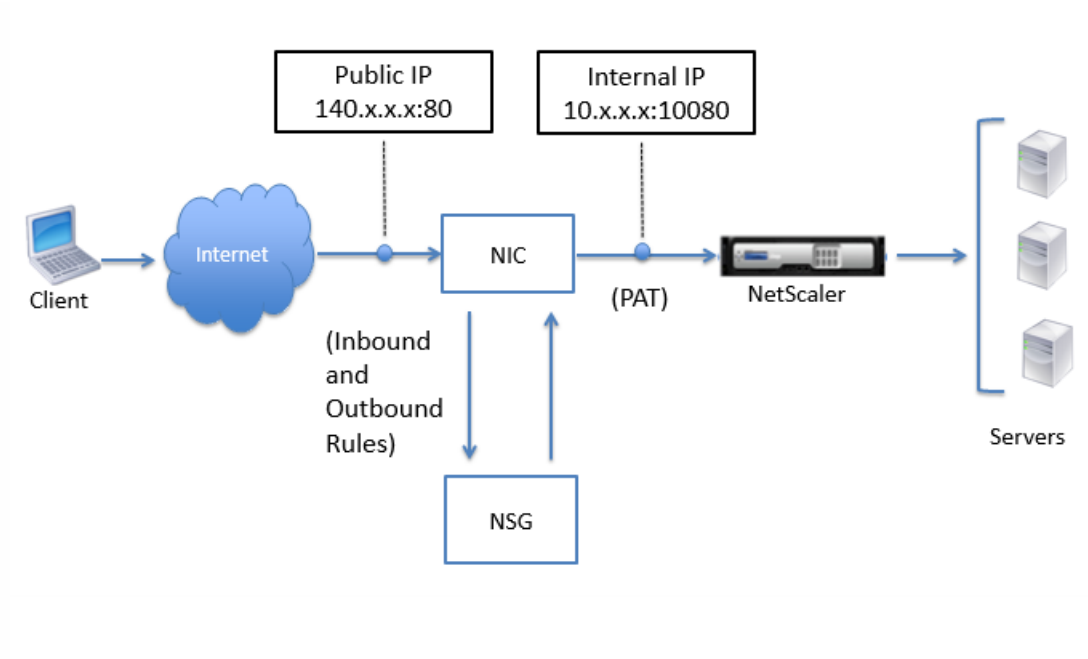
Traffic Flow through Port Address Translation

In an Azure deployment, when you provision the NetScaler VPX instance as a virtual machine (VM), Azure assigns a Public IP address and an internal IP address (non-routable) to the NetScaler virtual machine. Inbound and Outbound rules are defined on the NSG for the NetScaler instance, along with a public port and a private port for each rule defined. The NetScaler instance listens on the internal IP address and private port.

Any external request is received on the NetScaler VPX VM's virtual NIC. The NIC is bound to the NSG, which specifies the private IP and private port combination into which to translate the request's destination address and port (the Public IP address and port). ARM performs port address translation (PAT) to map the Public IP address and port to the internal IP address and private port of the NetScaler virtual machine, and forwards the traffic to the virtual machine.

The following figure shows how Azure performs port address translation to direct traffic to the NetScaler internal IP address and private port.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



In this example, the Public IP address assigned to the VM is 140.x.x.x, and the internal IP address is 10.x.x.x. When the inbound and outbound rules are defined, public HTTP port 80 is defined as the port on which the client requests are received, and a corresponding private port, 10080, is defined as the port on which the NetScaler virtual machine listens. The client request is received on the Public IP address 140.x.x.x at port 80. Azure performs port address translation to map this address and port to internal IP address 10.x.x.x on private port 10080 and forwards the client request.

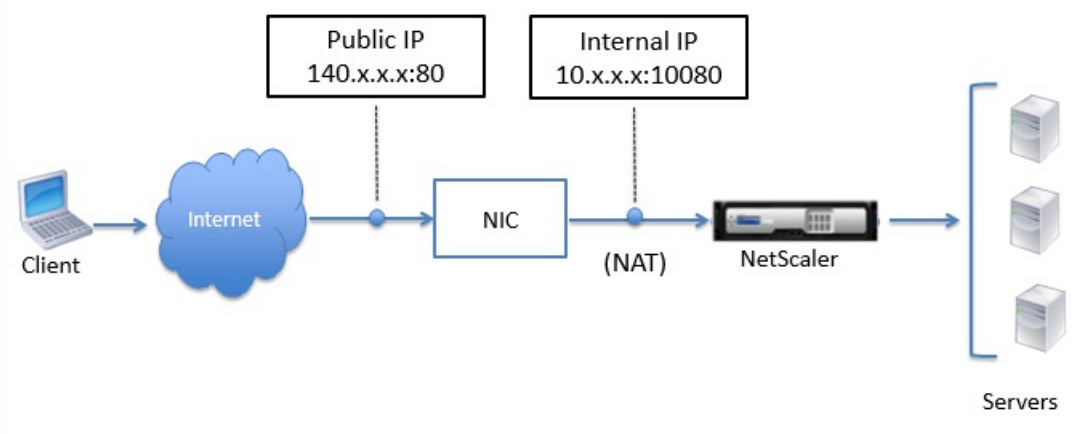
For information about port usage guidelines while, see Port Usage Guidelines.

For information about NSG and access control lists, see https://azure.microsoft.com/en-in/documentation/articles/virtual-networks-nsg/.

Traffic Flow Through Network Address Translation

You can also request a Public IP (PIP) address for your NetScaler virtual machine (instance level). If you use this direct PIP at the VM level, you need not define inbound and outbound rules to intercept the network traffic. The incoming request from the Internet is received on the VM directly. Azure performs network address translation (NAT) and forwards the traffic to the internal IP address of the NetScaler instance.

The following figure shows how Azure performs network address translation to map the NetScaler internal IP address.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

In this example, the Public IP assigned to the NSG is 140.x.x.x and the internal IP address is 10.x.x.x. When the inbound and outbound rules are defined, public HTTP port 80 is defined as the port on which the client requests are received, and a corresponding private port, 10080, is defined as the port on which the NetScaler virtual machine listens. The client request is received on the Public IP address (140.x.x.x). Azure performs network address translation to map the PIP to the internal IP address 10.x.x.x on port 10080, and forwards the client request.

> **Note**
>
> NetScaler VPX VMs in high availability are controlled by external or internal load balancers that have inbound rules defined on them to control the load balancing traffic. The external traffic is first intercepted by these load balancers and the traffic is diverted according to the load balancing rules configured, which has backend pools, NAT rules, and health probes defined on the load balancers.

Port Usage Guidelines

You can configure additional inbound and outbound rules n NSG while creating the NetScaler virtual machine or after the virtual machine is provisioned. Each inbound and outbound rule is associated with a public port and a private port.

Before configuring NSG rules, note the following guidelines regarding the port numbers you can use:

1. The following ports are reserved by the NetScaler virtual machine. You cannot define these as private ports when using the Public IP address for requests from the Internet.
   Ports 21, 22, 80, 443, 8080, 67, 161, 179, 500, 520, 3003, 3008, 3009, 3010, 3011, 4001, 5061, 9000, 7000.

   However, if you want Internet-facing services such as the VIP to use a standard port (for example, port 443) you have to create port mapping by using the NSG. The standard port is then mapped to a different port that is configured on the NetScaler for this VIP service.

   For example, a VIP service might be running on port 8443 on the NetScaler instance but be mapped to public port 443. So, when the user accesses port 443 through the Public IP, the request is actually directed to private port 8443.
   Note: If you access the NetScaler instance in Network Access mode (Full SSL VPN), you need to map the private port 8443 to the public port 8443.
2. Public IP address does not support protocols in which port mapping is opened dynamically, such as passive FTP or ALG.
3. Azure load balancer does not work with Public IP address. High availability does not work for traffic that uses PIP associated with VPX instance instead of PIP configured on the load balancer. For more information about configuring NetScaler VPX HA in ARM, see Configuring NetScaler VPX in High Availability Mode in Azure.
4. In a NetScaler Gateway deployment, you need not configure a SNIP address, because the NSIP can be used as a SNIP when no SNIP is configured.
   You must configure the VIP address by using the NSIP address and some nonstandard port number. For call-back configuration on the backend server, the VIP port number has to be specified along with the VIP URL (for example, url:port).

> **Note**
>
> Note: In ARM, a NetScaler VPX VM is associated with two IP addresses. Public IP address and Internal IP address. While the external traffic connects to the PIP, the internal IP address or the NSIP is non-routable. To configure VIP in VPX, use this combination of

© 1999-2017 Citrix Systems, Inc. All rights reserved.

internal IP address and port number.

Example:

If the VPN virtual server FQDN is vip.test.com, and the VPN virtual server is running on port 8443, the call-back URL will be: https://vip.test.com:8443.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring NetScaler VPX in a Standalone Mode in Azure Resource Manager (ARM)

May 20, 2016

Provision a single instance of NetScaler VPX in Microsoft Azure Resource Manager (ARM) portal in a standalone mode by creating the NetScaler VPX virtual machine in ARM, and by configuring other resources.

Before You Begin

Make sure that you have the following:

- A Microsoft Azure user account
- Access to Microsoft Azure Resource Manager
- Microsoft Azure SDK
- Microsoft Azure PowerShell

On the Microsoft Azure Portal page (https://portal.azure.com), log on to the Azure Resource Manager portal by providing your user name and password.

> **Note**
>
> In ARM portal, clicking an option in one pane opens a new pane to the right. Navigate from one pane to another to configure your device.

Summary of Configuration Tasks

The following list provides a summary of tasks that you have to perform to configure NetScaler VPX in standalone mode.

1. Configure a resource group
2. Configure a network security group
3. Configure virtual network and its subnets
4. Configure a storage account
5. Configure an availability set
6. Configure a NetScaler VPX instance

The steps required to perform these tasks are provided in this section.

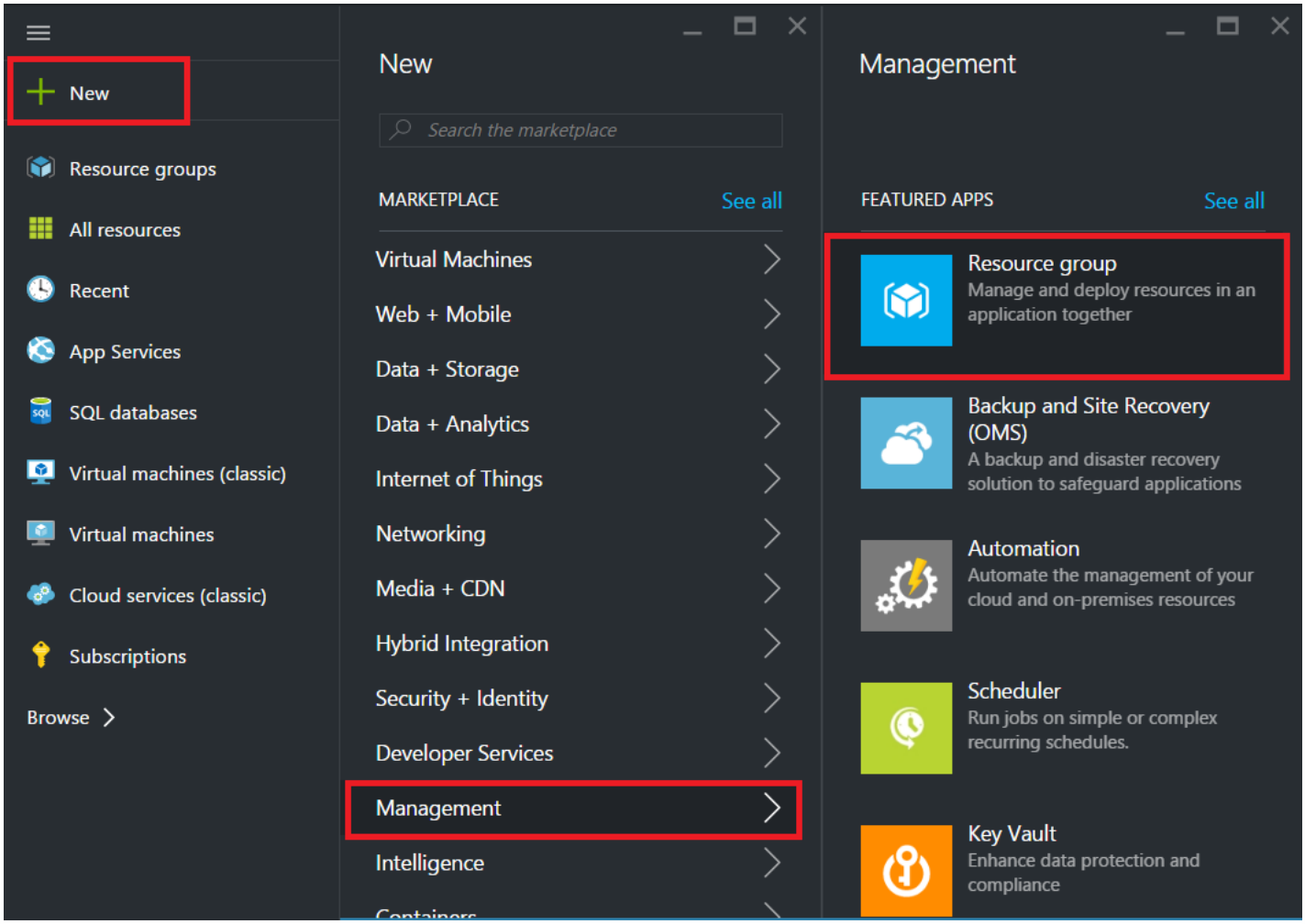
> **Note**
>
> You have to perform all these common tasks to configure NetScaler VPX in both standalone and in high availability mode. To deploy NetScaler VPX instances in high availability mode, perform these tasks and continue with the tasks in Configuring NetScaler VPX in High Availability Mode.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

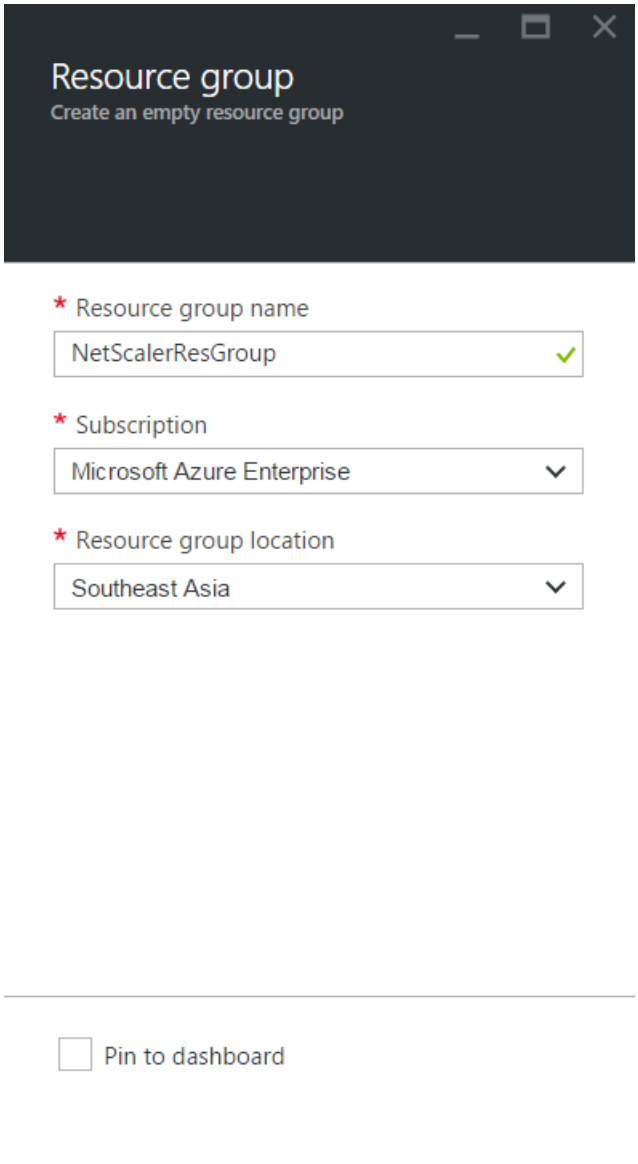
Configuring a Resource Group

Create a new resource group that is a container for all your resources. Use the resource group to deploy, manage, and monitor your resources as a group.

1. Click **New** > **Management** > **Resource group**.
2. In the **Resource group** pane, enter the following details:
   - Resource group name
   - Resource group location
3. Click **Create**.



© 1999-2017 Citrix Systems, Inc. All rights reserved.



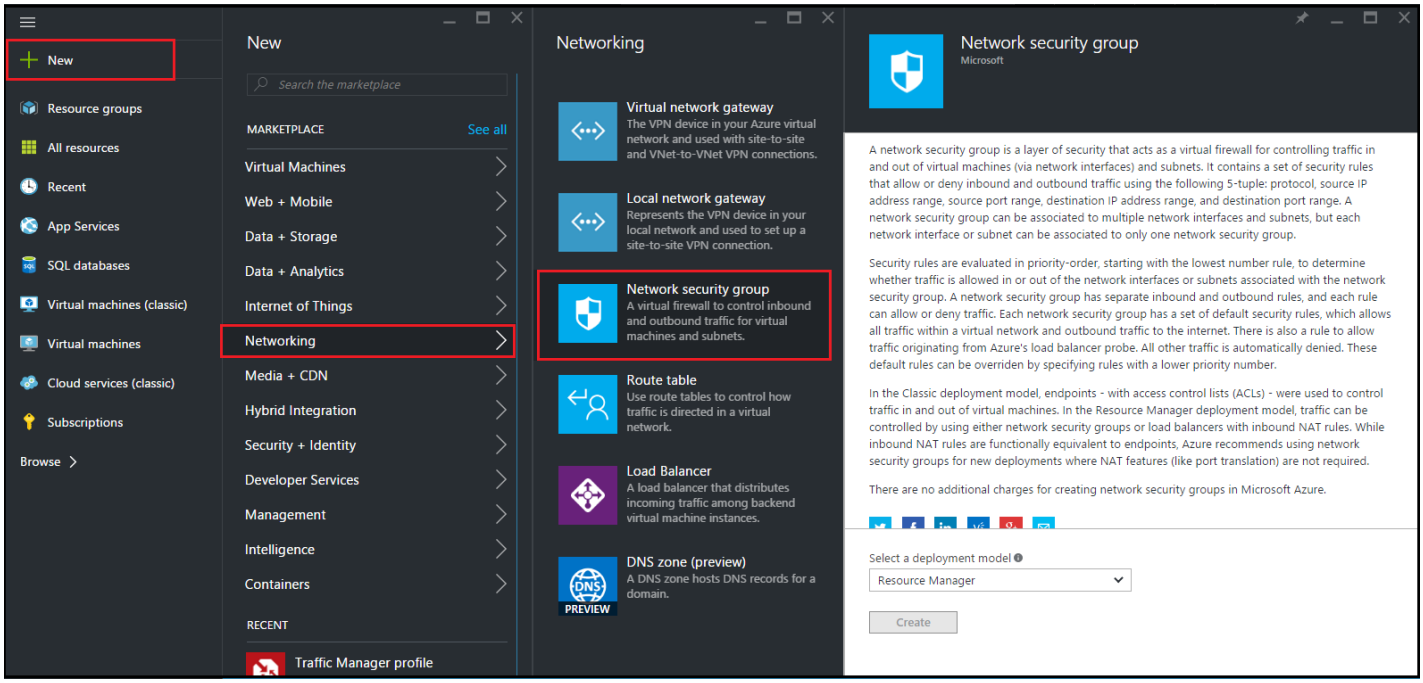
Configuring a Network Security Group

Create a network security group (NSG) to assign inbound and outbound rules to control the incoming and outgoing traffic within the virtual network. NSG allows you to define security rules for a single virtual machine and also to define security rules for a virtual network subnet.

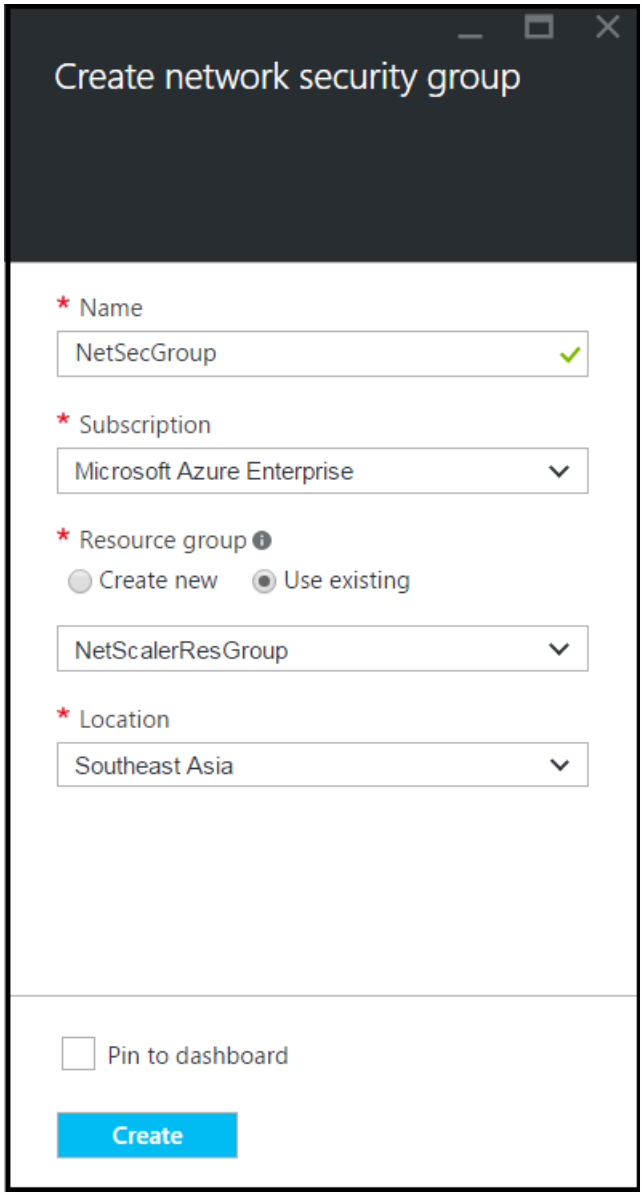
1. Click **New** > **Networking** > **Network security group**.
2. In the **Create network security group** pane, enter the following details, and then click **Create**.
   - Name - type a name for the security group
   - Resource group - select the resource group from the drop-down list

> **Note**
>
> Ensure that you have selected the correct location. The list of resources that appear in the drop-down list is different for different locations.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.



Configuring a Virtual Network and Subnets

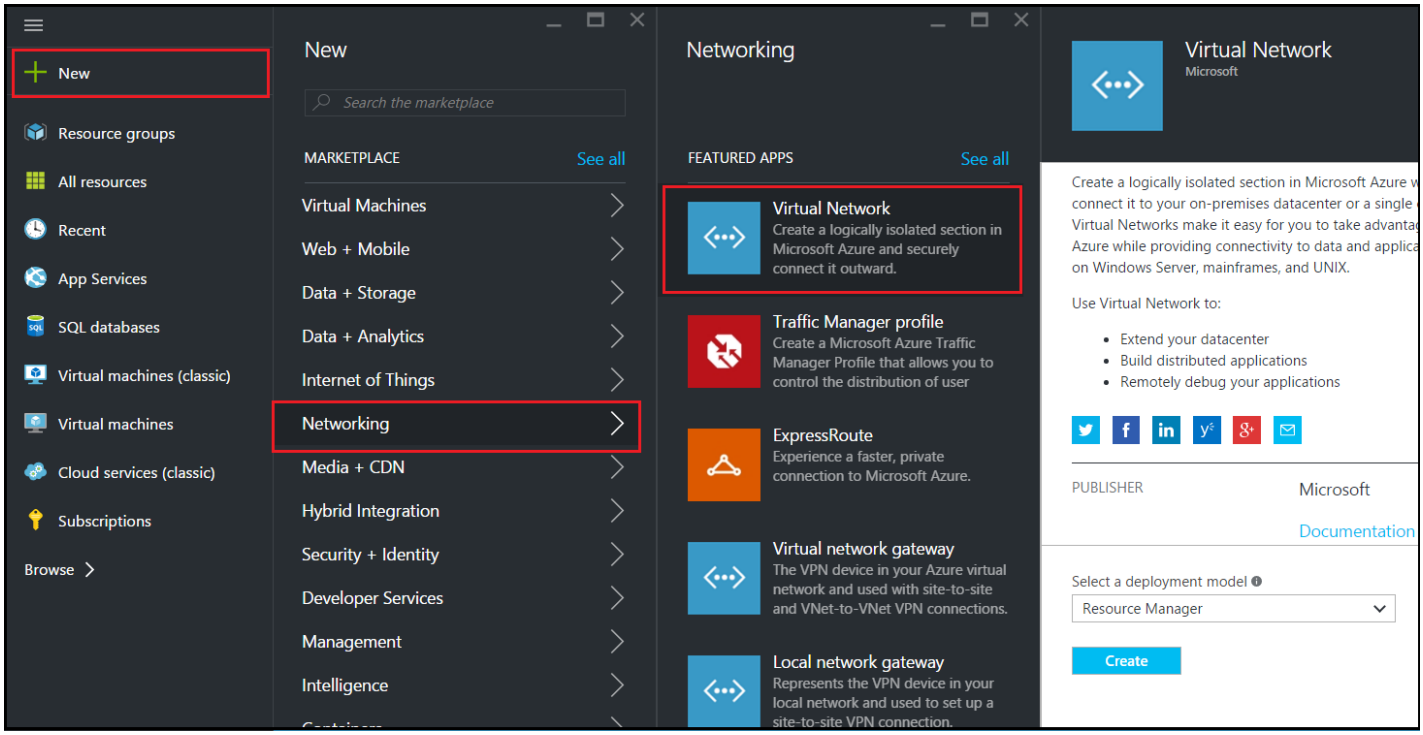
Virtual networks in ARM provide a layer of security and isolation to your services. Virtual machines and services that are part of the same virtual network can access each other.

For example, create a virtual network with a reserved CIDR block of 192.168.0.0/16 and two subnets having CIDR blocks as 192.168.1.0/24 and 192.168.2.0/24 respectively.

1. Click **New** > **Networking** > **Virtual Network**.

2. In the **Virtual Network** pane, select **Resource Manager** from the drop-down list, and click **Create**.

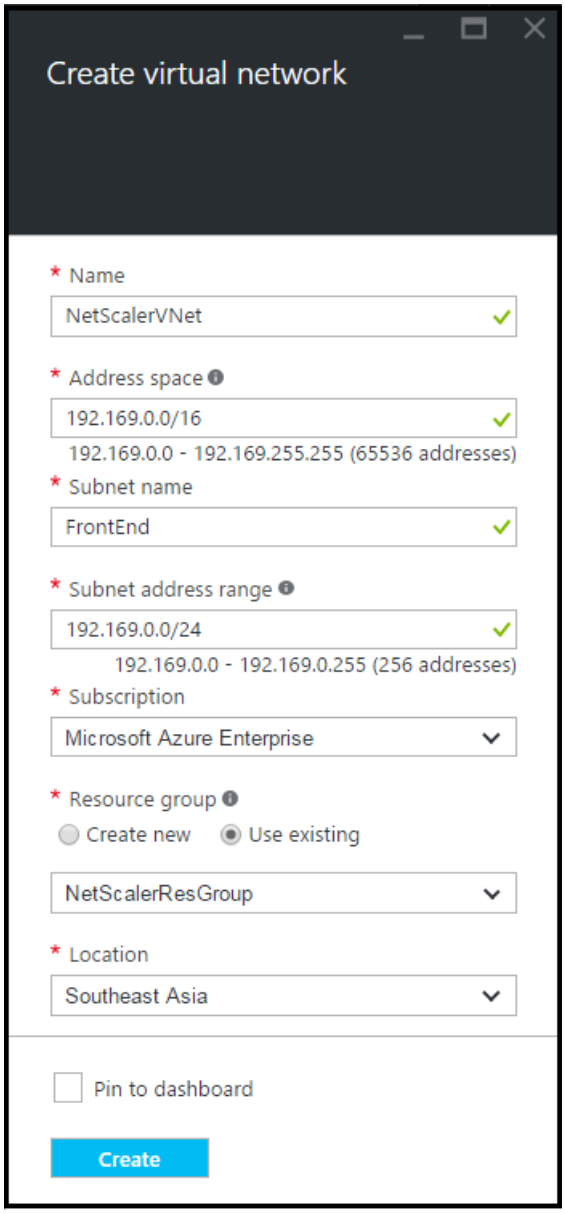
 © 1999-2017 Citrix Systems, Inc. All rights reserved.



3. In the **Create virtual network** pane, enter the following values, and then click **Create**.

- Name of the virtual network
- Address space - type the reserved IP address block for the virtual network
- Subnet - type the name of the first subnet (you will create the second subnet later in this step)
- Subnet address range - type the reserved IP address block of the subnet
- Resource group - select the resource group, that was created earlier, from the drop-down list

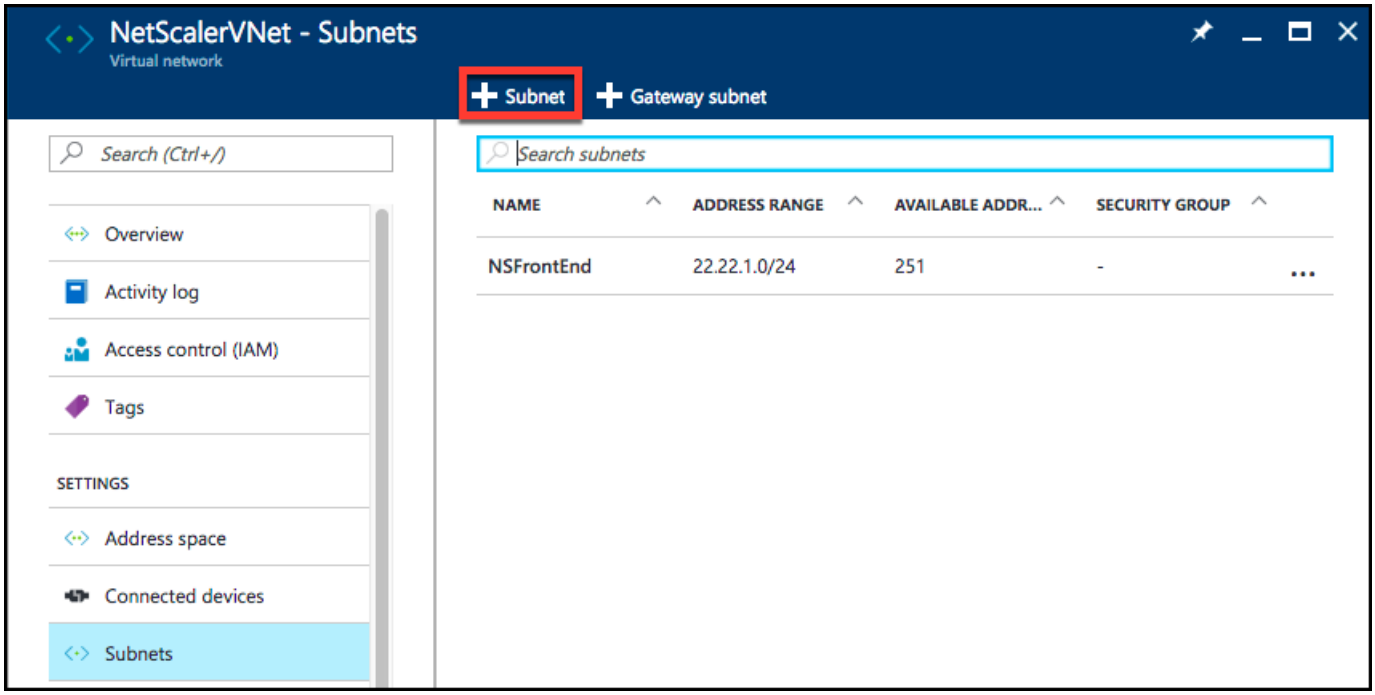
© 1999-2017 Citrix Systems, Inc. All rights reserved.



Configuring the second subnet

1. Select the newly created virtual network from **All resources** pane, and in the **Settings** pane, click **Subnets**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

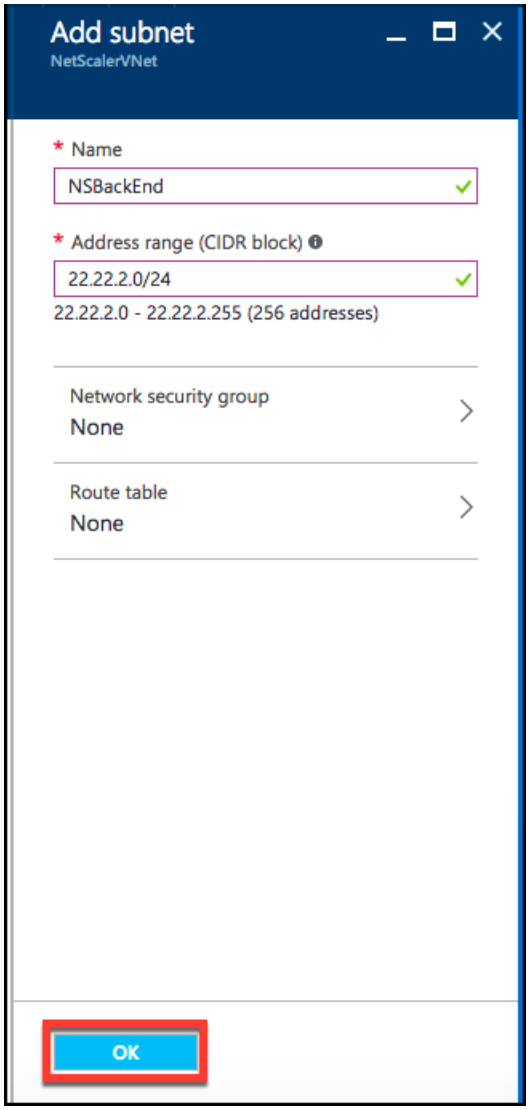


2. Click **+Subnet**, and create the second subnet by entering the following details.

- Name of the second subnet
- Address range - type the reserved IP address block of the second subnet
- Network security group - select the NSG from the drop-down list

3. Click **Create**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



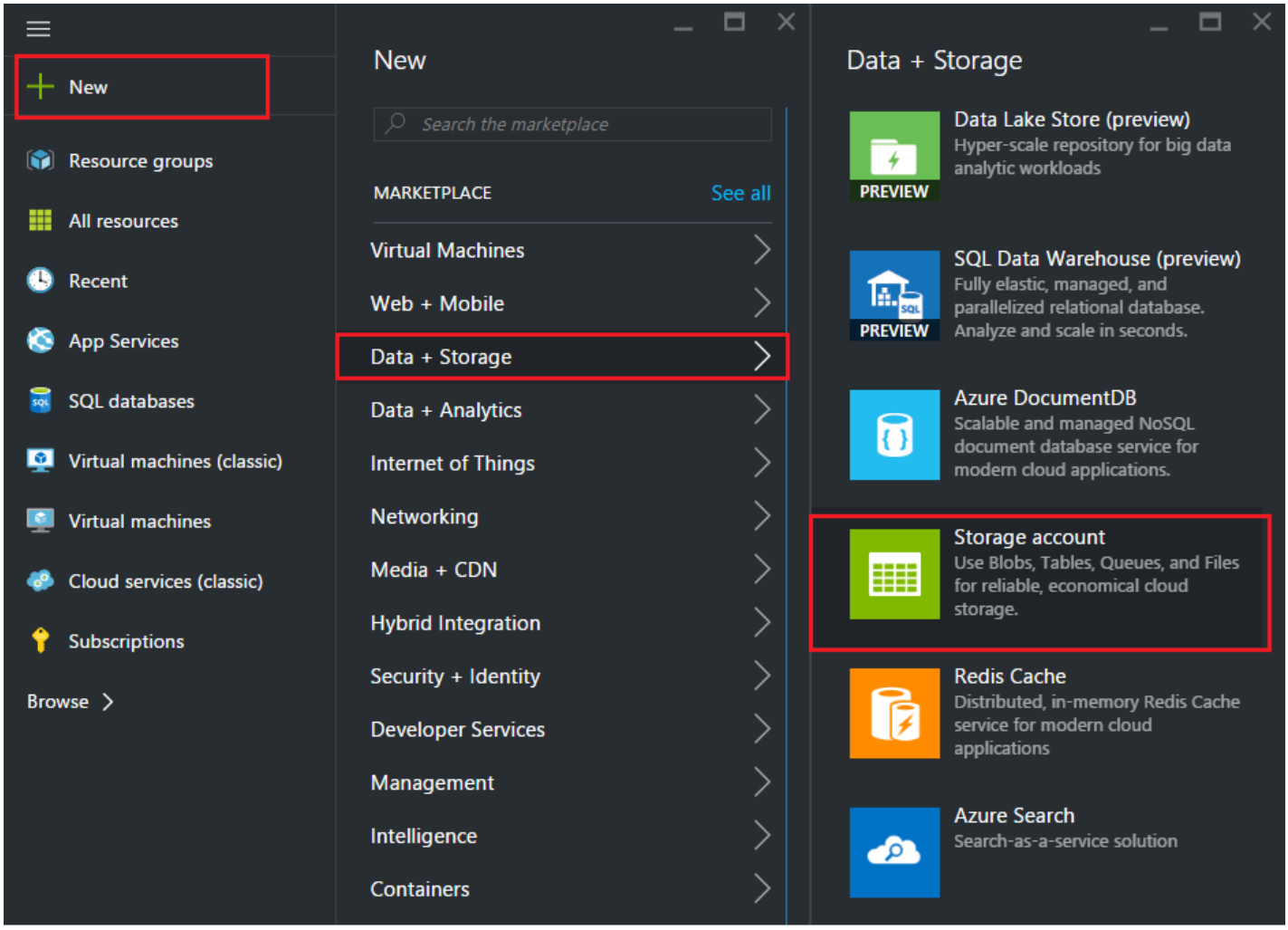
Configuring a Storage Account

The ARM IaaS infrastructure storage includes all services where we can store data in the form of blobs, tables, queues, and files. You can also create applications using these forms of storage data in ARM.

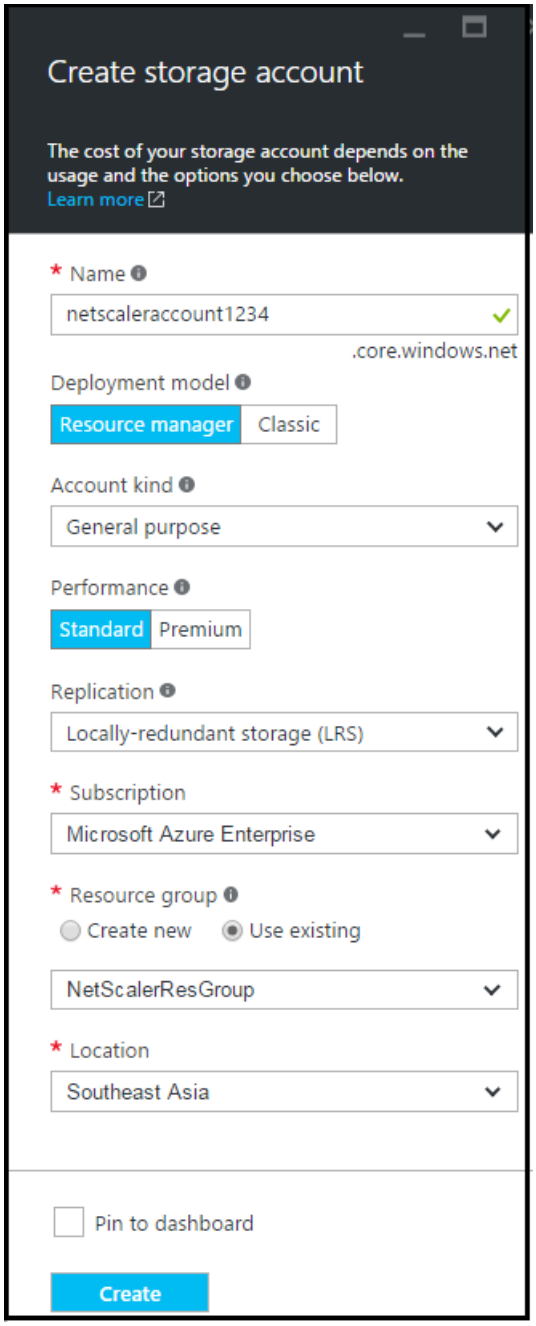
Create a storage account to store all your data.

1. Click **+New** > **Data + Storage** > **Storage account**.
2. In the **Create storage account** pane, enter the following details:
   - Name of the account
   - Deployment mode - make sure to select **Resource Manager**
   - Account kind - select **General purpose** from the drop-down list
   - Replication - select **Locally redundant storage** from the drop-down list
   - Resource group - select the newly created resource group from the drop-down list
3. Click **Create**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

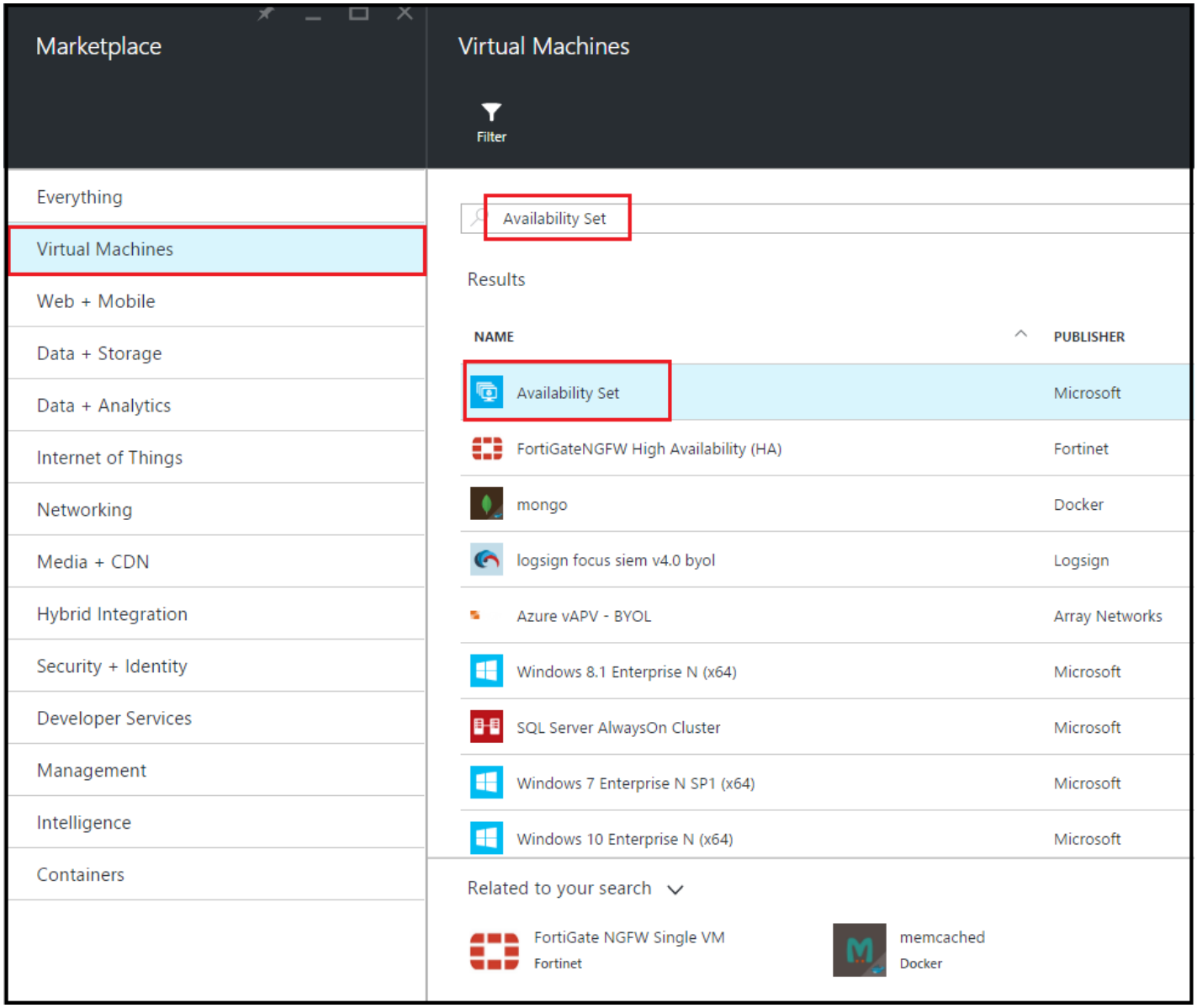


Configuring an Availability Set

An availability set guarantees that at least one virtual machine is kept up and running in case of planned or unplanned maintenance. Two or more virtual machines under the same 'availability set' are placed on different fault domains to achieve redundant services.

1. Click **+New**.
2. Click **See all** in the MARKETPLACE pane, and click **Virtual Machines**.
3. Type **Availability set,** and click **Availability set** entity from the list.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

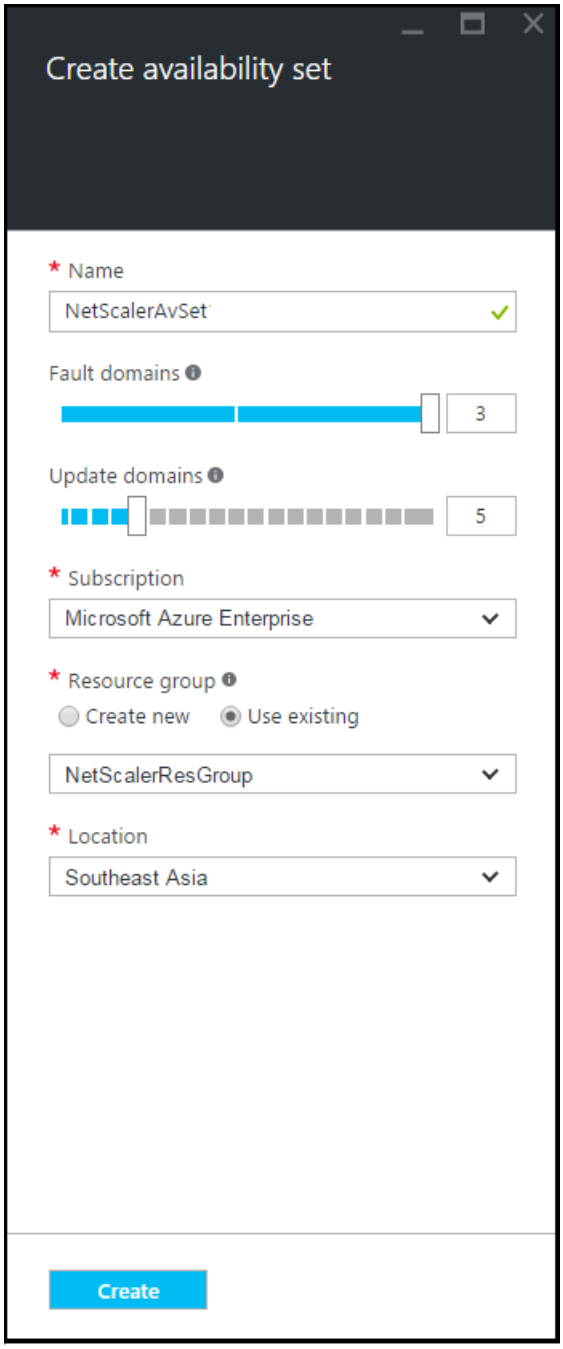


4. Click **Create,** and in the **Create availability set** pane, enter the following details:

- Name of the set
- Resource group - select the newly created resource group from the drop-down list

5. Click **Create**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Configuring a NetScaler VPX Instance

Create an instance of NetScaler VPX in the virtual network. Obtain the NetScaler VPX image from the Azure marketplace, and then use the Azure Resource Manager portal to create a NetScaler VPX instance.

Before you begin creating the NetScaler VPX instance, make sure that you have created a virtual network with required subnets in which the instance will reside. You can create virtual networks during VM provisioning, but without the flexibility to create different subnets. For information about creating virtual networks, see http://azure.microsoft.com/en-us/documentation/articles/create-virtual-network/.

Optionally, configure DNS server and VPN connectivity that allows a virtual machine to access Internet resources.

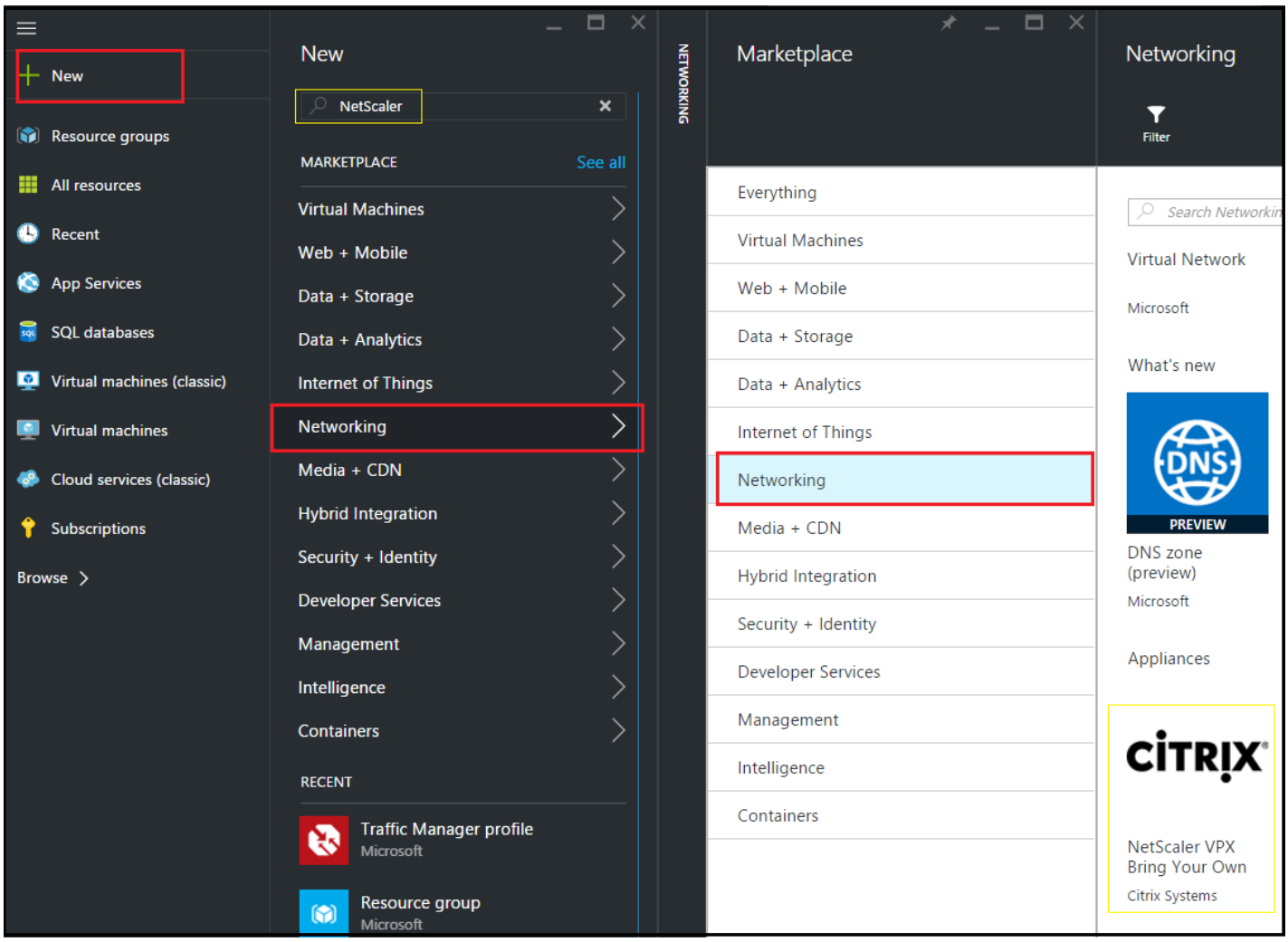
 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Citrix recommends that you create resource group, network security group, virtual network, and other entities before you provision the NetScaler VPX virtual machine, so that the network information is available during provisioning.

1. Click **+New** > **Networking**.
2. Click **See All,** and in the Networking pane, click **Citrix NetScaler VPX Bring Your Own License**.

Alternatively, as a quick way to find any entity on ARM portal, you can also type the name of the entity in the Azure Marketplace search box and press <Enter>. For example, type **NetScaler** in the search box to find the Citrix NetScaler images.

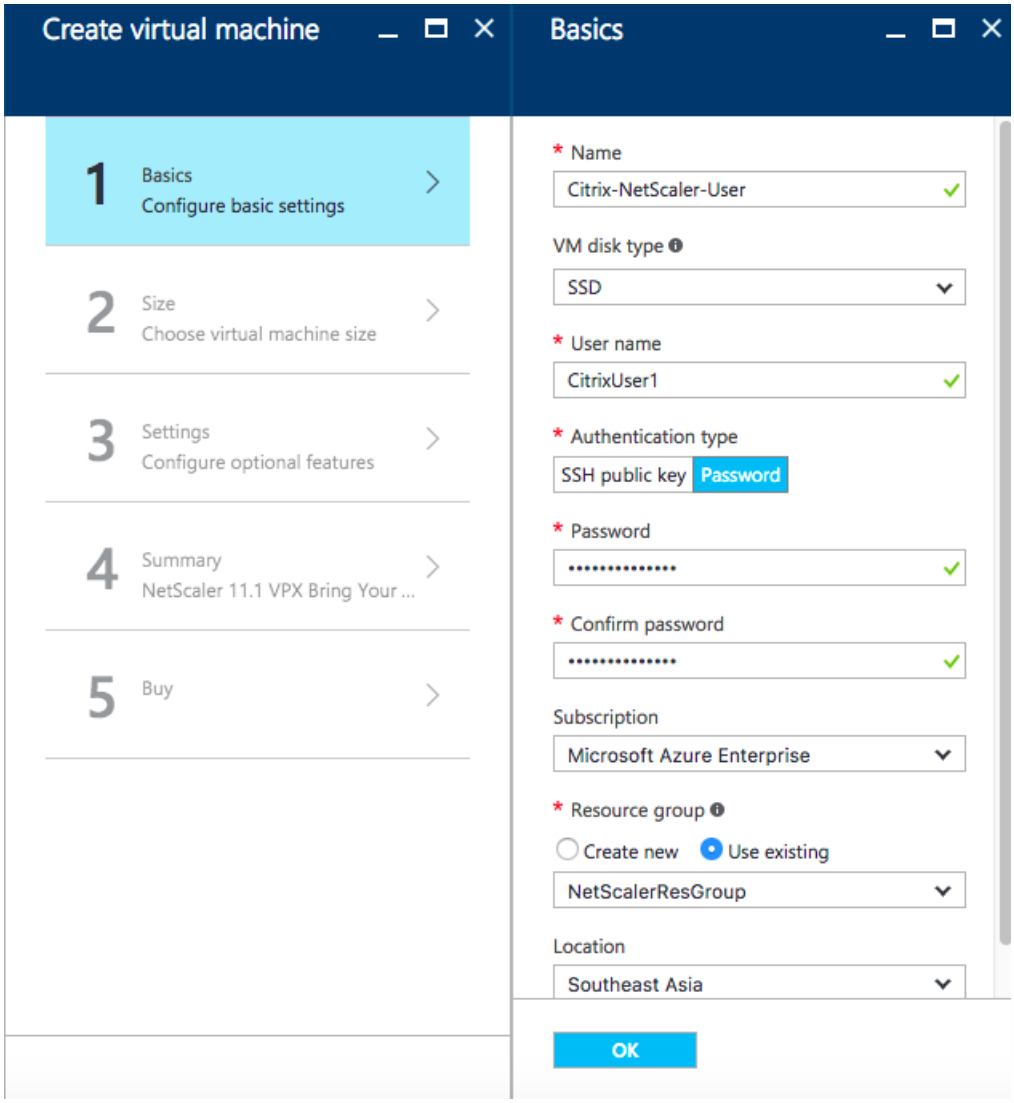


Note

Make sure that you select the latest NetScaler VPX image. Your Citrix NetScaler image might have the release number in the title.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. On the **NetScaler VPX Bring Your Own License** page, from the drop-down list, select **Resource Manager,** and click **Create**.



4. In the Create virtual machine pane, specify the required values in each section to create a virtual machine. Click OK in each section to save your configuration.

Basic

- Name - specify a name for the NetScaler VPX instance
- User name and Password - specify a user name and password to access the resources in the resource group that you have created
- Authentication Type - select SSH Public Key or Password
- Resource group - select the resource group you have created from the drop-down list.

You can create a resource group here, but Citrix recommends that you create a resource group from Resource groups in Azure Resource Manager and then select the group from the drop-down list

Size

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Select A2 Standard or larger for a single-instance NetScaler VPX deployment
- Click **Select**

Settings

- Select the default (Standard) disk type
- Storage account - select the storage account
- Virtual network - select the virtual network
- Subnet - set the subnet address
- Public IP address - select the type of IP address assignment
- Network security group - select the security group that you have created. Ensure that inbound and outbound rules are configured in the security group.
- Availability Set - select the availability set from the drop-down box

Summary

The configuration settings are validated and the **Summary** page displays the result of the validation. If the validation fails, the **Summary** page displays the reason of the failure. Go back to the particular section and make changes as required. If the validation passes, click **OK**.

Buy

Review the offer details and legal terms on the **Purchase** page, and click **Purchase**.

For high availability deployment, create two independent instances of NetScaler VPX in the same availability set and in the same resource group to deploy them in active-standby and active-active configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring NetScaler VPX in High Availability Mode in Azure Resource Manager (ARM)

May 20, 2016

In a Microsoft Azure Resource Manager (ARM) deployment, a high availability configuration of two NetScaler VPX virtual machines is achieved by using the Azure load balancer, which distributes the client traffic across the virtual servers configured on both the NetScaler instances. Two types of Azure load balancers are available for high availability:

- **Azure external load balancer**: If the client traffic originates from the Internet, you have to deploy the external load balancer between the Internet and the NetScaler VPX instances to distribute client traffic.
- **Azure internal load balancer**: If the client traffic originates from within the virtual network, or is forwarded by a gateway or firewall within the virtual network, you have to deploy the internal load balancer to distribute client traffic.

To achieve high availability on Azure, you must add the two NetScaler virtual machines as a load balanced set and configure the network security group (NSG).

When two NetScaler VPX instances are configured in active-active mode, both instances must have the same configuration. The client traffic is distributed across the virtual servers in both the instances by the Azure load balancer. The VIP addresses in both the instances are different and should match the NetScaler IP (NSIP) address of that NetScaler VPX instance.

The active-passive mode provides failover capability. In this mode, the VPX instances synchronize their configuration states. When the primary instance fails, the secondary instance takes over.

For information about high availability in NetScaler appliance, see http://docs.citrix.com/en-us/netscaler/11/system/high-availability-introduction.html

Before You Begin

Note the following before you begin configuring the NetScaler VPX instances in high availability mode in the Azure virtual network.

- The two NetScaler virtual machines that you want to add to a load balanced set should be provisioned in the same virtual network.
- A load balanced set applies only to the default NIC of the virtual instance. Therefore, the VIP has to be configured on the default NIC of the NetScaler VPX.
- In an active-passive deployment, the Azure load balancer monitors both the primary and the secondary NetScaler virtual machine by sending them TCP probes. These TCP probes are sent on port 9000.

Summary of Tasks to Configure NetScaler VPX in a High Availability Mode

The following list provides a summary of tasks that you have to perform to configure NetScaler VPX in high availability mode.

1. Configure a resource group
2. Configure a network security group
3. Configure virtual network and its subnets
4. Configure a storage account

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Configure an availability set
6. Configure a NetScaler VPX instance
7. Configure internal and external load balancers
8. Configure health probes
9. Configure backend pools
10. Configure NAT rules
11. Configure load balancing rules

> **Note**
>
> The first six tasks are common for configuring NetScaler VPX in both standalone and in high availability mode. To perform these common tasks, see Configuring NetScaler VPX in Standalone Mode. The steps required to perform the tasks specific to high availability deployment are provided here.

After configuring all the resources, you can configure the virtual machines in high availability mode with either an external load balancer or with an internal load balancer.

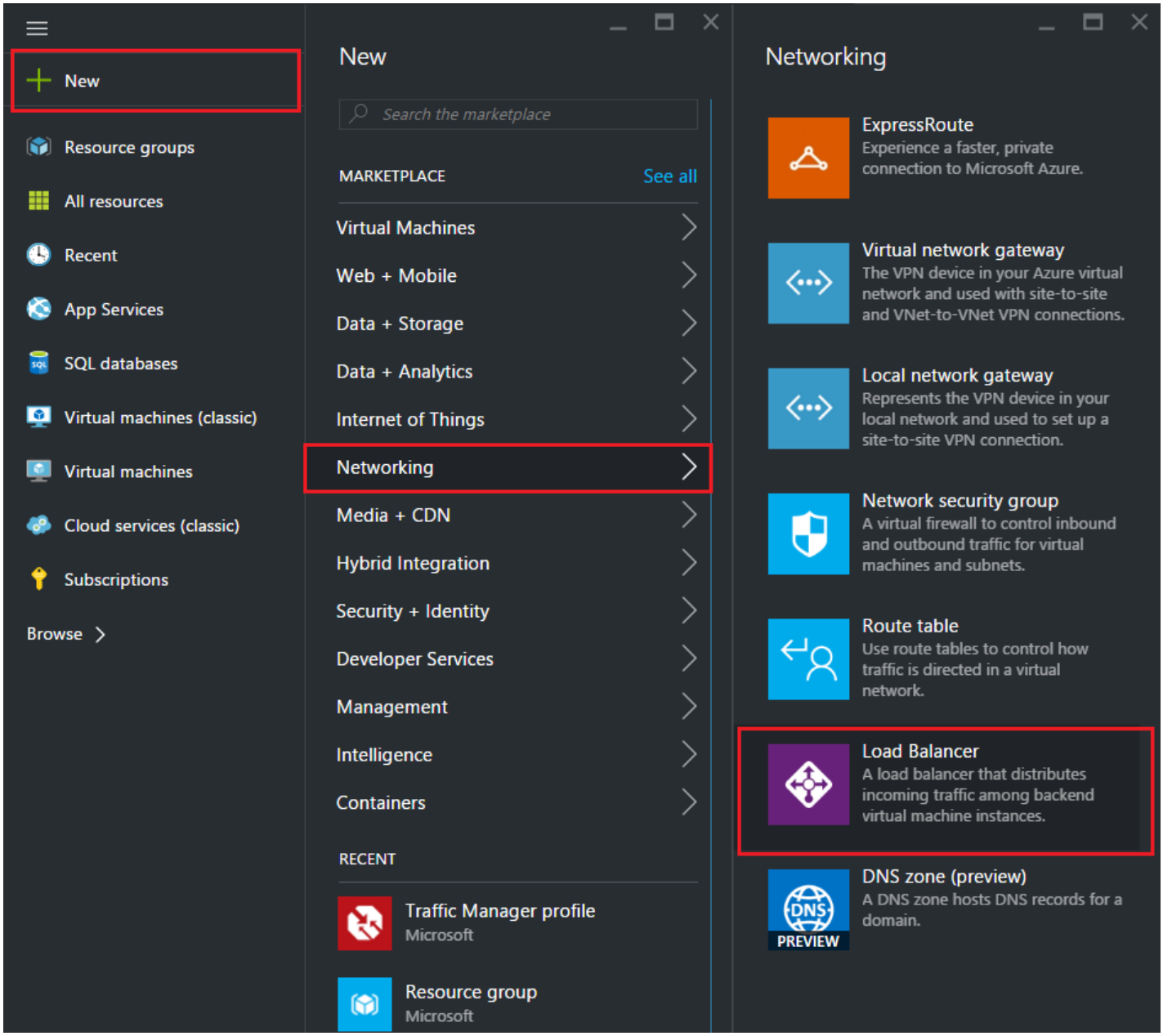
Configuring Internal and External Load Balancers

Create an Azure load balancer to use the load balancing features of ARM to distribute traffic between virtual machines that are part of the same virtual network. The load balancing features of Azure load balancer can load balance level 4 traffic, and support only TCP and UDP traffic.

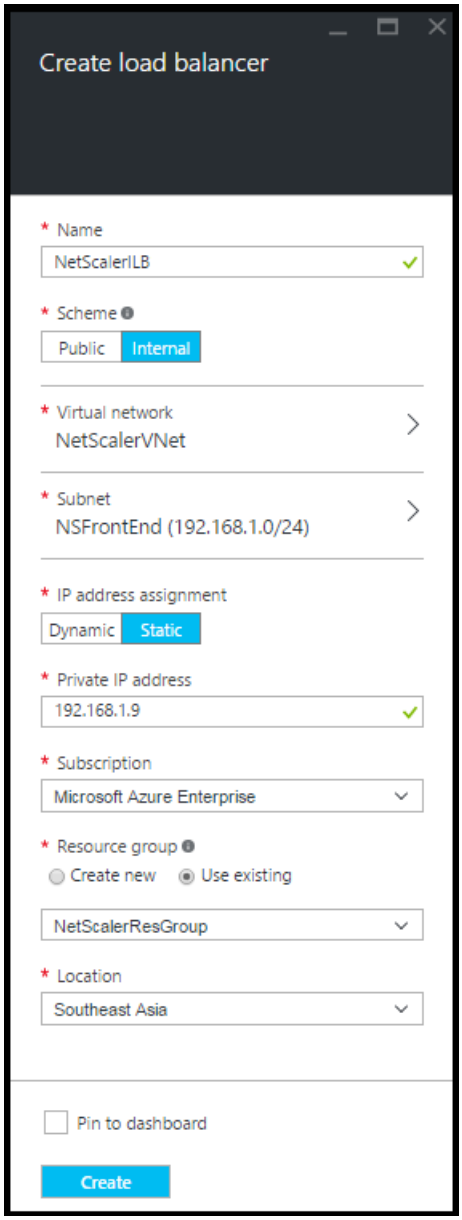
Configuring an Azure Internal Load Balancer

1. Click **+New** > **Networking** > **Load Balancer**.
2. On the **Create load balancer** pane, enter the following details:
   - Name of the load balancer
   - Scheme - select **Internal** to configure an internal load balancer
   - Virtual network - select the newly created virtual network from the drop-down list
   - Subnet - select the associated subnet
   - IP address assignment - select **Static**
   - Private IP address - assign a private IP address for the internal load balancer
   - Resource group - select the newly created resource group from the drop-down list
3. Click **Create**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



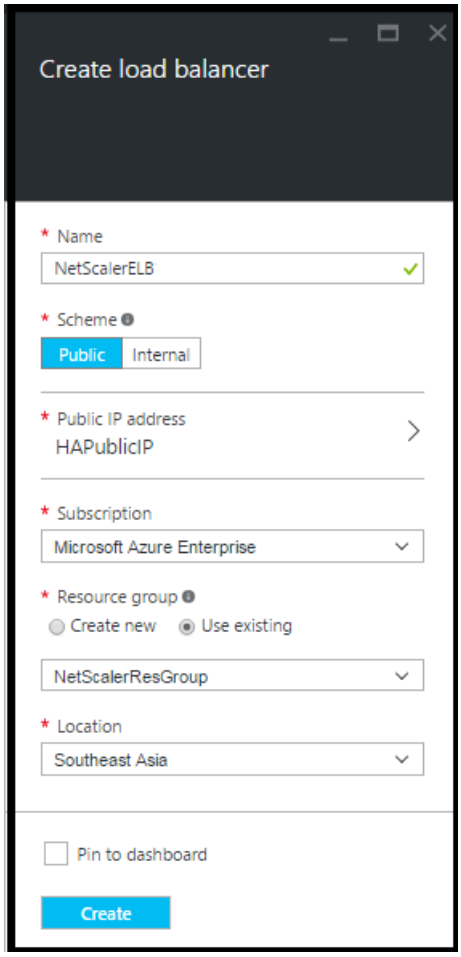
© 1999-2017 Citrix Systems, Inc. All rights reserved.



Configuring an External Load Balancer

1. To create an external load balancer, follow similar steps as creating an internal load balancer with the following differences:
   - Schema - select **Public**
   - Public IP address - assign a public IP address to the external load balancer
2. Click **Create**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

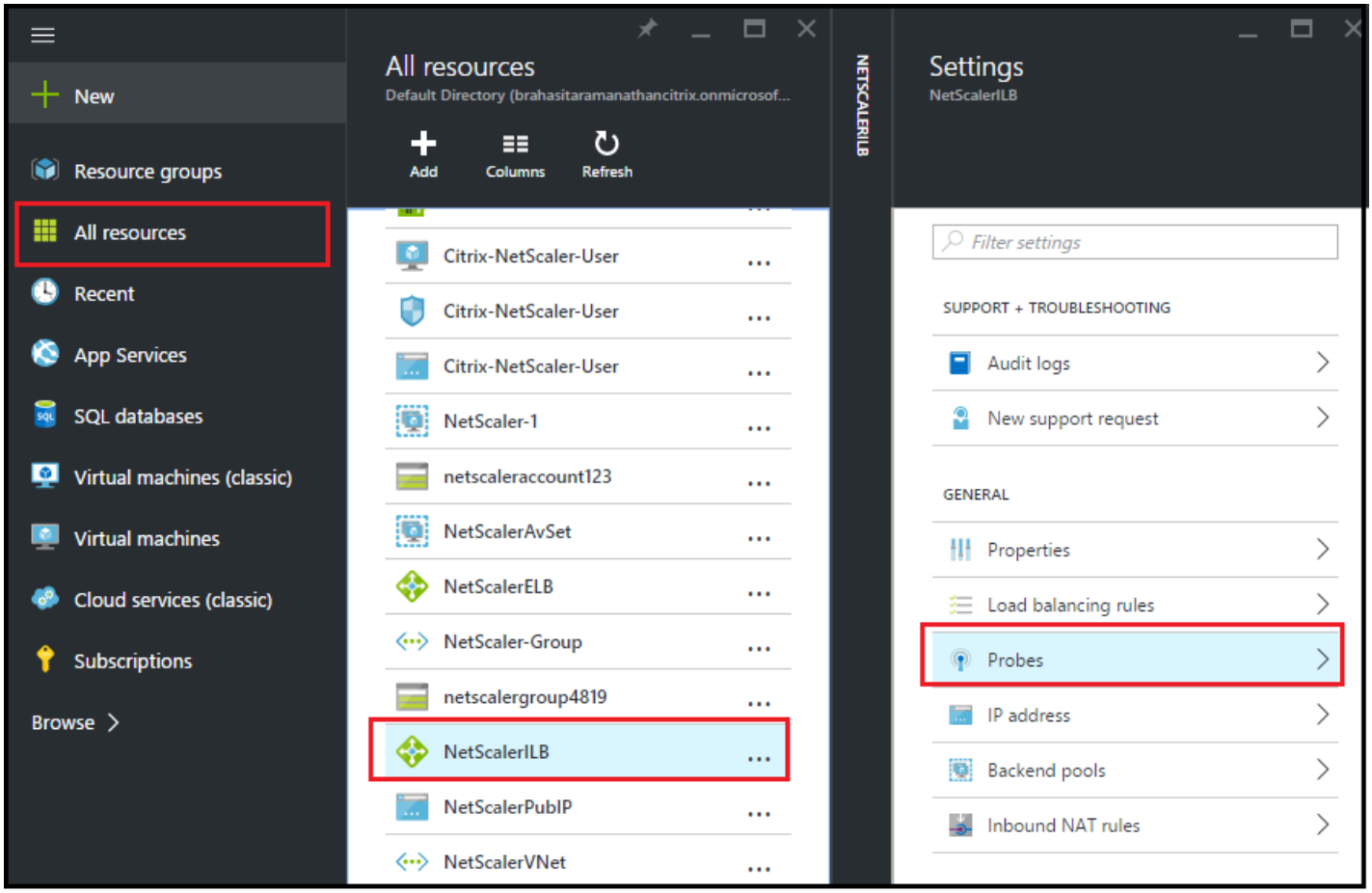


Configuring a Health Probe on a Load Balancer

Create custom TCP or HTTP probes to monitor the health of the various server instances. When the virtual machine fails to respond to the probe for three consecutive times, the Azure load balancer will not send the traffic to the nonresponsive virtual machine.

1. Click **All resources,** and search for the load balancer that you created in the previous task by typing the name in the search box.
2. In the **Settings** pane, click **Probes**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

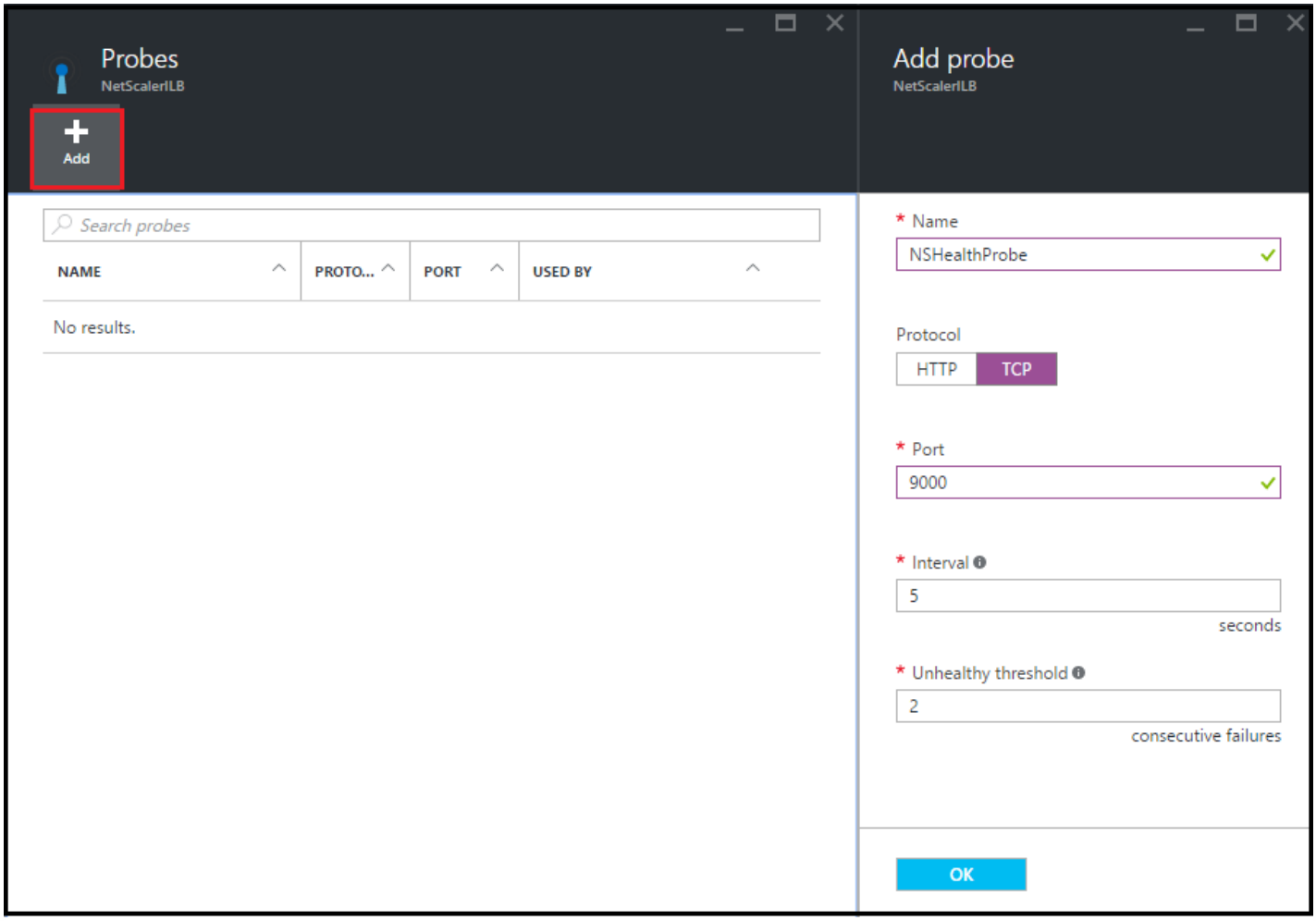


3. Click **+Add** and in the Add probe pane, enter the following details:

- Name of the health probe
- Protocol - select **TCP**
- Port - type 9000
- Set the Interval and Unhealthy threshold limits

4. Click **OK**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

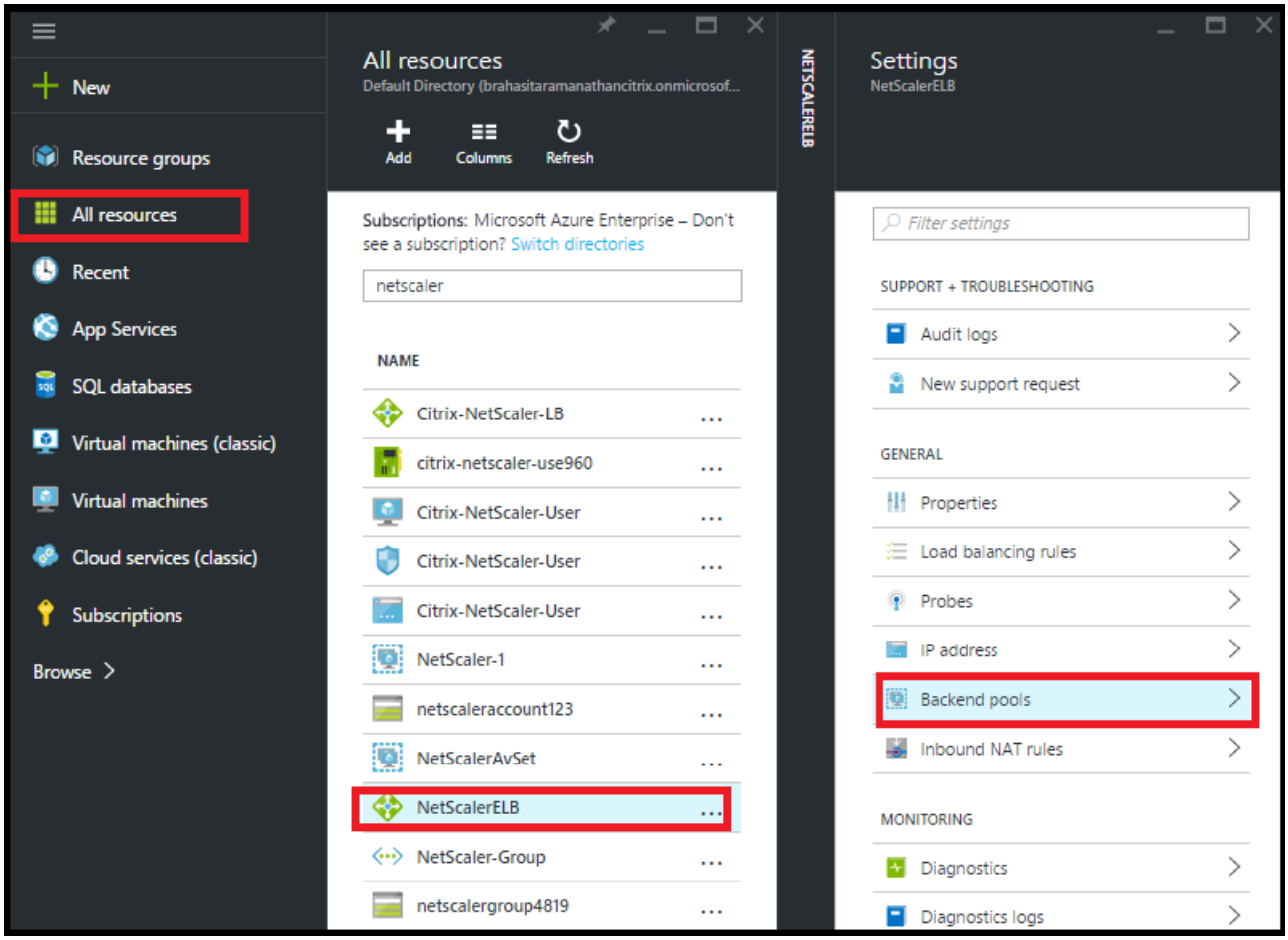


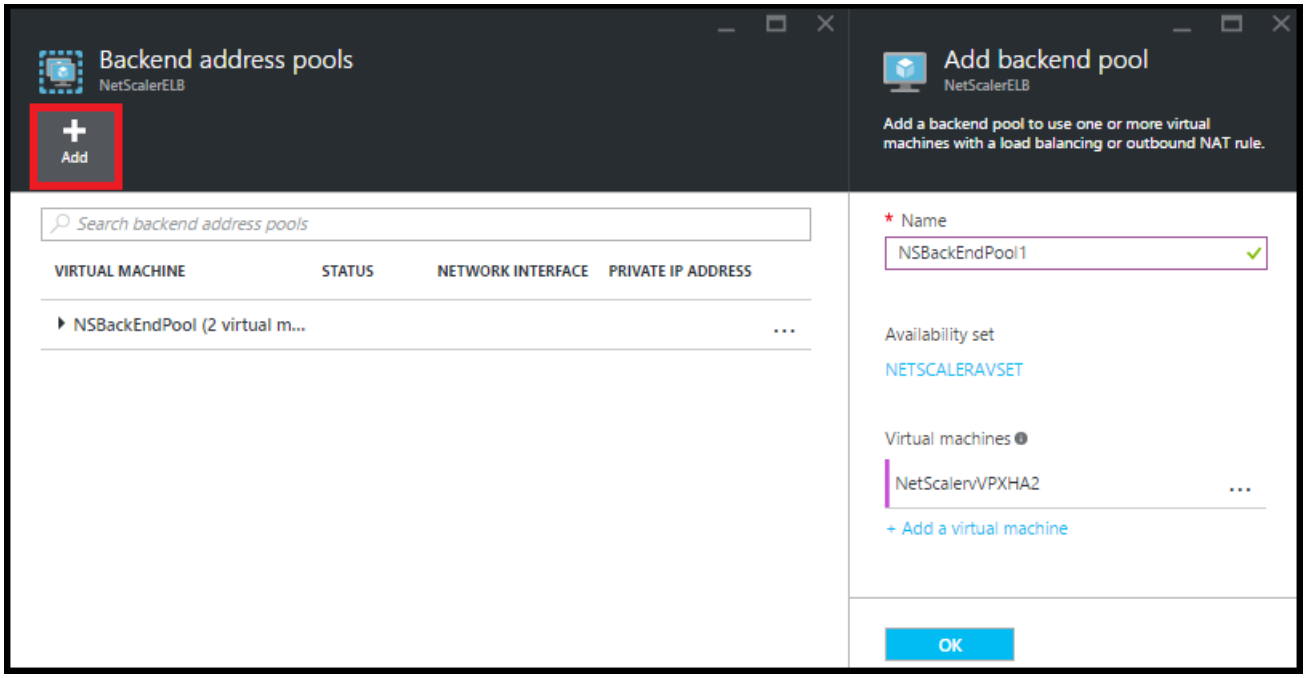
Configuring a Backend Pool on a Load Balancer

Create backend pools, that is, a pool of IP addresses associated with the virtual machine Network Interface Cards (NIC) to which the load is distributed.

1. Click **All resources** and search for the load balancer that you have created by typing the name in the search box.
2. In the **Settings** pane, select **Backend pools**.
3. Click **+Add,** and in the **Add backend pool** pane, enter the following details:
   - Name of the backend pool
   - Availability set - select the availability set created earlier
   - Virtual machines - select the NetScaler VPX instances that are in high availability deployment. Press <Ctrl> to select multiple instances.
4. Click **OK**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.





Configuring a NAT Rule on a Load Balancer

Create custom NAT rules on the Azure load balancer to define the inbound traffic flowing through the front-end IP address

© 1999-2017 Citrix Systems, Inc. All rights reserved.

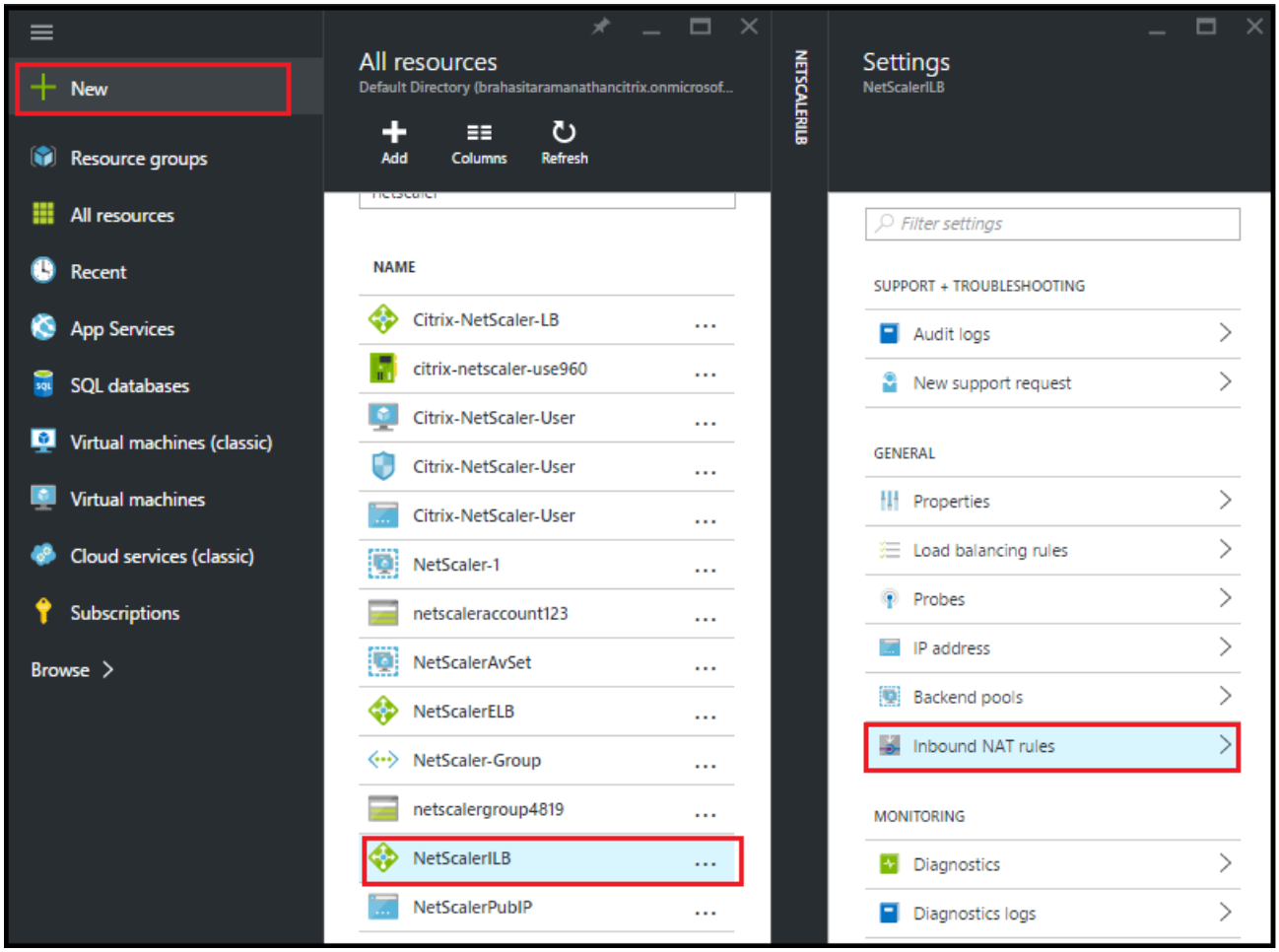
and distributed to the back-end IP address. Make sure that no two NAT rules has the combination of same service, same source port, and same target port.

> **Note**
>
> A front-end IP address is the external IP address on the load balancer that faces the incoming traffic, and a back-end IP address is the virtual machine facing IP address that receives the traffic from the load balancer.

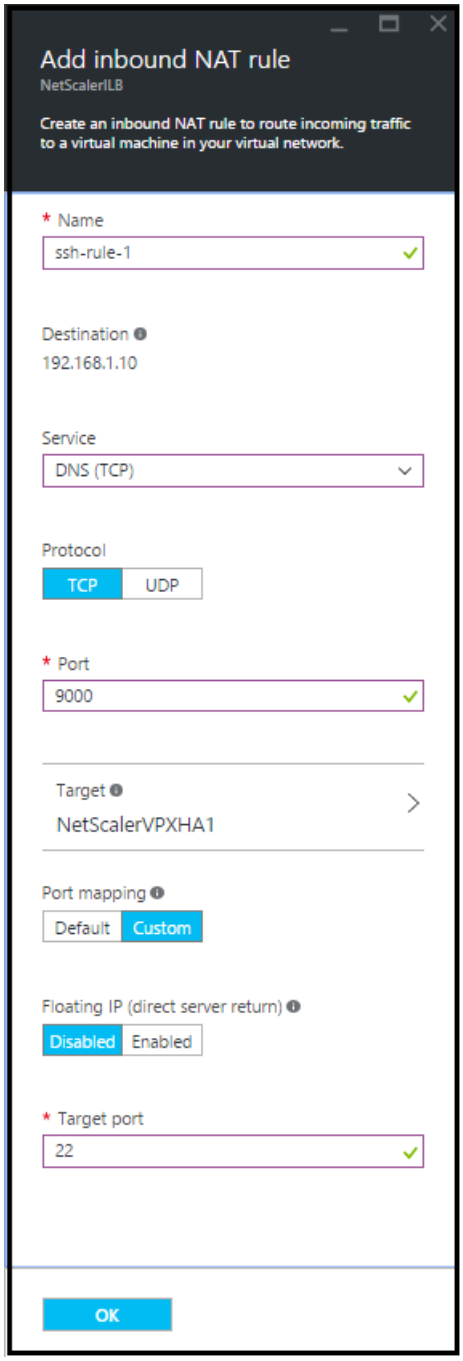
1. Click **All resources,** and search for the load balancer that you have created by typing the name in the search box.
2. In the **Settings** pane, select **Inbound NAT rules**.
3. Click **+Add,** and in the **Add inbound NAT rule** pane, add a NAT rule for each type of request. You can add multiple NAT rules.
4. Enter the following details, and then click **OK**.
   - Name of the rule
   - Service - select the required service from the drop-down list
   - Port - type the correct port number
   - Target - select the NetScaler VPX that will be the target of this rule
   - Target port - the target port is automatically populated depending on the service selected

> **Note**
>
> Citrix recommends TCP services for the NetScaler VPX VM on port 9000.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.



Configuring a Load Balancing Rule on a Load Balancer

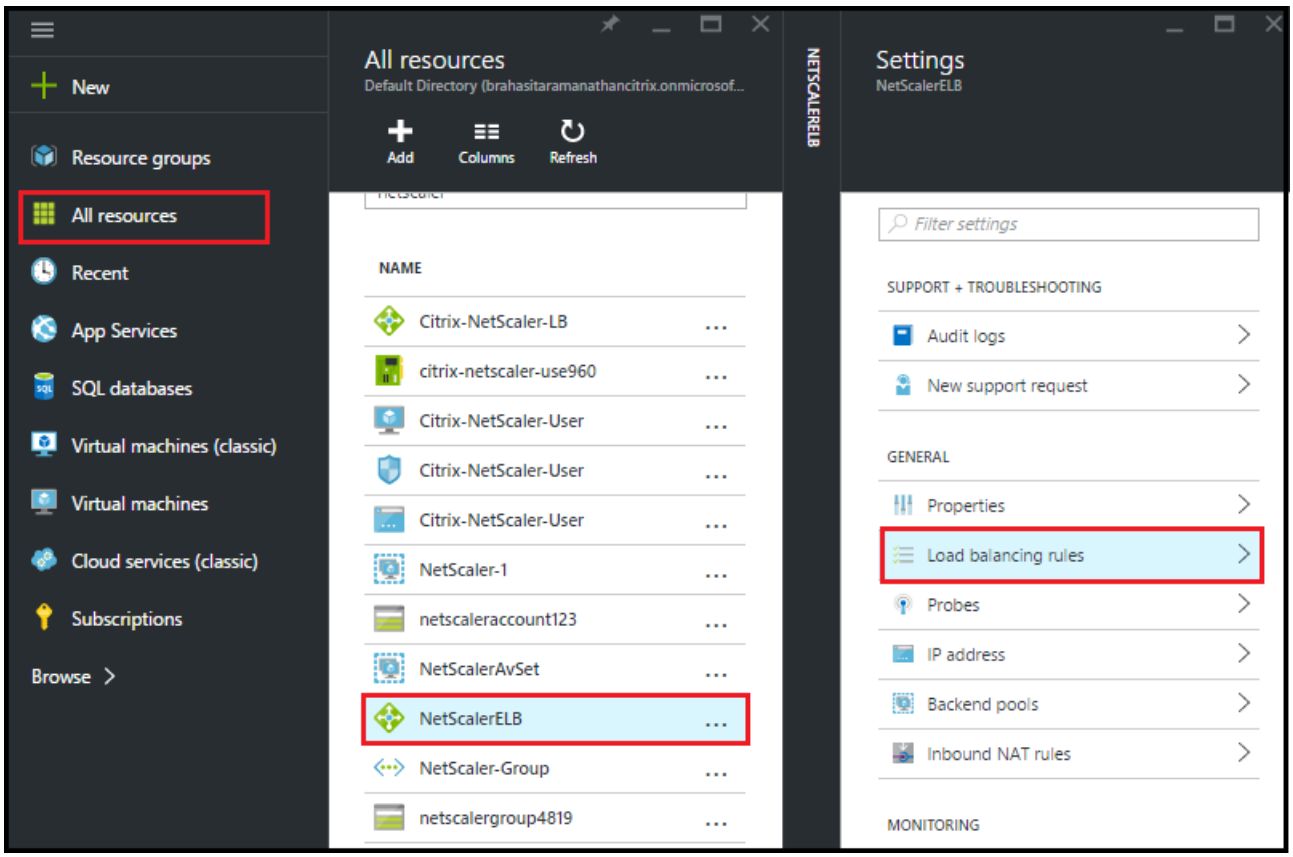
By creating a load balancer rule on Azure load balancer, you can define a combination of a front-end IP address and port, and back-end IP address and port associated with virtual machines.

For example, create a rule so that all HTTP requests coming on the public IP will be forwarded to the availability set on their port 80.

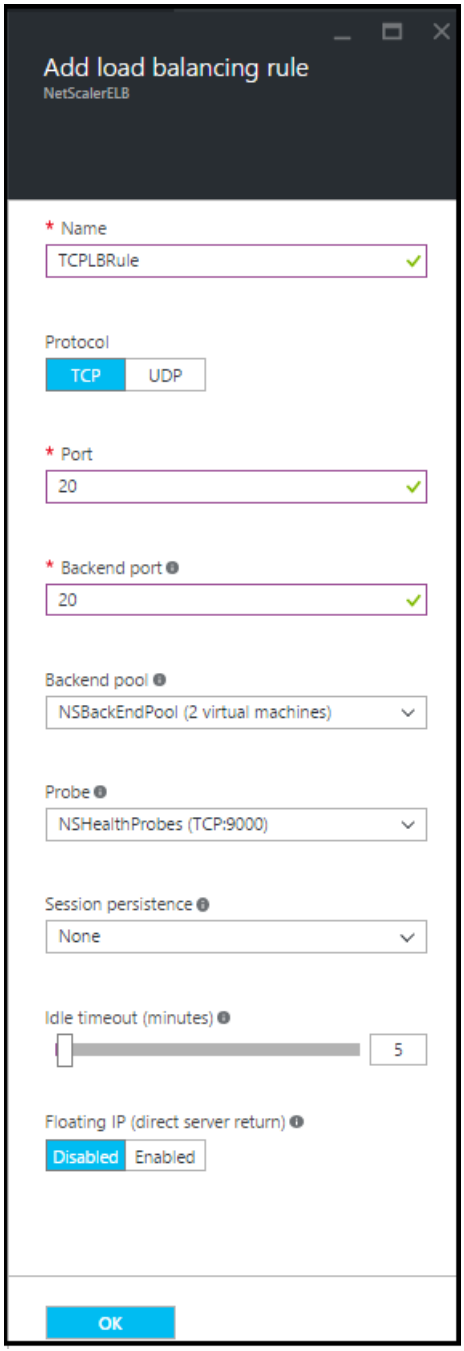
1. Click **All resources,** and search for the Azure load balancer that you have created by typing the name in the search box.
2. In the **Settings** pane, select **Load balancing rules**.
3. Click **+Add,** and in the **Add load balancing rules** pane, create load balancing rules for each type of incoming network traffic.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Enter the following details:
   - Name of the rule
   - Protocol - select the protocol
   - Port - type the port number based on the port selected
   - Backend pool - select the back-end pool from the drop-down list
   - Probe - select the health probe from the drop-down list
5. Click **OK**.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

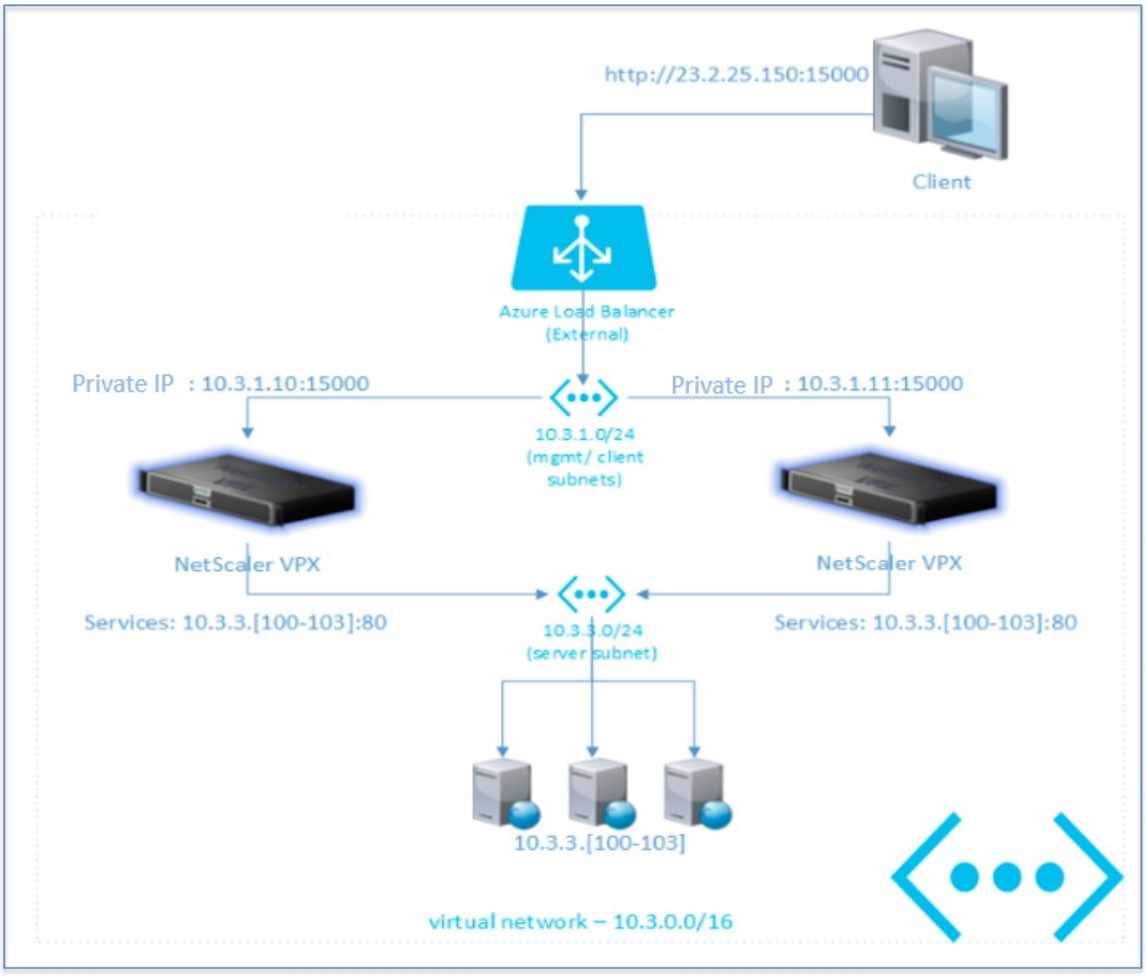


Configuring NetScaler High Availability with the Azure External Load Balancer

For client traffic originating from Internet, you have to deploy an external Azure load balancer to create a high availability configuration of NetScaler virtual machines in a load-balanced set.

The following figure shows how high availability is achieved in active-active mode by using the external Azure load balancer. The two NetScaler virtual machines are in a load-balanced set that accepts client traffic from the Internet over port 15000. The external Azure load balancer load balances these client requests between the two virtual machines.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.



Before you begin configuring the load-balanced set through the Azure portal, perform one of the following tasks:

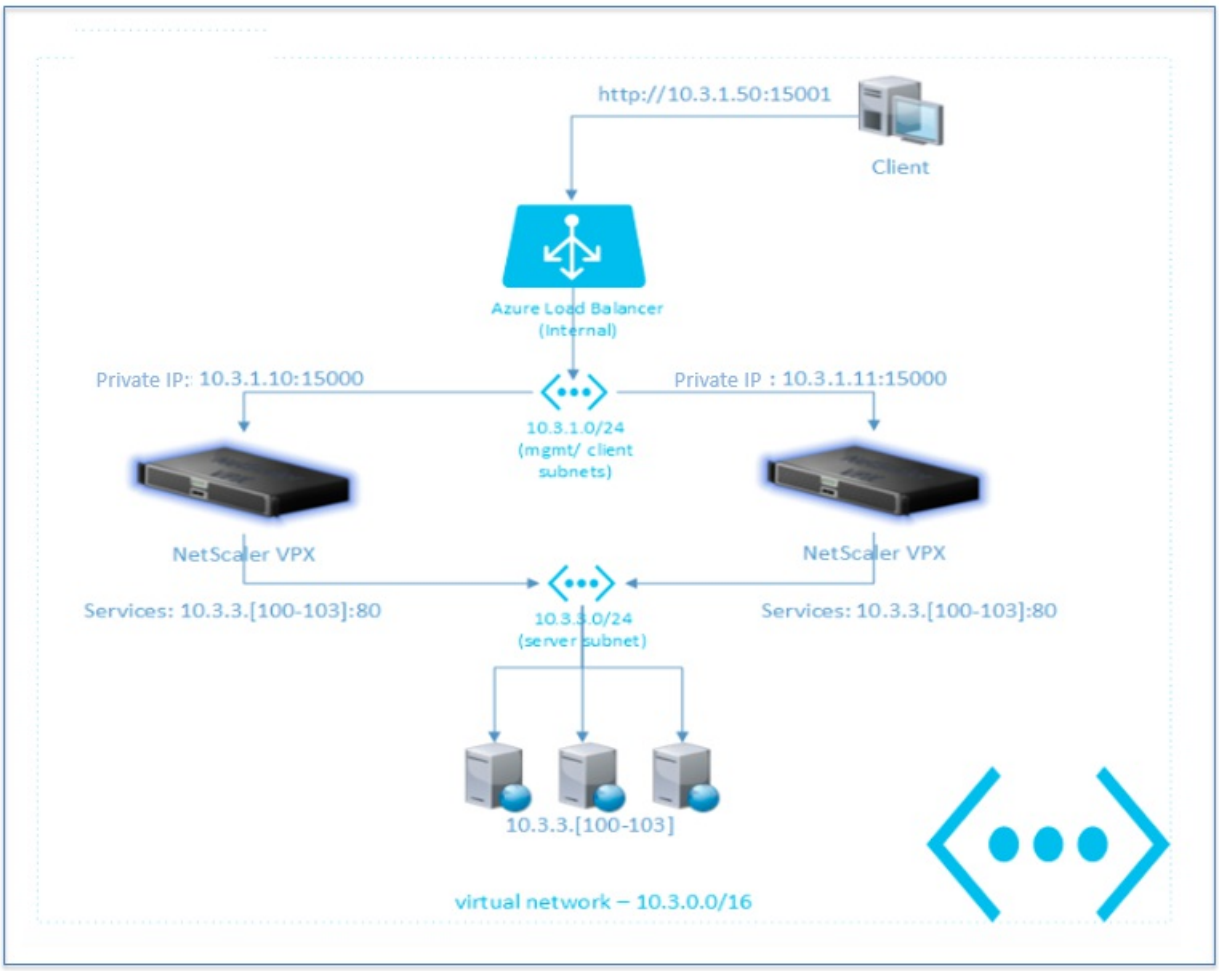
- For an active-passive deployment, configure NetScaler virtual machines as primary and secondary nodes by using the following command: **add ha node** <ID> <IP address>.
- For an active-active deployment, configure the required services on both NetScaler virtual machines.

Configuring NetScaler High Availability with the Azure Internal Load Balancer

If your client traffic originating from within the virtual network with a regional scope, deploy an internal Azure load balancer to achieve high availability of NetScaler virtual machines added to a load-balanced set.

The following figure shows how high availability is achieved in an active-active mode by using the internal Azure load balancer. The two NetScaler virtual machines are in a load-balanced set that accepts client traffic from the Internet at port 15001. The internal Azure load balancer load balances these client requests between the two virtual machines.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Before you begin configuring the load-balanced set by using Azure PowerShell, perform one of the following tasks:

- For an active-passive deployment, configure NetScaler virtual machines as primary and secondary nodes by using the following command: **add ha node** <ID> <IP address>.
- For an active-active deployment, configure the required services on both NetScaler virtual machines.

You can configure a load-balanced set by using PowerShell commands alone.

Accessing the NetScaler VPX Virtual Machine

You can access the NetScaler instance either through its graphical user interface (GUI) or through the command line interface (CLI). You can use the PIP to access the NetScaler virtual machine instance.

To log on to the virtual machine, use your username and password specified while creating the virtual machine.

You can change the password after you log on to the instance.

To access the NetScaler instance through the GUI

Open a browser and in the address field, type the virtual network public IP address provided by Azure during virtual machine provisioning, or type the PIP address.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Make sure you have created NSG inbound or outbound rules to allow access to the private port 80 or 443 when accessing the GUI by using the virtual network IP.

To access the NetScaler instance through command line

Use any command line access tool (for example, Putty) to access NetScaler instance through the command line. Specify either the virtual network public IP address provided by Azure during NetScaler VPX provisioning, or specify the PIP address. Use SSH protocol with port 22.

> **Note**
>
> Make sure that you have created NSG inbound or outbound rules to allow access to private port 22 when accessing the commnad line by using the virtual network IP.

For information about getting started with a NetScaler appliance, see http://docs.citrix.com/en-us/netscaler/11.html.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Multiple Azure VIPs for a NetScaler VPX in the Azure Resource Manager (ARM)

Jun 20, 2016

In Azure Resource Manager (ARM), configure multiple public virtual IP addresses (VIPs) for both single NetScaler VPX deployment, and also for high availability (HA) deployment.

Note: Multiple VIPs can be configured for external load balancers only.

For deploying NetScaler VPX instances in both single mode and high availability mode, you have to use PowerShell commands alone, as you can create multiple front-end IP addresses (FIP) through PowerShell only.

Deploying NetScaler VPX in Standalone Mode

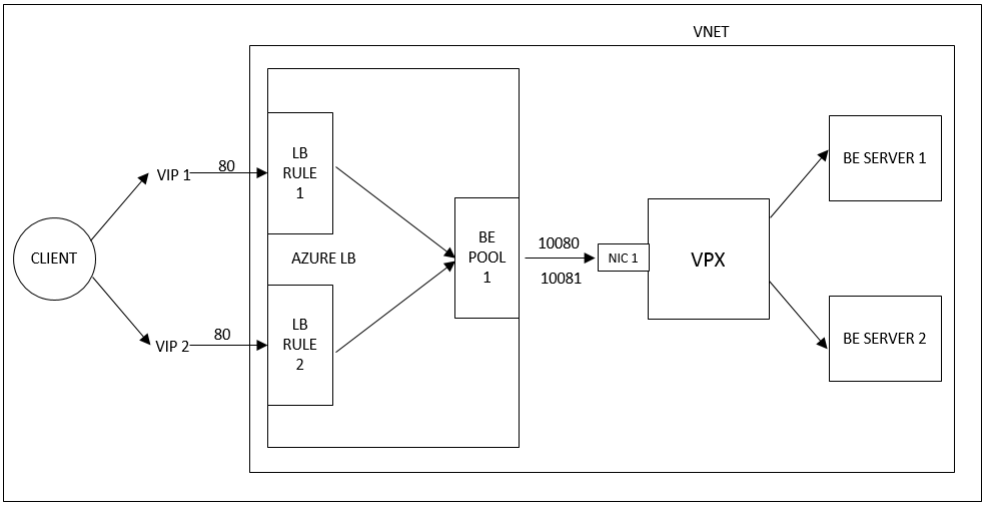
This section provides PowerShell commands to deploy a NetScaler VPX in a standalone mode with multiple front-end IPs mapped to a single back-end pool.

Configure multiple FIPs, back-end pools, LB rules, and inbound NAT rules as part of configuring Azure load balancer.

Make sure that the following conditions are met in a single NetScaler VPX deployment:

1. A back-end pool contains only one NetScaler VPX instance.
2. Two load balancer rules are defined and are mapped to the following two VIPs:
   1. VIP1:80 > Back-end Pool 1:10080
   2. VIP2:80 > Back-end Pool 1:10081
3. A load balancer rule is defined to map VIP1:10080 > Back-end Pool 1:80, to access NetScaler VPX user interface.
4. An inbound NAT rule is defined to map VIP1:22 > Back-end Pool 1:22 to access NetScaler VPX through SSH.

The following image illustrates how you can configure multiple cloud service IP addresses on Azure Resource Manager for NetScaler virtual servers.



Run the following PowerShell commands to deploy a NetScaler VPX instance in a standalone mode:

1. Create Resource Group

$rgName="<resource group name>"

$locName="<location name, such as West US>"

Command:

New-AzureRmResourceGroup -Name $rgName -Location $locName

For example:

$rgName = "ARM-LB-NS"

$locName = "East Asia"

New-AzureRmResourceGroup -Name $rgName -Location $locName

2. Create Storage Account

You must select a globally unique name for your storage account that contains only lowercase letters and numbers.

$saName="<storage account name>"

$saType="<storage account type, specify one: Standard_LRS, Standard_GRS, Standard_RAGRS, or Premium_LRS>"

Command:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName

For example:

$saName="vpxstorage"

$saType="Standard_LRS"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName

3. Create Availability Set

$avName="<availability set name>"

Command:

New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -Location $locName

For example:

$avName="avNSSet"

4. Create Virtual Network and Subnet

Add new virtual network with at least one subnet, if it is not created previously.

$vnetName = "LBVnet"

Commands:

Create subnets:

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix 10.0.1.0/24

(Configure this parameter value as per your requirement)

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix 10.0.2.0/24

Create Virtual Network:

New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName $rgName -Location $locName -AddressPrefix 10.0.0.0/16 -Subnet $frontendSubnet,$backendSubnet

5. Create Public IP Address

The number of public IPs created should be equal to number of External VIPs required.

- Before using, check for the availability of the value for DomainNameLabel.
- Create two VIPs.

Commands:

$pubName1 ="PublicIp1"

$dnsName1="nsvpx1"

$pubName2 ="PublicIp2"

$dnsName2="nsvpx2"

$publicIP1 = New-AzureRmPublicIpAddress -Name $pubName1 -ResourceGroupName $rgName -Location $locName -AllocationMethod Static -DomainNameLabel $dnsName1

$publicIP2 = New-AzureRmPublicIpAddress -Name $pubName2 -ResourceGroupName $rgName -Location $locName -AllocationMethod Static -DomainNameLabel $dnsName2

6. Create Front-end IP for Specified Public IP Addresses

$FIPName1 ="VIP1"

$FIPName2="VIP2"

Commands:

$frontendIP1 = New-AzureRmLoadBalancerFrontendIpConfig -Name $FIPName1 -PublicIpAddress $publicIP1

$frontendIP2 = New-AzureRmLoadBalancerFrontendIpConfig -Name $FIPName2 -PublicIpAddress $publicIP2

7. Create Back-end Pool

$BEPool1 = "backend-Pool1"

Command:

$beaddresspool1= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name $BEPool1

8. Create Health Probe

Create TCP health probe with port 9000 and interval 5 seconds.

Command:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name HealthProbe -Protocol Tcp -Port 9000 -IntervalInSeconds 5 -ProbeCount 2

9. Create Load Balancer Rule

For each frond end IP and service, we need to create lbRule.

Here back-end address pool can contain set of virtual machines. For single VPX deployment only single VPX instance will be part of this pool.

Note: Combined values for front-end IP configuration, back-end address pool, front-end port, back-end port parameters should not be same for any two rules.

For example, every FIP/VIP access, and HTTP service uses front-end port 80. As back-end pool is same, back-end port needs to be used differently for each load balancer rule.

Commands:

$lbrule1 = New-AzureRmLoadBalancerRuleConfig -Name "HTTP1" -FrontendIpConfiguration $frontendIP1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort 80 -BackendPort 10080

$lbrule2 = New-AzureRmLoadBalancerRuleConfig -Name "HTTP2" -FrontendIpConfiguration $frontendIP2 -BackendAddressPool $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort 80 -BackendPort 10081

LB rule to access http service of NS can be added in the following way:

Command:

$lbrule3 = New-AzureRmLoadBalancerRuleConfig -Name "HTTPNS" -FrontendIpConfiguration $frontendIP1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort 10080 -BackendPort 80

10. Create Inbound NAT Rules

Create NAT rules for services that does not require to be load balanced.

For example, create an ssh access to VPX instance.

Protocol - FrontEndPort - BackendPort triplet should not be same for two NAT rules belonging to same front-end IP.

Command:

$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -Name SSH1 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -FrontendPort 22 -BackendPort 22

11. Create Load Balancer

Create load balancer with all of the above defined rules, front-end IPs, and a back-end pool.

Command:

$lbName = "NSALB"

$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name $lbName -Location $locName -InboundNatRule $inboundNATRule1 -FrontendIpConfiguration $frontendIP1, $frontendIP2 -LoadBalancingRule $lbrule1, $lbrule2, $lbrule3 -BackendAddressPool $beAddressPool1 -Probe $healthProbe

12. Create NIC

Create a NIC and associate it with the NetScaler VPX instance.

Commands:

$nicName="NIC1"

$lbName="NSALB"

$bePoolIndex=0

$natRuleIndex=0

$subnetIndex=0 ß Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName

$nic1=New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -Subnet $vnet.Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule $lb.InboundNatRules[$natRuleIndex]

13. Create NetScaler VPX Instance

Create NetScaler VPX instance from MarketPlace image and attach the NIC to the virtual instance.

Commands:

$vmName="VPX1"

$vmSize="Standard_A3" / "Standard_DS4"

$pubName="citrix"

$skuName = "netscalerbyol"

$offerName="netscalervpx110-6531"

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message "Type Credentials which will be used to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $nic1.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/" + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm1
```

The above commands creates a NetScaler VPX instance, then add virtual servers to the NetScaler VPX instance for the specified front end services.

Deploying NetScaler VPX in High Availability Mode

This section provides PowerShell commands to deploy a NetScaler VPX in HA deployment with multiple front-end IPs mapped to a single back-end pool.

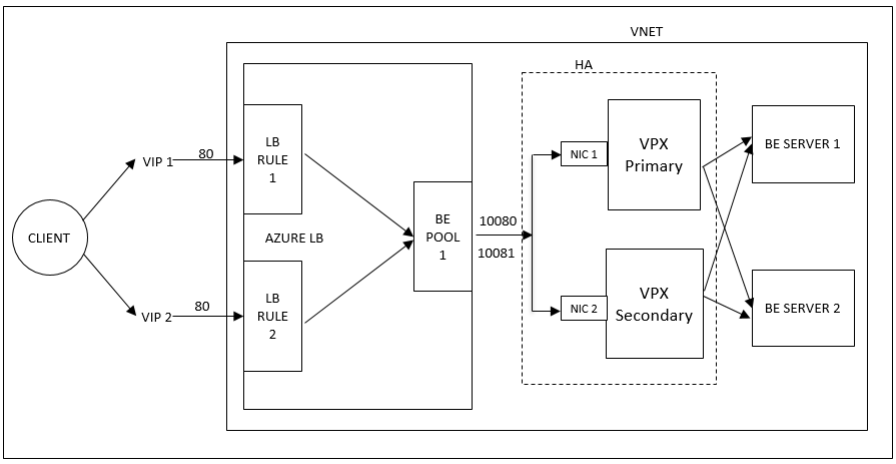
Configure multiple FIPs, backend pools, load balance rules, and inbound NAT rules as part of the Azure load balancer.

Make sure that the following conditions are met in HA deployment of NetScaler VPX instances:

1. A back-end pool contains two NetScaler VPX instances, which are part of HA.
2. Two load balancer rules are defined and are mapped to following two VIPs:
   1. VIP1:80 > Back-end Pool 1:10080
   2. VIP2:80 > Back-end Pool 1:10081
3. A load balancer rule is defined, which maps VIP1:10080 > Back-end Pool 1:80, to access NetScaler VPX GUI.
4. Two inbound NAT rules are defined to map the following two VIPS:
   1. VIP1:22 > Back-end Pool 1:22 to access NetScaler VPX Primary
   2. VIP1:10022 > Back-end Pool-1:22 to access NetScaler VPX Secondary through SSH

All services that are defined as part of Azure load balancer rules will get load balanced. That is, if the primary VPX fails, the secondary VPX will take care of all of the services in Active-Passive HA deployment.

The following image illustrates how you can configure multiple cloud service IP addresses on Azure Resource Manager for NetScaler virtual servers in HA mode.



1. Create Resource Group

```
$rgName="<resource group name>"

$locName="<location name, such as West US>"
```

Commands:

```
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

For example:

```
$rgName = "ARM-Mult-VIP-HA"

$locName = "East Asia"
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

New-AzureRmResourceGroup -Name $rgName -Location $locName

2. Create Storage Account

You must select a globally unique name for your storage account that contains only lowercase letters and numbers.

$saName="<storage account name>"

$saType="<storage account type, specify one: Standard_LRS, Standard_GRS, Standard_RAGRS, or Premium_LRS>"

Commands:

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName

For example:

$saName="vpxstorage1"

$saType="Standard_LRS"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName

3. Create Availability Set

$avName="<availability set name>"

Command:

New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -Location $locName

For example:

$avName="avNSSetARM"

4. Create Virtual Network and Subnet

Add new virtual network with at least one subnet if it is not created previously.

$vnetName = "LBVnet"

Commands:

Create subnets:

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix 10.0.7.0/24 ß (this parameter value should be as per your requirement)

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix 10.0.8.0/24

Create Virtual Network:

New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName $rgName -Location $locName -AddressPrefix 10.0.0.0/16 -Subnet $frontendSubnet,$backendSubnet

5. Create Public IP Address

The number of public IP addresses created should be equal to the number of external VIPs required.

- Before using, check for the availability of the value for Domain.Name.Label.
- Create two VIPs.

Commands:

$pubName1 ="PublicIp1"

$dnsName1="nsvpx1"

$pubName2 =”PublicIp2"

$dnsName2="nsvpx2"

$publicIP1 = New-AzureRmPublicIpAddress -Name $pubName1 -ResourceGroupName $rgName -Location $locName -AllocationMethod Static -DomainNameLabel $dnsName1

$publicIP2 = New-AzureRmPublicIpAddress -Name $pubName2 -ResourceGroupName $rgName -Location $locName -AllocationMethod Static -DomainNameLabel $dnsName2

6. Create Front-end IP Addresses

$FIPName1 = "VIP1"

$FIPName2="VIP2"

Commands:

$frontendIP1 = New-AzureRmLoadBalancerFrontendIpConfig -Name $FIPName1 -PublicIpAddress $publicIP1

$frontendIP2 = New-AzureRmLoadBalancerFrontendIpConfig -Name $FIPName2 -PublicIpAddress $publicIP2

7. Create Back-end Pool

© 1999-2017 Citrix Systems, Inc. All rights reserved.

$BEPool1 = "backend-Pool1"

Command:

$beaddresspool1= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name $BEPool1

8. Create Health Probe

Create TCP health probe with port 9000 and interval 5 seconds.

Command:

$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name HealthProbe -Protocol Tcp -Port 9000 -IntervalInSeconds 5 -ProbeCount 2

9. Create Load Balancer Rules

For each frond end IP and service, you have to create a separate load balancer rule.

Back-end address pool can contain set of virtual machines. For a single NetScaler VPX deployment, only single NetScaler VPX instance will be part of this pool.

Note: Combined values for front-end IP configuration, back-end address pool, front-end port, back-end port parameters should not be the same for any two rules.

Examples:

Each FIP/VIP access and HTTP service uses front-end port 80. As back-end pool is same, back-end port has to be used differently for each load balancer rule.

Command:

$lbrule1 = New-AzureRmLoadBalancerRuleConfig -Name "HTTP1" -FrontendIpConfiguration $frontendIP1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort 80 -BackendPort 10080

$lbrule2 = New-AzureRmLoadBalancerRuleConfig -Name "HTTP2" -FrontendIpConfiguration $frontendIP2 -BackendAddressPool $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort 80 -BackendPort 10081

Load balancer configuration rule to access the HTTP service of a netScaler can be added in the following way:

Command:

$lbrule3 = New-AzureRmLoadBalancerRuleConfig -Name "HTTPNS" -FrontendIpConfiguration $frontendIP1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort 10080 -BackendPort 80

10. Create Inbound NAT Rules

Create NAT rules for services that does not require to be load balanced.

For example, create an ssh access to VPX instance.

Note: Protocol - Front-end Port - Back-end Port triplet should not be the same for two NAT rules belonging to the same front-end IP address.

Commands:

$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -Name SSH1 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -FrontendPort 22 -BackendPort 22

$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -Name SSH2 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -FrontendPort 10022 -BackendPort 22

11. Create Load Balancer

Create load balancer with all of the above defined rules, front-end IPs, and a back-end pool.

Command:

$lbName ="NSALB"

$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name $lbName -Location $locName -InboundNatRule $inboundNATRule1, $inboundNATRule2 -FrontendIpConfiguration $frontendIP1, $frontendIP2 -LoadBalancingRule $lbrule1, $lbrule2, $lbrule3 -BackendAddressPool $beAddressPool1 -Probe $healthProbe

12. Create NIC

Create a NIC and associate it with the NetScaler VPX instance.

Commands:

$nicName="NIC1"

$lbName="NSALB"

$bePoolIndex=0

$natRuleIndex=0

$subnetIndex=0 ß Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName

$nic1=New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -Subnet $vnet.Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule $lb.InboundNatRules[$natRuleIndex]

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
$nicName="NIC2”

$lbName="NSALB"

$bePoolIndex=0

$natRuleIndex=1ß 2nd SSH rule

$subnetIndex=0

$nic2=New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -Subnet $vnet.Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule $lb.InboundNatRules[$natRuleIndex]
```

13. Create NetScaler VPX Instances

Create a NetScaler VPX instance from MarketPlace image and attach a NIC to it.

Commands:

```
$vmName="VPX1"

$vmSize="Standard_A3"

$pubName="citrix"

$skuName = "netscalerbyol"

$offerName="netscalervpx110-6531"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message "Type Credentials which will be used to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $nic1.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/" + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm1

$vmName="VPX2"

$vmSize="Standard_A3"

$pubName="citrix"

$skuName = "netscalerbyol"

$offerName="netscalervpx110-6531"

$avSet=Get-AzureRmAvailabilitySet – Name $avName – ResourceGroupName $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message "Type Credentials which will be used to login to VPX instance"

$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $nic2.Id

$diskName="dynamic”

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri2=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/" + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri $osDiskUri2 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm2
```

14. Create High Availability

© 1999-2017 Citrix Systems, Inc. All rights reserved.

When both NetScaler VPX instances are running, connect to both VPX instances through SSH to configure the virtual machines.

1. To configure Active-Passive HA, run "add HA node #nodeID" command on both nodes, and then run configuration commands on Primary VPX instance.
2. To configure Active-Active HA, run same set of configuration commands on both nodes.

Azure ARM Components

This table lists those of the Azure Resource Manager (ARM) components that can be created using PowerShell, and those that can be created using the Azure Resource Manager portal.

	PowerShell	ARM Portal
Resource Group	Yes	Yes
Storage Account	Yes	Yes
Availability Set	Yes	Yes
Virtual Network and Subnet	Yes	Yes
Public IP	Yes	Yes
Multiple Frontend IP	Yes	No
Backend Pool	Yes	Yes
Health Probes	Yes	Yes
LB Rules with each rule using only one front end IP	Yes	Yes
LB Rules with each rule using different front end IP	Yes	No
Inbound NAT Rules with same front end IP for all	Yes	Yes
Inbound NAT rules with different front end IP	Yes	No
External Load Balancer	Yes	Yes
Internal Load Balancer	Yes	Yes
Load balancer with front end IP	Yes	Yes
Load balancer with multiple front end IP	Yes	No
Network Security Group (NSG)	Yes	Yes
Network Interface Card (NIC)	Yes	Yes
Virtual Machine	Yes	Yes

Yes

Yes

Yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Azure Resource Manager (ARM) Terminology

Jun 20, 2016

Terms in Azure Resource Manager

Some of the new terms used in Azure Resource Manager are listed below. This list helps you relate to the terms and concepts used in Azure Service Management or classical mode of deployment.

1. Azure Load Balancer – Azure load balancer is a resource that distributes incoming traffic among computers in a network. Traffic is distributed among virtual machines defined in a load-balancer set. A load balancer can be external or internet-facing, or it can be internal.

2. Azure Resource Manager (ARM) – ARM is the new management framework for services in Azure. Azure Load Balancer is managed using ARM-based APIs and tools.

3. Back-End Address Pool – These are IP addresses associated with the virtual machine Network Interface Card (NIC) to which load will be distributed.

4. BLOB - Binary Large Object – Any binary object like a file or an image that can be stored in Azure storage.

5. Front-End IP Configuration – An Azure Load balancer can include one or more front-end IP addresses, also known as a virtual IPs (VIPs). These IP addresses serve as ingress for the traffic.

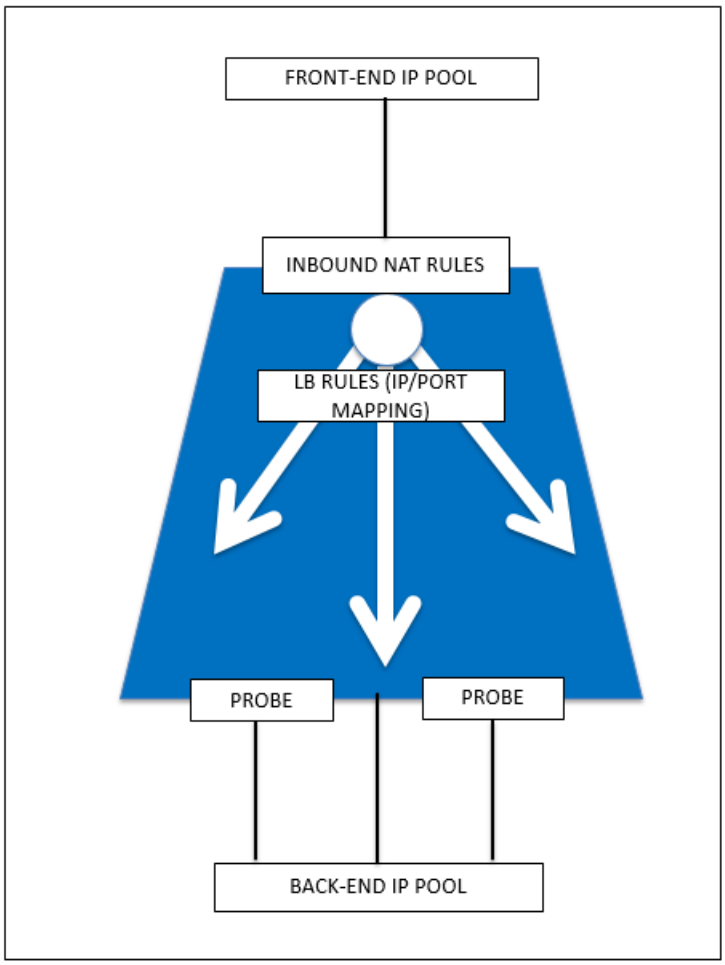
6. Instance Level Public IP (ILPIP) – An ILPIP is a public IP address that you can assign directly to your virtual machine or role instance, rather than to the cloud service that your virtual machine or role instance resides in. This does not take the place of the VIP (virtual IP) that is assigned to your cloud service. Rather, it's an additional IP address that you can use to connect directly to your virtual machine or role instance.

   Note: In the past, an ILPIP was referred to as a PIP, which stands for public IP.

7. Inbound NAT Rules – This contains rules mapping a public port on the load balancer to a port for a specific virtual machine in the back end address pool.

8. Load Balancing Rules – A rule property that maps a given front-end IP and port combination to a set of back-end IP addresses and port combination. With a single definition of a load balancer resource, you can define multiple load balancing rules, each rule reflecting a combination of a front end IP and port and back end IP and port associated with virtual machines.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



9. Network Security Group (NSG) – NSG contains a list of Access Control List (ACL) rules that allow or deny network traffic to your virtual machine instances in a virtual network. NSGs can be associated with either subnets or individual virtual machine instances within that subnet. When a NSG is associated with a subnet, the ACL rules apply to all the virtual machine instances in that subnet. In addition, traffic to an individual virtual machine can be restricted further by associating a NSG directly to that virtual machine.

10. Private IP addresses – Used for communication within an Azure virtual network, and your on-premises network when you use a VPN gateway to extend your network to Azure. Private IP addresses allow Azure resources to communicate with other resources in a virtual network or an on-premises network through a VPN gateway or ExpressRoute circuit, without using an Internet-reachable IP address. In the Azure Resource Manager deployment model, a private IP address is associated with the following types of Azure resources – virtual machines, internal load balancers (ILBs), and application gateways.

11. Probes – This contains health probes used to check availability of virtual machines instances in the back end address pool. If a particular virtual machine does not respond to health probes for some time, then it is taken out of traffic serving. Probes enable you to keep track of the health of virtual instances. If a health probe fails, the virtual instance will be taken out of rotation automatically.

12. Public IP Addresses (PIP) – PIP is used for communication with the Internet, including Azure public-facing services and is associated with virtual machines, Internet-facing load balancers, VPN gateways, and application gateways.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

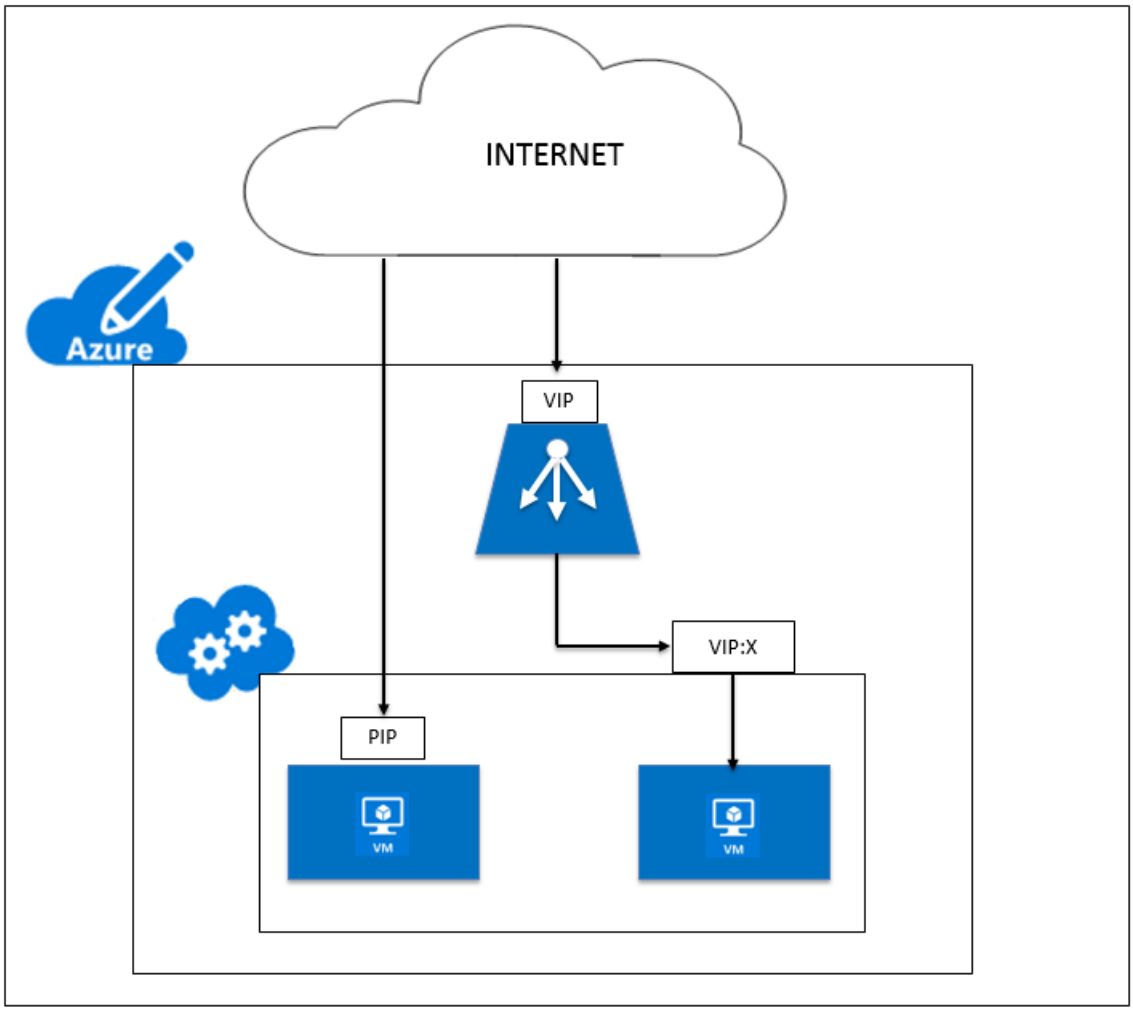
13. Region - An area within a geography that does not cross national borders and that contains one or more datacenters. Pricing, regional services, and offer types are exposed at the region level. A region is typically paired with another region, which can be up to several hundred miles away, to form a regional pair. Regional pairs can be used as a mechanism for disaster recovery and high availability scenarios. Also referred to generally as location.

14. Resource Group - A container in Resource Manager holds related resources for an application. The resource group can include all of the resources for an application, or only those resources that are logically grouped together

15. Storage Account – An Azure storage account gives you access to the Azure blob, queue, table, and file services in Azure Storage. Your storage account provides the unique namespace for your Azure storage data objects.

16. Virtual Machine – The software implementation of a physical computer that runs an operating system. Multiple virtual machines can run simultaneously on the same hardware. In Azure, virtual machines are available in a variety of sizes.

17. Virtual Network - An Azure virtual network is a representation of your own network in the cloud. It is a logical isolation of the Azure cloud dedicated to your subscription. You can fully control the IP address blocks, DNS settings, security policies, and route tables within this network. You can also further segment your VNet into subnets and launch Azure IaaS virtual machines and cloud services (PaaS role instances). Additionally, you can connect the virtual network to your on-premises network using one of the connectivity options available in Azure. In essence, you can expand your network to Azure, with complete control on IP address blocks with the benefit of enterprise scale Azure provides.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring NetScaler VPX in the Azure Service Management (ASM)

Mar 21, 2016

Provisioning a NetScaler VPX instance in the Azure cloud involves obtaining the NetScaler VPX image either from the Azure marketplace or from the Citrix download page, creating the virtual machine instance in Azure, and defining endpoints.

You can create a NetScaler VPX instance in Azure either by using the new Azure Preview portal or by using the classic Azure Management portal.

Before you start provisioning the VM, review the Before You Begin section.

This document includes the following information:

- Before You Begin
- Creating a NetScaler VPX Instance on Microsoft Azure by Using the Azure Preview Portal
- Creating a NetScaler VPX Instance on Microsoft Azure by Using the Classic Azure Management Portal
- Accessing the NetScaler VPX Virtual Machine

Before You Begin

Make sure that you have the following:

- A Microsoft Azure user account
- Access to Microsoft Azure Preview Management portal
- Access to the Microsoft Azure Classic Management portal
- Microsoft Azure SDK
- Microsoft Azure PowerShell

Before you begin creating the NetScaler VPX instance, make sure that you have created a virtual network with required subnets in which the VPX will reside. You can create virtual networks during VM provisioning but you will not have the flexibility to set different subnets. For information about creating virtual networks, see http://azure.microsoft.com/en-us/documentation/articles/create-virtual-network/

Optionally, configure DNS server and VPN connectivity that allows a virtual machine to access Internet resources.

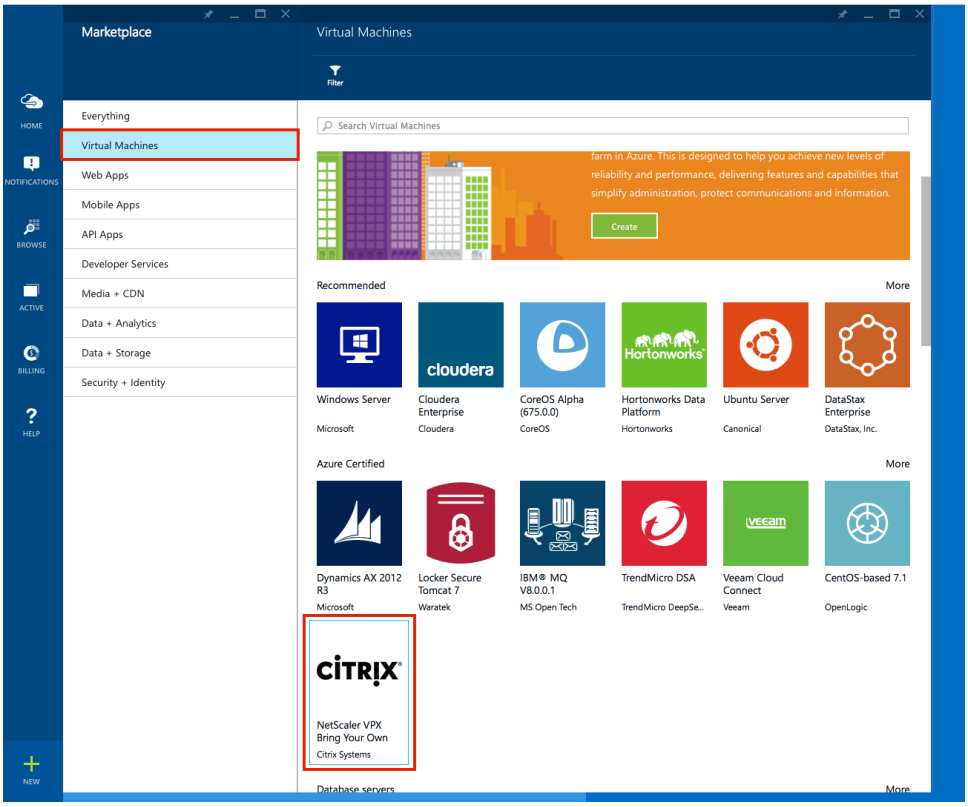
Creating a NetScaler VPX Instance on Microsoft Azure by Using the Azure Preview Portal

Obtain the NetScaler VPX image from the Azure MarketPlace, and then use the Azure MarketPlace Preview portal to create a NetScaler VPX instance.

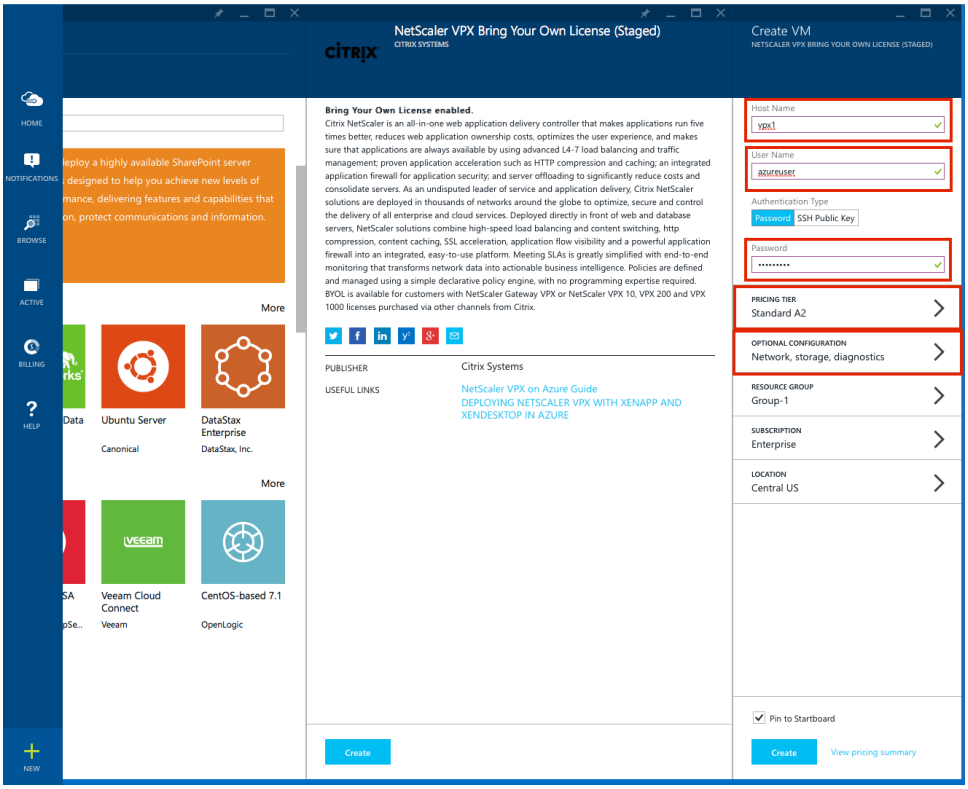
To create a NetScaler VPX instance on Microsoft Azure by using the Azure Marketplace Preview Portal

1. On the Microsoft Azure Portal page (https://portal.azure.com), click Marketplace.
2. On the Marketplace page, click Virtual Machine and select Citrix NetScaler VPX Bring Your Own.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



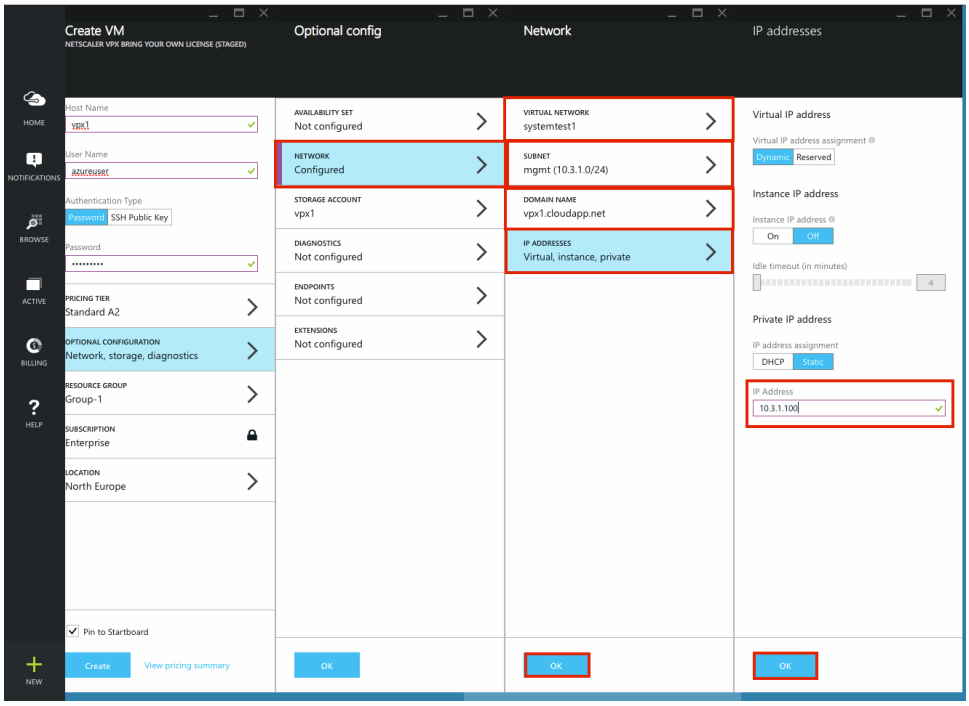
3. On the NetScaler VPX Bring Your Own License page, click Create.
4. On the Create VM page:
   1. Specify values for Host Name and User Name.
   2. Under Authentication Type, select SSH Public Key or Password.
   3. Click Pricing Tier and select the instance type you want to use.



4. In Optional Configuration, do the following:
   1. Configure Availability Set as required.

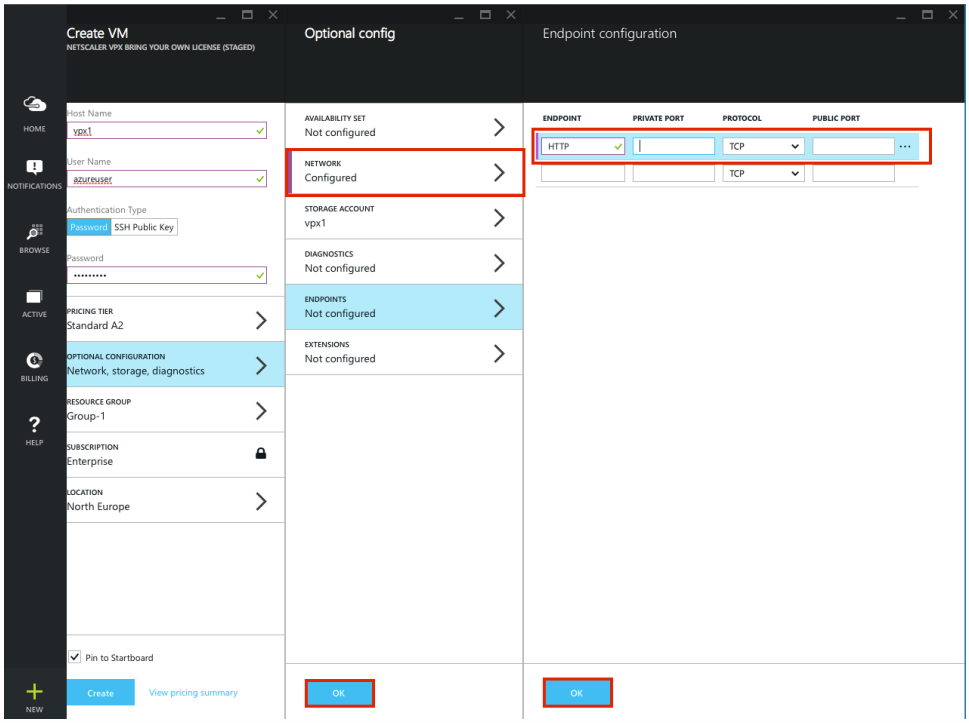
© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Click Virtual Network and select the virtual network you have created.
   Note: It is recommended that you create network settings before you begin provisioning the VM so that the network information is available during provisioning. If you create a virtual network during VM provisioning, you will not be able to set different subnets.
3. Click Subnet to select the specific subnet.
4. Click Domain Name to select an existing cloud service or to create a new one.
5. In IP Addresses, specify a static IP address.
   Note: If you expect that you might have to shut down and temporarily de-allocate the NetScaler VPX virtual machine at any time, assign a static Internal IP address. If you do not assign a static internal IP address, Azure might assign the virtual machine a different IP address each time it restarts, and the virtual machine might become inaccessible.
6. Click OK for each option to make sure each configured setting is saved.

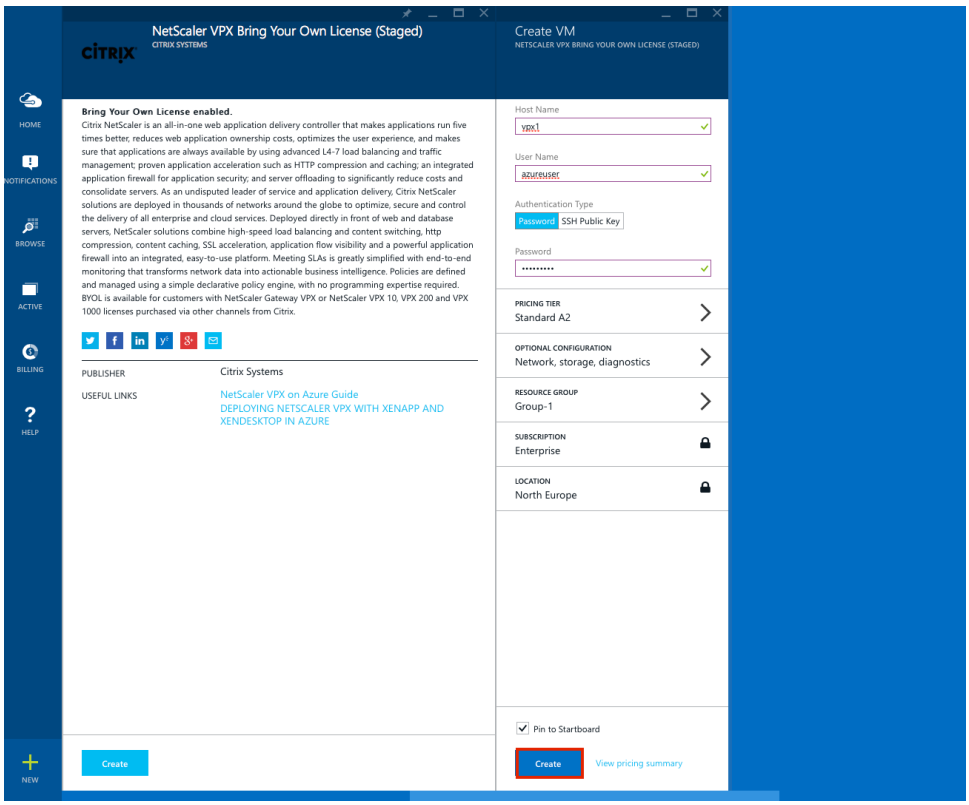


7. Click Endpoints and add the public and private endpoints you want to use. Note that SSH endpoint is automatically set by Azure with a random public port and a private port as 22. Therefore, provisioning will fail if you add SSH. Click OK to save the endpoints.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



8. Click Create.



9. Review the offer details and legal terms on the Create VM page and click Create.
   On the Portal page you can view the progress of the NetScaler VPX virtual machine provisioning.

   After the virtual machine is provisioned, a “deployment ID” is generated for use as the password to log on to the NetScaler virtual machine the first time. To view this deployment ID, double-click the virtual machine instance.

   Notes:

   - The default username to log on to the NetScaler virtual machine is nsroot. However, the default password is set

© 1999-2017 Citrix Systems, Inc. All rights reserved.

to the “deployment ID” that is generated after the virtual machine is provisioned. You can change the password after you log on to the instance.
- The password for the user that is created during the virtual machine provisioning remains the same as the one you had provided. Make sure that you do not provide nsroot as this user ID.

Creating a NetScaler VPX Instance on Microsoft Azure by Using the Classic Azure Management Portal

Obtain the NetScaler VPX image from the Citrix download page and upload it to Azure. Then, use the classic Azure management portal to create a NetScaler VPX instance.

To create a NetScaler VPX instance on Microsoft Azure by using the classic Azure management portal

1. On the **Virtual Machines** page, click **New**.
2. Navigate to **Compute** > **Virtual Machine** > **From Gallery** > **My Images**, select the VPX image that you have uploaded, and click the arrow at the right bottom corner.
3. On the **Virtual machine configuration** page, specify values for the required fields.
4. On the next screen, create a new cloud service. You can either configure the endpoints here, or at a later stage after the virtual machine is provisioned.
5. On the next screen, be sure to clear the **The VM agent that supports extensions is already installed** checkbox. The agent brings up the NetScaler virtual machine with the specified parameters by exchanging heartbeats and status with the Azure Fabric Controller.

For instructions on how to create a NetScaler VPX instance on Azure by using the Azure PowerShell, see Provisioning NetScaler VPX with Single NIC by Using Azure PowerShell.

After the virtual machine is provisioned, a "deployment ID" is generated for use as the password to log on to the NetScaler virtual machine the first time. To view this deployment ID, double-click the virtual machine instance.

> **Note**
>
> - The default username to log on to the NetScaler virtual machine is nsroot. However, the default password is set to the “deployment ID” that is generated after the virtual machine is provisioned. You can change the password after you log on to the instance.
> - The password for the user that is created during the virtual machine provisioning remains the same as that you had provided. Make sure that you do not provide nsroot as this user ID.

If you expect that you might have to shut down and temporarily deallocate the NetScaler VPX virtual machine at any time, assign a static Internal IP address while creating the virtual machine. If you do not assign a static internal IP address, Azure might assign the virtual machine a different IP address each time it restarts, and the virtual machine might become inaccessible. You can assign static internal IP address only when you provision the virtual machine through Azure PowerShell. For information about creating a virtual machine with a static internal IP address, see Provisioning NetScaler VPX on Azure with a Static IIP by Using Azure PowerShell.

Accessing the NetScaler VPX Virtual Machine

You can access the NetScaler instance either through its graphical user interface (GUI) or through the command line interface (CLI). You can use either the cloud service IP address or the PIP to access the NetScaler virtual machine instance.

To log on to the virtual machine, use either of the following credentials:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- The username and password specified while creating the virtual machine.
- nsroot as the username and the “deployment ID” that was generated by Azure during virtual machine provisioning.

You can change the password after you log on to the instance.

To access the NetScaler instance through the GUI

In a browser’s address field, type the cloud service public IP address provided by Azure during virtual machine provisioning, or type the PIP address.
Note: Make sure you have created an endpoint to allow access to the private port 80 or 443 when accessing the GUI by using the cloud service IP.

To access the NetScaler instance through the CLI

Use any command line access tool (for example, Putty). Specify either the cloud service public IP address provided by Azure during NetScaler VPX provisioning, or specify the PIP address. Use SSH protocol with port 22.

Note: Make sure that you have created an endpoint to allow access to the private port 22 when accessing the CLI by using the cloud service IP.
For information about getting started with a NetScaler appliance, see http://support.citrix.com/proddocs/topic/netscaler-10-5/ns-gen-getting-started-wrapper-10-con.html

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Multiple Azure VIPs for a NetScaler VPX in the Azure Service Management (ASM)

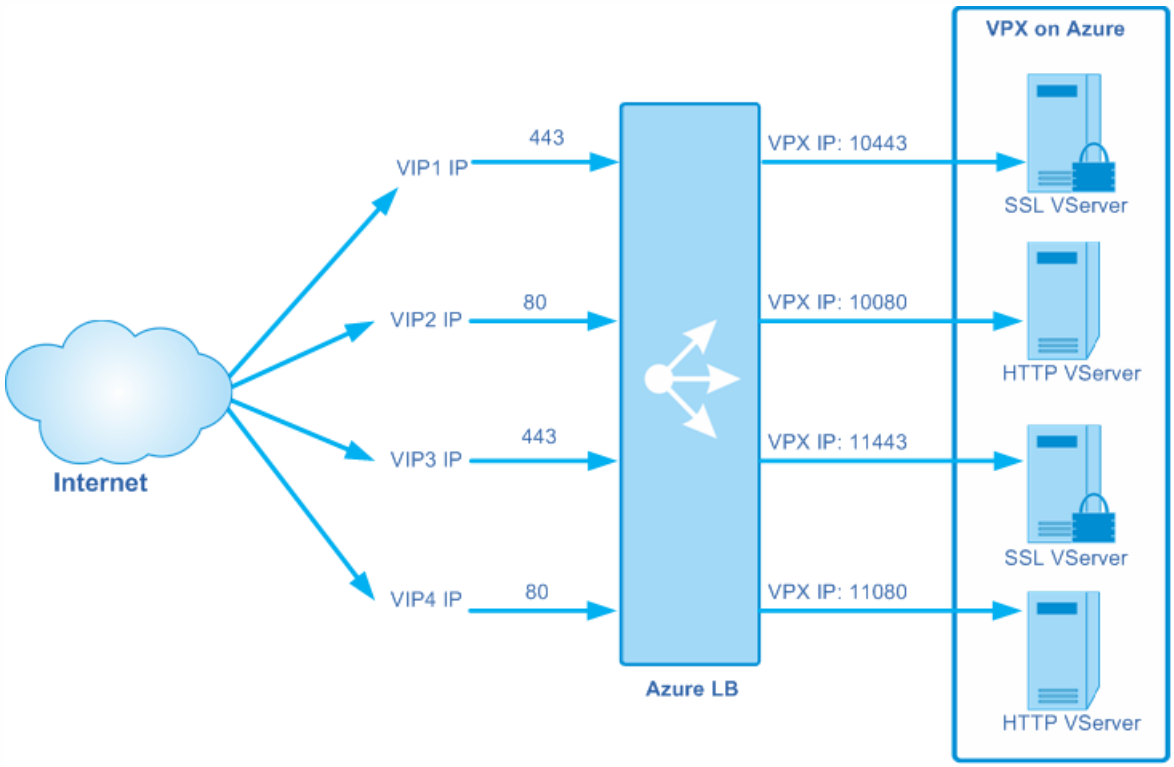
May 20, 2016

> **Note**
>
> This document refers to the Azure public virtual IP (VIP) address as the cloud service IP address to avoid confusion with the NetScaler virtual IP (VIP) address.

Microsoft Azure now allows you to configure multiple cloud service IP addresses (Azure public virtual IP addresses (VIPs)) to access Azure cloud services. This enables you to host multiple NetScaler virtual servers with unique FQDNs.

For example, you have four virtual servers with unique FQDNs configured on your NetScaler VPX VM. In the Azure architecture, these virtual servers are represented by the internal single IP assigned to your VM and a private port. However, you want different public cloud service IP addresses for each of these virtual servers. To achieve this, you can configure four public cloud service IP addresses to access the Azure load balancer. Then, map each public cloud service IP address and port to a virtual server represented by the internal IP of your VM and the private port.

The following image illustrates how you can configure multiple cloud service IP addresses (Azure public VIPs) for NetScaler virtual servers with unique FQDNs.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

In this example, there are two SSL virtual servers and two HTTP virtual servers configured on the NetScaler VPX VM. Each virtual server is represented by the VPX VM internal IP address and a unique private port. Four cloud service IP addresses (Azure public VIPs) are configured that are used by the clients to access the Azure load balancer. Each cloud service IP:public port is mapped to an internal IP:unique port. For example, VIP1 IP:443 is mapped to VPX IP:10443. So, all traffic from the client to this SSL virtual server is accessed through public VIP1 IP:443.

For more information about multiple VIPs per cloud service, see https://azure.microsoft.com/en-us/documentation/articles/load-balancer-multivip/

Configuration Steps

You can configure multiple cloud service IP addresses by using Azure PowerShell. You have to first add a cloud service IP address (Azure VIP) to a cloud service and then associate this IP address to an endpoint. For information about the commands to configure this, see https://azure.microsoft.com/en-us/documentation/articles/load-balancer-multivip/

You can also associate one cloud service IP address with two NetScaler VPX virtual machines in an HA mode. For example, you have configured multiple cloud service IP addresses to access the Azure load balancer and you want to configure two virtual machines named VPX1 and VPX2 in an HA mode. You can associate only one cloud service IP (say, VIP1) to both the VPX VMs. For information about the configuration steps, see https://azure.microsoft.com/en-us/documentation/articles/load-balancer-multivip/#how-to-enable-load-balancing-on-a-specific-vip

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring NetScaler VPX in High Availability Mode in Azure Service Management (ASM)

May 20, 2016

You can set up two NetScaler VPX instances for high availability in either active-active or active-passive mode.

When two NetScaler VPX instances are configured in active-active mode, both instances must have the same configuration. The client traffic is distributed across the virtual servers in both the instances by the Azure load balancer. The VIP addresses in both the instances are different and should match the NSIP of that VPX instance.

The active-passive mode provides failover capability. In this mode, the VPX instances synchronize their configuration states. When the primary instance fails, the secondary instance takes over.

For information about high availability in NetScaler appliance, see http://support.citrix.com/proddocs/topic/ns-system-10-5-map/ns-nw-ha-intro-wrppr-con.html

In a Microsoft Azure deployment, a high availability configuration of two NetScaler virtual machines is achieved by using the Azure load balancer, which distributes the client traffic across the virtual servers configured on both the NetScaler instances. Two types of Azure load balancers are available for high availability:

- **Azure external load balancer**: If the client traffic originates from the Internet, you have to deploy the external load balancer between the Internet and the NetScaler VPX instances to distribute client traffic.
- **Azure internal load balancer**: If the client traffic originates from within the cloud service, or is forwarded by a gateway or firewall within the cloud service, you have to deploy the internal load balancer to distribute client traffic.

To achieve high availability on Azure, you must add the two NetScaler VMs as a load balanced set and configure the endpoints.

Points to Consider Before You Begin Configuration

Note the following before you begin configuring the NetScaler instances in high availability mode in the Azure cloud.

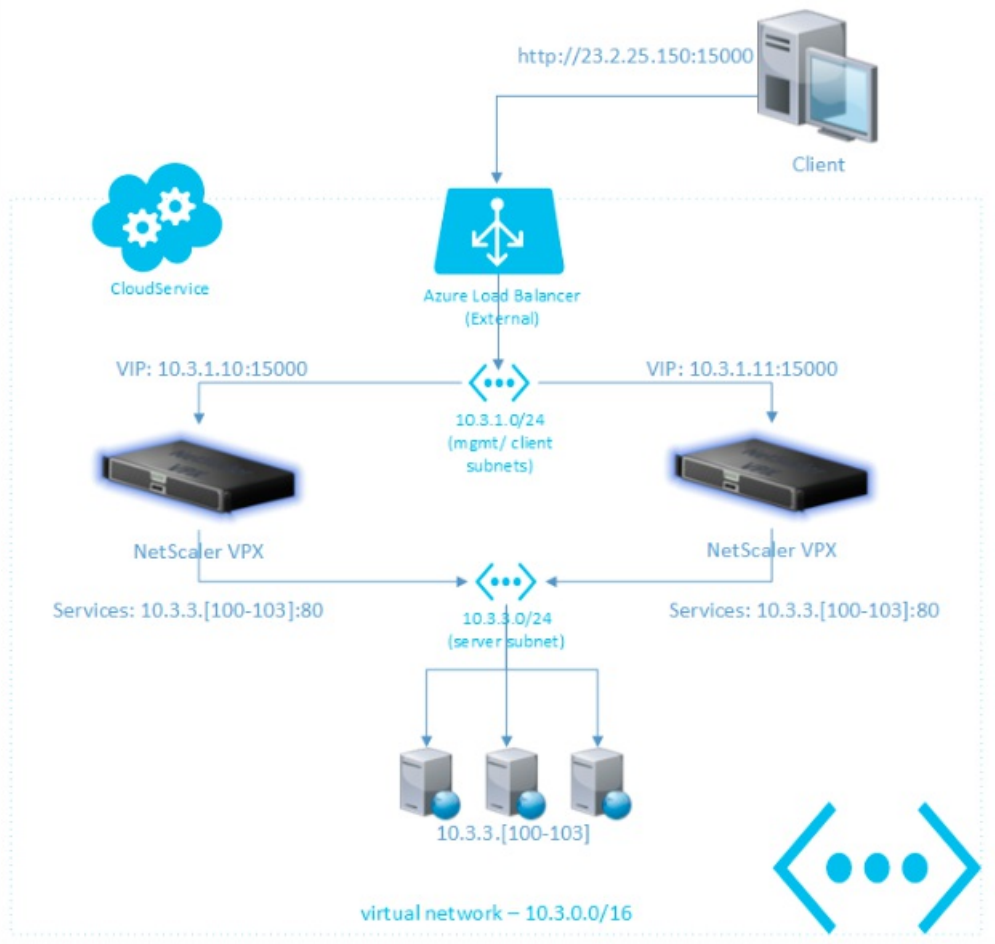
- The two NetScaler virtual machines that you want to add to a load balanced set should be provisioned in the same cloud service.
- A load balanced set applies only to a VM's default NIC. Therefore the VIP has to be configured on the VPX's default NIC.
- The endpoints are configured on both the NetScaler VMs and are bound to the Azure load balancer.
- In an active-passive deployment, the Azure load balancer monitors both the primary and the secondary NetScaler VM by sending them TCP probes. These TCP probes are sent on port 9000.

Configuring NetScaler High Availability with the Azure External Load Balancer

If your client traffic originates from the Internet, you have to deploy the external load balancer to create a high availability configuration of NetScaler virtual machines in a load-balanced set.

The following figure shows how high availability is achieved in active-active mode by using the external load balancer. The two NetScaler VMs are in a load-balanced set sharing an endpoint that accepts client traffic from the Internet over port 15000. The Azure external load balancer load balances these client requests between the two virtual machines.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Before you begin configuring the load-balanced set through the Azure portal, do one of the following:

- For an active-active deployment, configure the required services on the two NetScaler virtual machines.
- For an active-passive deployment, configure the NetScaler virtual machines as primary and secondary nodes by using the following command: add ha node <ID> <IP address>.

To configure the load-balanced set by using the Azure management portal

1. Select the first virtual machine that you want to make part of HA pair, click Endpoint, and then click Add.
2. Select Add a stand-alone endpoint, specify values for name, protocol, public and private ports, and select Create a load-balanced set.
3. Set the parameters as follows:
   - Probe protocol – TCP
   - Probe port – 9000
   - Probe interval – 5 seconds
   - Number of probes - 2
4. Select the second virtual machine of the HA pair, and add an endpoint.
5. Select Add an endpoint to an existing load-balanced set, and then specify values for name, protocol, public ports, and private ports. Click the tick mark to complete the configuration.

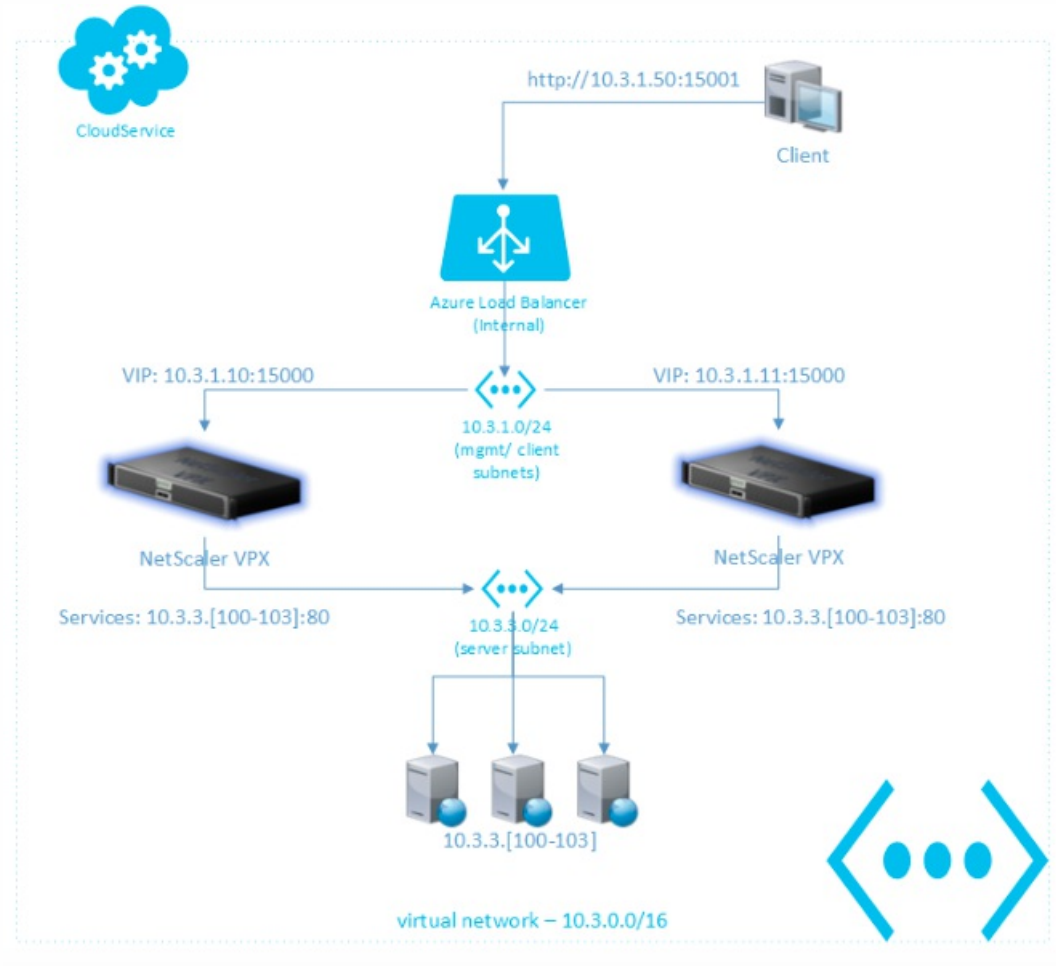
Configuring NetScaler High Availability with the Azure Internal Load Balancer

If your client traffic originates from within the cloud service or a virtual network with a regional scope, you have to deploy the internal load balancer to achieve high availability of NetScaler virtual machines added to a load-balanced set.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: You can configure an internal load balancer only by using Azure PowerShell.

The following figure shows how high availability is achieved in an active-active mode by using the internal load balancer. The two NetScaler virtual machines are in a load-balanced set sharing an endpoint that accepts client traffic from the Internet at port 15001. The Azure internal load balancer load balances these client requests between the two virtual machines.



Before you begin configuring the load-balanced set by using Azure PowerShell, do one of the following:

- For an active-active deployment, configure the required services on the two NetScaler virtual machines.
- For an active-passive deployment, configure the NetScaler virtual machines as primary and secondary nodes by using the following command: add ha node <ID> <IP address>.

You can configure the load-balanced set only by using Azure PowerShell. For information about the cmdlet that you need to run, see Configuring NetScaler VPX HA with Azure Internal Load Balancer by Using Azure PowerShell.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Appendix: PowerShell Commands

May 20, 2016

This appendix provides cmdlets with which you can perform the following tasks using Azure PowerShell:

- Provision and configure NetScaler in Azure Resource Manager (ARM)
  - Provision NetScaler VPX in Standalone Mode
  - Configure NetScaler VPX HA with Azure External Load Balancer
  - Configure NetScaler VPX HA with Azure Internal Load Balancer
- Provision and configure NetScaler in Azure Service Management (ASM)
  - Provision NetScaler VPX in the Azure cloud
  - Configure NetScaler VPX HA with Azure External Load Balancer
  - Configure NetScaler VPX HA with Azure Internal Load Balancer
  - Provision NetScaler VPX in Azure with Static IIP

Provision NetScaler VPX in Standalone Mode in Azure PowerShell in ARM

1. Create Resource Group

$rgName="<resource group name>"

$locName="<location name, such as West US>"

New-AzureRmResourceGroup -Name $rgName -Location $locName

For example:

$rgName = "ARM-VPX"

$locName = "East Asia"

New-AzureRmResourceGroup -Name $rgName -Location $locName

2. Create Storage Account

You must pick a globally unique name for your storage account that contains only lowercase letters and numbers.

$saName="<storage account name>"

$saType="<storage account type, specify one: Standard_LRS, Standard_GRS, Standard_RAGRS, or Premium_LRS>"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName –Type $saType -Location $locName

For example:

$saName="vpxstorage"

$saType="Standard_LRS"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName –Type $saType -Location $locName

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. Create Availability Set

$avName="<availability set name>"

New-AzureRmAvailabilitySet – Name $avName – ResourceGroupName $rgName -Location $locName

4. Create Virtual Network

Add a new virtual network with at least one subnet if it is not created previously.

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix 10.0.1.0/24 ß (this parameter value should be as per your requirement)

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix 10.0.2.0/24

New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName -Location $locName -AddressPrefix 10.0.0.0/16 -Subnet $frontendSubnet,$backendSubnet

For example:

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix 10.0.1.0/24

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix 10.0.2.0/24

New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName -Location $locName -AddressPrefix 10.0.0.0/16

-Subnet $frontendSubnet,$backendSubnet

5. Create NIC

From above subnets, frontend Subnet is at index 0 and backend Subnet is at index 1. NIC can be created in multiple ways:

a) **NIC with Public IP address**

$nicName="<name of the NIC of the VM>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName $rgName -Location $locName -AllocationMethod Dynamic

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -PublicIpAddressId $pip.Id

b) **NIC with Public IP and DNS label**

$nicName="<name of the NIC of the VM>"

$domName="<domain name label>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName $rgName -DomainNameLabel $domName -Location $locName -AllocationMethod Dynamic

Before assigning $domName, check it is available or not by using command:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Test-AzureRmDnsAvailability -DomainQualifiedName $domName -Location $locName

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -PublicIpAddressId $pip.Id

For example:

$nicName="frontendNIC"

$domName="vpxazure"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName $rgName -DomainNameLabel $domName -Location

$locName -AllocationMethod Dynamic

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -SubnetId $vnet.

Subnets[0].Id -PublicIpAddressId $pip.Id

c) **NIC with Dynamic Public Address and Static Private IP address**

$nicName="<name of the NIC of the VM>"

$staticIP="<available static IP address on the subnet>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName $rgName -Location $locName -AllocationMethod Dynamic

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -PublicIpAddressId $pip.Id -PrivateIpAddress $staticIP

6. Create Virtual Machine

$vmName="<VM name>"

$vmSize="<VM size string>"

$avSet=Get-AzureRmAvailabilitySet –Name $avName –ResourceGroupName $rgName

$vm=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

7. Get NetScaler Image

$pubName="<Image publisher name>"

$offerName="<Image offer name>"

$skuName="<Image SKU name>"

$cred=Get-Credential -Message "Type the name and password of the local administrator account."

Provide your credentials that is used to login into NetScaler VPX

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

$vm=Set-AzureRmVMOperatingSystem -VM $vm -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm=Set-AzureRmVMSourceImage -VM $vm -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm=Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id

For example:

$pubName="citrix"

Note: The following command is used for displaying all offers from Citrix:

Get-AzureRMVMImageOffer -Location $locName -Publisher $pubName | Select Offer

$offerName="netscalervpx110-6531"

Note: The following command is used to know sku offered by the publisher for specific offer name:

Get-AzureRMVMImageSku -Location $locName -Publisher $pubName -Offer $offerName | Select Skus

8. Create NetScaler Virtual Instance

$diskName="<name identifier for the disk in Azure storage, such as OSDisk>"

For example:

$diskName="dynamic”

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds/" + $diskName + ".vhd"

$vm=Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri -CreateOption fromImage

When you create a NetScaler virtual machine from the images present in MarketPlace, use the following command to specify the virtual machine plan:

Set-AzureRmVMPlan -VM $vm -Publisher citrix -Product netscalervpx110-6531 -Name netscalerbyol

Note: Product parameter should match with offer name parameter specified earlier. Name parameter should match with SKU that was selected previously.

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm

Provision NetScaler VPX in HA with Azure External Load Balancer in ARM

1. Create Resource Group

$rgName="<resource group name>"

$locName="<location name, such as West US>"

New-AzureRmResourceGroup -Name $rgName -Location $locName

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

For example:

$rgName = "ARM-LB-NS"

$locName = "East Asia"

New-AzureRmResourceGroup -Name $rgName -Location $locName

2. Create Storage Account

You must pick a globally unique name for your storage account that contains only lowercase letters and numbers.

$saName="<storage account name>"

$saType="<storage account type, specify one: Standard_LRS, Standard_GRS, Standard_RAGRS, or Premium_LRS>"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName –Type $saType -Location $locName

For example:

$saName="vpxstorage"

$saType="Standard_LRS"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName –Type $saType -Location $locName

3. Create Availability Set

$avName="<availability set name>"

New-AzureRmAvailabilitySet –Name $avName –ResourceGroupName $rgName -Location $locName

4. Create Virtual Network

Add a new Vnet with at least one subnet, if it was not created previously.

$vnetName = “LBVnet”

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix 10.0.1.0/24

Note: The AddressPrefix parameter value should be as per your requirement.

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix 10.0.2.0/24

New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName $rgName -Location $locName -AddressPrefix 10.0.0.0/16 -Subnet $frontendSubnet,$backendSubnet

5. Create Public IP Address

$publicIP1 = New-AzureRmPublicIpAddress -Name PublicIp1 -ResourceGroupName $rgName -Location $locName –AllocationMethod Static -DomainNameLabel nsvpx

Note: Before using, check for the availability of the value for DomainNameLabel.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

6. Create Front-end IP Address

$FIPName = “ELBFIP”

$frontendIP1 = New-AzureRmLoadBalancerFrontendIpConfig -Name $FIPName -PublicIpAddress $publicIP1

7. Create Back-end Pool

$BEPool = “LB-backend-Pool”

$beaddresspool1= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name $BEPool

8. Create Health Probe

Create TCP health probe with port 9000 and interval of 5 seconds.

$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name HealthProbe -Protocol Tcp -Port 9000 –IntervalInSeconds 5 -ProbeCount 2

9. Create Load Balancer Rules

Create a load balancer rule for each service for which you want to perform load balancing.

For example:

For load balancing HTTP service:

$lbrule1 = New-AzureRmLoadBalancerRuleConfig -Name "HTTP-LB" -FrontendIpConfiguration $frontendIP1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort 80 -BackendPort 80

10. Create Inbound NAT Rules

Create NAT rules for services that does not require to be load balanced.

For example:

SSH access to NetScaler VPX instance:

$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -Name SSH1 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -FrontendPort 22 -BackendPort 22

$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -Name SSH2 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -FrontendPort 10022 -BackendPort 22

Note: Protocol - FrontEndPort - BackendPort triplet should not be the same for any two NAT rules.

11. Create Load Balancer Entity

$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name ELB -Location $locName -InboundNatRule $inboundNATRule1, $inboundNATRule2 -FrontendIpConfiguration $frontendIP1 -LoadBalancingRule $lbrule1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe

© 1999-2017 Citrix Systems, Inc. All rights reserved.

12. Create NIC

Create two NICs and associate each NIC with each NetScaler VPX instance

a) NIC1 with VPX1

For example:

$nicName="NIC1”

$lbName="ELB"

$bePoolIndex=0 ß Rule indexes starts from 0.

$natRuleIndex=0

$subnetIndex=0 ß Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName

$nic1=New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -Subnet $vnet.Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule $lb.InboundNatRules[$natRuleIndex]

b) NIC2 with VPX2

For example:

$nicName="NIC2”

$lbName="ELB"

$bePoolIndex=0

$natRuleIndex=1 ß Second Inbound NAT (ssh) rule we need to use

$subnetIndex=0 ß Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName

$nic2=New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -Subnet $vnet.Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule $lb.InboundNatRules[$natRuleIndex]

13. Create NetScaler VPX Instances

Create two NetScaler VPX instances as part of the same resource group and availability set, and attach them to the same external Load balancer.

a) NetScaler VPX instance 1

For example:

$vmName="VPX1"

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
$vmSize="Standard_A3"

$avSet=Get-AzureRmAvailabilitySet –Name $avName –ResourceGroupName $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message "Type Credentials which will be used to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version
"latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $nic1.Id

$diskName="dynamic”

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/" + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm1
```

b) NetScaler VPX instance 2

For example:

```
$vmName="VPX2"

$vmSize="Standard_A3"

$avSet=Get-AzureRmAvailabilitySet –Name $avName –ResourceGroupName $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message " Type Credentials which will be used to login to VPX instance "

$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version
"latest"

$vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $nic2.Id

$diskName="dynamic”

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/" + $diskName  + ".vhd"

$vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm2

14. Configure NetScaler VPX Instances

When both NetScaler VPX instances are running, connect to both VPX instances through SSH to configure the virtual machines:

a) Active-Active: Run the same set of configuration commands on both NetScaler instances command line.

b) Active-Passive: Run the following command on NetScaler command line on both nodes:

- add ha node #nodeID <nsip of otherVPX>
- run configuration commands on primary node only.

Provision NetScaler VPX in HA with Azure Internal Load Balancer in ARM

1. Create Resource Group

$rgName="<resource group name>"

$locName="<location name, such as West US>"

New-AzureRmResourceGroup -Name $rgName -Location $locName

For example:

$rgName = "ARM-LB-NS"

$locName = "East Asia"

New-AzureRmResourceGroup -Name $rgName -Location $locName

2. Create Storage Account

Select a globally unique name that contains only lowercase letters and numbers for the storage account.

$saName="<storage account name>"

$saType="<storage account type, specify one: Standard_LRS, Standard_GRS, Standard_RAGRS, or Premium_LRS>"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName –Type $saType -Location $locName

For example:

$saName="vpxstorage"

$saType="Standard_LRS"

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName –Type $saType -Location $locName

3. Create Availability Set

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

$avName="<availability set name>"

New-AzureRmAvailabilitySet –Name $avName –ResourceGroupName $rgName -Location $locName

For example:

$avname=”HA-AV1”

4. Create Virtual Network and Subnets

Create a new virtual network with at least one subnet, if it was not created previously.

$vnetName = “LBVnet”

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix 10.0.1.0/24 ß (this parameter value should be as per your requirement)

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix 10.0.2.0/24 ß (this parameter value should be as per your requirement)

Assign frontendSubnet and Backendsubnet to the virtual network:

$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName $rgName -Location $locName -AddressPrefix 10.0.0.0/16 -Subnet $frontendSubnet,$backendSubnet

For example:

AddressPrefix-2.0.0.0/16

5. Create Front-end IP Address

$frontendIP = New-AzureRmLoadBalancerFrontendIpConfig -Name LB-Frontend -PrivateIpAddress 10.0.2.5 -SubnetId $vnet.subnets[0].Id

If the front-end subnet is the first element of array vnet, subnetId should be $vnet.Subnets[0].Id

If the front-end subnet is the second element, subnetId should be $vnet.Subnets[1].Id, and so on.

For example: PrivateAddressIP - 10.0.2.6

For example: PrivateAddressIP - 10.0.2.5

6. Create Back-end Pool

$beaddresspool= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name "LB-backend"

7. Create Inbound NAT Rules

1) $inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -Name "Inboundnatrule1" -FrontendIpConfiguration $frontendIP -Protocol TCP -FrontendPort 3441 -BackendPort 3389

Front-end port and back-end port are as per user requirement.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

For example: 8080, 8081

2) $inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -Name "RDP2" -FrontendIpConfiguration $frontendIP -Protocol TCP -FrontendPort 3442 -BackendPort 3389

Front-end port and back-end port are as per user requirement.

For example: 8080, 8081

8. Create Health Probe

$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name "HealthProbe" " -Protocol tcp -Port 9000 -IntervalInSeconds 5 -ProbeCount 2

9. Create Load Balancer Rules

$lbrule = New-AzureRmLoadBalancerRuleConfig -Name "lbrule1" -FrontendIpConfiguration $frontendIP -BackendAddressPool $beAddressPool -Probe $healthProbe -Protocol Tcp -FrontendPort 80 -BackendPort 80

Front-end port and back-end port are as per user requirement.

10. Create Load Balancers

$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgname -Name "InternalLB" -Location $locName -FrontendIpConfiguration $frontendIP -InboundNatRule $inboundNATRule1,$inboundNatRule2 -LoadBalancingRule $lbrule -BackendAddressPool $beAddressPool -Probe $healthProbe

11. Create NIC

1) Create NIC for NetScaler VPX instance 1

$backendnic1= New-AzureRmNetworkInterface -ResourceGroupName $rgName -Name lb-nic1-be -Location $locName -PrivateIpAddress 10.0.2.6 -Subnet $backendSubnet -LoadBalancerBackendAddressPool $nrplb.BackendAddressPools[0] -LoadBalancerInboundNatRule $nrplb.InboundNatRules[0]

The parameter PrivateIpAddress can have any private IP address as per user requirement.

For example: 10.0.2.14

2) Create NIC for NetScaler VPX instance 2

$backendnic2= New-AzureRmNetworkInterface -ResourceGroupName "NRP-RG" -Name lb-nic2-be -Location $locName -PrivateIpAddress 10.0.2.7 -Subnet $backendSubnet -LoadBalancerBackendAddressPool $nrplb.BackendAddressPools[0] -LoadBalancerInboundNatRule $nrplb.InboundNatRules[1].

The parameter PrivateIpAddress can have any private IP address as per user requirement.

For example: 10.0.2.15

12. Create NetScaler VPX Instances

Create two NetScaler VPX instances as part of the same resource group and availability set, and attach the VPX instances

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

to the same external Azure load balancer.

1) Create NetScaler VPX instance 1

Parameter samples:

```
$vmName="VPX1"

$vmSize="Standard_A3"

$avSet=Get-AzureRmAvailabilitySet –Name $avName –ResourceGroupName $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message "Type Credentials which will be used to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $backendnic1.Id

$diskName="dynamic”

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/" + $diskName  + ".vhd"

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm1
```

2) Create NetScaler VPX instance 2

Sample Parameters:

```
$vmName="VPX2"

$vmSize="Standard_A3"

$avSet=Get-AzureRmAvailabilitySet –Name $avName –ResourceGroupName $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message " Type Credentials which will be used to login to VPX instance "

$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version "latest"

$vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $backendnic2.Id
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
$diskName="dynamic”

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/" + $diskName  + ".vhd"

$vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm2
```

13. Configure NetScaler VPX Instances

When both NetScaler VPX instances are running, connect to both VPX instances through SSH to configure the virtual machines:

a) Active-Active: Run the same set of configuration commands on both NetScaler instances command line.

b) Active-Passive: Run the following command on NetScaler command line on both nodes:

- add ha node #nodeID <nsip of otherVPX>
- run configuration commands on primary node only.

Provisioning NetScaler VPX with Single NIC by Using Azure PowerShell in ASM

```
$image = Get-AzureVMImage -ImageName "vpx_105_51_1040"
$vm = New-AzureVMConfig -Name "vpx_bwc_105" -InstanceSize "Medium" -Image $image.ImageName
Add-AzureProvisioningConfig -VM $vm -Linux -Linuxuser "user" -Password "Citrix123"
Set-AzureSubnet –SubnetNames “mgmt.-1” –VM  $vm
Set-AzureStaticVNetIP -IPAddress "10.3.1.10" -VM $vm
New-AzureService -ServiceName "vpx105" -Location "North Europe"
Set-AzureEndpoint -Name “SSH” –LocalPort 22 -PublicPort 22 -Protocol tcp -VM $vm
Add-AzureEndpoint -Name “HTTP” -LocalPort 80 -PublicPort 80 -Protocol tcp -VM $vm
Add-AzureEndpoint -Name “SSL” –LocalPort 443 -PublicPort 443 -Protocol tcp -VM $vm
New-AzureVM -ServiceName "vpx105" -VNetName "systemtestazure" -VM $vm
```

Configuring NetScaler VPX HA with Azure External Load Balancer by Using Azure PowerShell in ASM

```
$vm1 = "vpxne1"
$vm2 = "vpxne2”
$svc = "vpxne"
$lbsetname = "azurelb"
$epname = "vip"

Get-AzureVM -Name $vm1 -ServiceName $svc | Add-AzureEndpoint -LBSetName $lbsetname -Name
$epname -Protocol "tcp" -LocalPort 15000 -PublicPort 15000 –ProbeProtocol TCP –ProbePort
9000 –ProbeIntervalInSeconds 5 | Update-AzureVM

Get-AzureVM -Name $vm1 -ServiceName $svc | Add-AzureEndpoint -LBSetName $lbsetname -Name
$epname -Protocol "tcp" -LocalPort 15000 -PublicPort 15000 –ProbeProtocol TCP –ProbePort
9000 –ProbeIntervalInSeconds 5 | Update-AzureVM
```

Configuring NetScaler VPX HA with Azure Internal Load Balancer by Using Azure PowerShell in ASM

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
$svc = "vpxne" [Cloud service name]
$ilb = "vpx-ilb" [Name of the internal load balancer] $subnet = "mgmtsub"
[Management subnet where the default NIC exists]
$IP = "10.3.1.50" [IP address from the management subnet]

Add-AzureInternalLoadBalancer -ServiceName $svc -InternalLoadBalancerName $ilb
–SubnetName $subnet –StaticVNetIPAddress $IP

$vm1 = "vpxne1"
$vm2 = "vpxne2"
$epname = "vip2"
$prot = "TCP"
$locport = 1500
$pubport = 1500
$lbsetname = "azure_ilb"

Get-AzureVM -ServiceName $svc -Name $vm1 | Add-AzureEndpoint -Name $epname -Protocol
$prot -LocalPort $locport -PublicPort $pubport -LBSetName $lbsetname -InternalLoadBalancerName
 $ilb –ProbeProtocol TCP –ProbePort 9000 –ProbeIntervalInSeconds 5 | Update-AzureVM

Get-AzureVM -ServiceName $svc -Name $vm2 | Add-AzureEndpoint -Name $epname -Protocol
$prot -LocalPort $locport -PublicPort $pubport -LBSetName $lbsetname -InternalLoadBalancerName
 $ilb –ProbeProtocol TCP –ProbePort 9000 –ProbeIntervalInSeconds 5 | Update-AzureVM
```

Provisioning NetScaler VPX on Azure with a Static IIP by Using Azure PowerShell in ASM

Before you begin provisioning the virtual machine with a static internal IP address, make sure that

- Azure virtual network is configured
- You have downloaded Azure PowerShell

Run the following commands on Azure PowerShell to provision NetScaler VPX on Azure with static IIP.

```
$image = Get-AzureVMImage -ImageName “image1045"

$vm = New-AzureVMConfig -Name "vpx3" -InstanceSize "Medium" -Image $image.ImageName

Add-AzureProvisioningConfig -VM $vm -Linux -Linuxuser "user" -Password “Citrix123”

Set-AzureSubnet

–SubnetNames

“mgmt-1” -VM $vm

Set-AzureStaticVNetIP -IPAddress “1.10.1.5" -VM $vm

New-AzureService -ServiceName “singlevpx" -Location "North Europe”  <To add new cloud service>

Set-AzureEndpoint -Name “SSH” –LocalPort 22 -PublicPort 1322 -Protocol tcp -VM $vm
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
New-AzureVM -ServiceName “singlevpx" -VNetName “pcwestus" -VM $vm
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Basic System Settings

Aug 23, 2013

After installing a Citrix NetScaler virtual appliance, you need to access it to configure the basic settings. Initially, you must access the NetScaler command line through the respective management application of the virtualization host (either Citrix XenCenter for Citrix XenServer or VMware vSphere client for VMware ESX) to specify a NetScaler IP (NSIP) address, subnet mask, and default gateway. The NSIP is the management address at which you can then access the NetScaler command line, through an SSH client, or access the configuration utility. You can use either of these access methods, or the console, to continue with basic configuration.

To access the configuration utility, type the NSIP into the address field of any browser (for example, http://<NSIP_address>). You need Java RunTime Environment (JRE) version 1.6 or later.

Setting Up the Initial Configuration by Using the NetScaler Virtual Appliance Console

Updated: 2013-08-23

Your first task after installing a NetScaler virtual appliance on a virtualization host is to use the NetScaler virtual appliance console in the XenCenter client or vSphere client to configure the following initial settings.

Note: If you have installed a virtual appliance on XenServer by using Command Center, you do not have to configure these settings. Command Center implicitly configures the settings during installation. For more information about provisioning virtual appliance from Command Center, see the "Command Center" documentation.

NetScaler IP address (NSIP):

The IP address at which you access a NetScaler or a NetScaler virtual appliance for management purposes. A physical NetScaler or virtual appliance can have only one NSIP. You must specify this IP address when you configure the virtual appliance for the first time. You cannot remove an NSIP address.

Netmask:

The subnet mask associated with the NSIP address.

Default Gateway:

You must add a default gateway on the virtual appliance if you want access it through SSH or the configuration utility from an administrative workstation or laptop that is on a different network.

To configure the initial settings on the virtual appliance through the virtual appliance Console by using the management application

1. Connect to the XenServer or VMware ESX server on which the virtual appliance is installed by using XenCenter or vSphere client, respectively.
2. In the details pane, on the Console tab, log on to the virtual appliance by using the administrator credentials.
3. At the prompts, enter the NSIP address, subnet mask, and default gateway, and then save the configuration.

After you have set up an initial configuration through the NetScaler virtual appliance Console in the management application, you can use either the NetScaler command-line interface or the configuration utility to complete the configuration or to change the initial settings.

Configuring NetScaler Virtual Appliance by Using the Command Line Interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2013-08-23

You can use the command line interface to set up the NSIP, Mapped IP (MIP), Subnet IP (SNIP), and hostname. You can also configure advanced network settings and change the time zone.

To complete initial configuration by using the command line interface

1. Use either the SSH client or the NetScaler virtual appliance Console in XenCenter to access the command line interface.
2. Log on to the virtual appliance, using the administrator credentials.
3. At the command prompt, type config ns to run the configuration script.
4. To complete the initial configuration, follow the prompts.

You have now completed the basic configuration of the virtual appliance. To continue the configuration process, choose one of the following options:

Citrix NetScaler Load Balancing Switch

If you are configuring the virtual appliance as a standard NetScaler load balancing switch with other licensed features, see "Traffic Management."

Citrix NetScaler Application Firewall

If you are configuring the virtual appliance as a standalone application firewall, see "Application Firewall."

For more information about the various features supported on the NetScaler virtual appliance, see Features at a Glance.

Configuring NetScaler Virtual Appliance by Using the Configuration Utility

Updated: 2013-08-23

To use the Setup Wizard to set up the NetScaler virtual appliance, you must access the configuration utility from your Web browser. You can use the Setup Wizard to configure the NSIP, MIP, SNIP, hostname, and default gateway. You can also configure settings for a Web application by using an application template. You can also configure the appliance as a load balancer for Citrix XenDesktop or Citrix XenApp.

For information about application templates, see "AppExpert."

For information about the load balancing feature of a virtual appliance, see "Traffic Management."

To configure initial settings by using the configuration utility

1. In the address field of a Web browser, type: http://<NSIP address>
2. In User Name and Password, type the administrator credentials.
3. In Deployment Type, select NetScaler ADC.
4. In Start in, select Configuration, and then click Login.
5. In the Setup Wizard, click Next and follow the instructions.

You have now completed the basic configuration of the virtual appliance. To continue the configuration process, choose one of the following options:

Citrix NetScaler Load Balancing Switch

If you are configuring the virtual appliance as a standard NetScaler load balancing switch with other licensed features, see "Traffic Management."

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Citrix NetScaler Application Firewall

If you are configuring the virtual appliance as a standalone application firewall, see "Application Firewall."

For more information about the various features supported on the NetScaler virtual appliance, see Features at a Glance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding Common Network Topologies

Sep 04, 2013

As described in "Physical Deployment Modes," you can deploy the Citrix NetScaler appliance either inline between the clients and servers or in one-arm mode. Inline mode uses a two-arm topology, which is the most common type of deployment.

This document includes the following:

- Setting Up Common Two-Arm Topologies
- Setting Up Common One-Arm Topologies

Setting Up Common Two-Arm Topologies

In a two-arm topology, one network interface is connected to the client network and another network interface is connected to the server network, ensuring that all traffic flows through the appliance. This topology might require you to reconnect your hardware and also might result in a momentary downtime. The basic variations of two-arm topology are multiple subnets, typically with the appliance on a public subnet and the servers on a private subnet, and transparent mode, with both the appliance and the servers on the public network.

Setting Up a Simple Two-Arm Multiple Subnet Topology

Updated: 2013-10-01

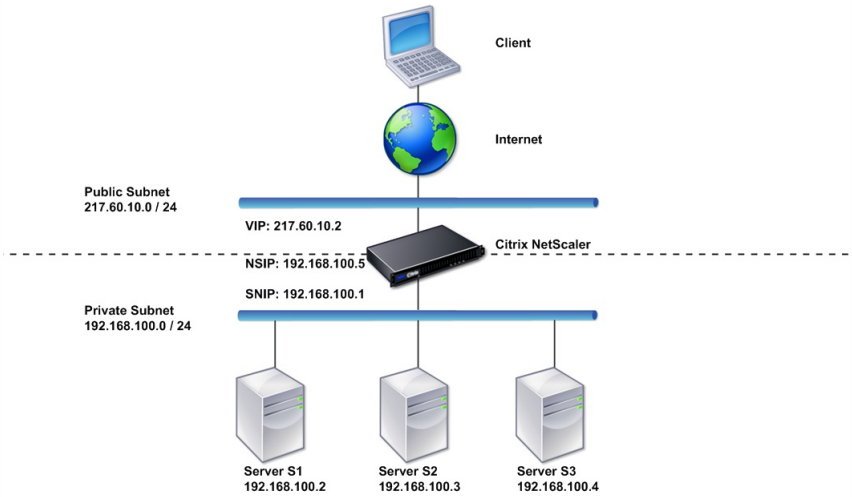
One of the most commonly used topologies has the NetScaler appliance inline between the clients and the servers, with a virtual server configured to handle the client requests. This configuration is used when the clients and servers reside on different subnets. In most cases, the clients and servers reside on public and private subnets, respectively.

For example, consider an appliance deployed in two-arm mode for managing servers S1, S2, and S3, with a virtual server of type HTTP configured on the appliance, and with HTTP services running on the servers. The servers are on a private subnet and a SNIP is configured on the appliance to communicate with the servers. The Use SNIP (USNIP) option must be enabled on the appliance so that it uses the SNIP instead of the MIP.

As shown in the following figure, the VIP is on public subnet 217.60.10.0, and the NSIP, the servers, and the SNIP are on private subnet 192.168.100.0/24.

Figure 1. Topology Diagram for Two-Arm Mode, Multiple Subnets

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Task overview: To deploy a NetScaler appliance in two-arm mode with multiple subnets

1. Configure the NSIP and default gateway, as described in "Configuring the NetScaler IP Address (NSIP)."
2. Configure the SNIP, as described in "Configuring Subnet IP Addresses."
3. Enable the USNIP option, as described in "To enable or disable USNIP mode."
4. Configure the virtual server and the services, as described in "Creating a Virtual Server" and "Configuring Services."
5. Connect one of the network interfaces to a private subnet and the other interface to a public subnet.

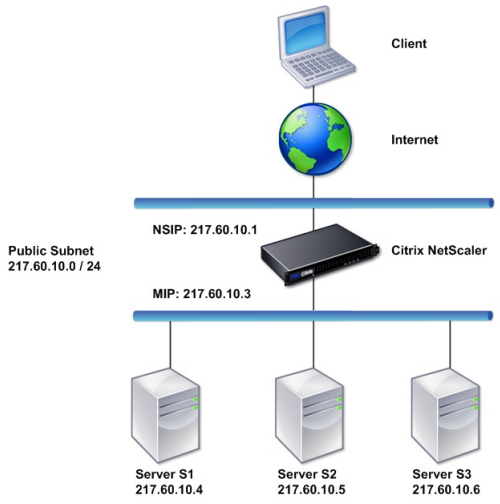
Setting Up a Simple Two-Arm Transparent Topology

Updated: 2013-09-16

Use transparent mode if the clients need to access the servers directly, with no intervening virtual server. The server IP addresses must be public because the clients need to be able to access them. In the example shown in the following figure, a NetScaler appliance is placed between the client and the server, so the traffic must pass through the appliance. You must enable L2 mode for bridging the packets. The NSIP and MIP are on the same public subnet, 217.60.10.0/24.

Figure 2. Topology Diagram for Two-Arm, Transparent Mode

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Task overview: To deploy a NetScaler in two-arm, transparent mode

1. Configure the NSIP, MIP, and default gateway, as described in "Configuring the Basic System Settings."
2. Enable L2 mode, as described in "Enabling and Disabling Layer 2 Mode."
3. Configure the default gateway of the managed servers as the MIP.
4. Connect the network interfaces to the appropriate ports on the switch.

Setting Up Common One-Arm Topologies

The two basic variations of one-arm topology are with a single subnet and with multiple subnets.

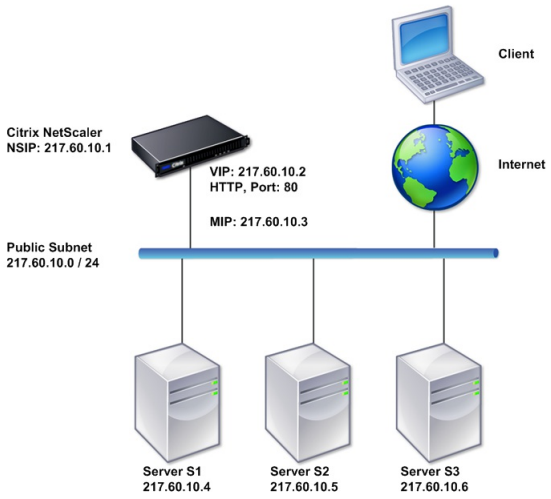
Setting Up a Simple One-Arm Single Subnet Topology

Updated: 2013-08-23

You can use a one-arm topology with a single subnet when the clients and servers reside on the same subnet. For example, consider a NetScaler deployed in one-arm mode for managing servers S1, S2, and S3. A virtual server of type HTTP is configured on a NetScaler, and HTTP services are running on the servers. As shown in the following figure, the NetScaler IP address (NSIP), the Mapped IP address (MIP), and the server IP addresses are on the same public subnet, 217.60.10.0/24.

Figure 3. Topology Diagram for One-Arm Mode, Single Subnet

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Task overview: To deploy a NetScaler in one-arm mode with a single subnet

1. Configure the NSIP, MIP, and the default gateway, as described in "Configuring the NetScaler IP Address (NSIP)".
2. Configure the virtual server and the services, as described in "Creating a Virtual Server" and "Configuring Services".
3. Connect one of the network interfaces to the switch.

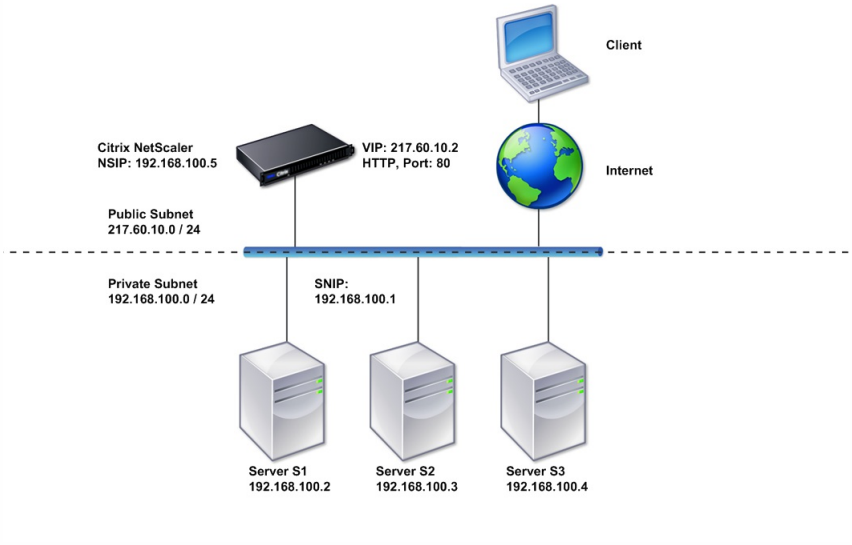
Setting Up a Simple One-Arm Multiple Subnet Topology

Updated: 2013-08-23

You can use a one-arm topology with multiple subnets when the clients and servers reside on the different subnets. For example, consider a NetScaler appliance deployed in one-arm mode for managing servers S1, S2, and S3, with the servers connected to switch SW1 on the network. A virtual server of type HTTP is configured on the appliance, and HTTP services are running on the servers. These three servers are on the private subnet, so a subnet IP address (SNIP) is configured to communicate with them. The Use Subnet IP address (USNIP) option must be enabled so that the appliance uses the SNIP instead of a MIP. As shown in the following figure, the virtual IP address (VIP) is on public subnet 217.60.10.0/24; the NSIP, SNIP, and the server IP addresses are on private subnet 192.168.100.0/24.

Figure 4. Topology Diagram for One-Arm Mode, Multiple Subnets

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Task overview: To deploy a NetScaler appliance in one-arm mode with multiple subnets

1. Configure the NSIP and the default gateway, as described in "Configuring the NetScaler IP Address (NSIP)".
2. Configure the SNIP and enable the USNIP option, as described in "Configuring Subnet IP Addresses".
3. Configure the virtual server and the services, as described in "Creating a Virtual Server" and "Configuring Services".
4. Connect one of the network interfaces to the switch.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring System Management Settings

Sep 04, 2013

Once your initial configuration is in place, you can configure settings to define the behavior of the Citrix NetScaler appliance and facilitate connection management. You have a number of options for handling HTTP requests and responses. Routing, bridging, and MAC based forwarding modes are available for handling packets not addressed to the NetScaler. You can define the characteristics of your network interfaces and can aggregate the interfaces. To prevent timing problems, you can synchronize the NetScaler clock with a Network Time Protocol (NTP) server. The NetScaler can operate in various DNS modes, including as an authoritative domain name server (ADNS). You can set up SNMP for system management and customize syslog logging of system events. Before deployment, verify that your configuration is complete and correct.

This document includes the following:

- Configuring System Settings
- Configuring Modes of Packet Forwarding
- Configuring Network Interfaces
- Configuring Clock Synchronization
- Configuring DNS
- Configuring SNMP
- Verifying the Configuration

Note: In addition to the tasks listed above, you can configure Syslog logging. For instructions, see "Audit Logging."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring System Settings

Sep 04, 2013

Configuration of system settings includes basic tasks such as configuring HTTP ports to enable connection keep-alive and server offload, setting the maximum number of connections for each server, and setting the maximum number of requests per connection. You can enable client IP address insertion for situations in which a proxy IP address is not suitable, and you can change the HTTP cookie version.

You can also configure a NetScaler appliance to open FTP connections on a controlled range of ports instead of ephemeral ports for data connections. This improves security, because opening all ports on the firewall is insecure. You can set the range anywhere from 1,024 to 64,000.

Before deployment, go through the verification checklists to verify your configuration. To configure HTTP parameters and the FTP port range, use the NetScaler configuration utility.

You can modify the types of HTTP parameters described in the following table.

Table 1. HTTP Parameters

Parameter Type	Specifies
HTTP Port Information	The web server HTTP ports used by your managed servers. If you specify the ports, the appliance performs request switching for any client request that has a destination port matching a specified port. Note: If an incoming client request is not destined for a service or a virtual server that is specifically configured on the appliance, the destination port in the request must match one of the globally configured HTTP ports. This allows the appliance to perform connection keep-alive and server off-load.
Limits	The maximum number of connections to each managed server, and the maximum number of requests sent over each connection. For example, if you set Max Connections to 500, and the appliance is managing three servers, it can open a maximum of 500 connections to each of the three servers. By default, the appliance can create an unlimited number of connections to any of the servers it manages. To specify an unlimited number of requests per connection, set Max Requests to 0. Note: If you are using the Apache HTTP server, you must set Max Connections equal to the value of the MaxClients parameter in the Apache httpd.conf file. Setting this parameter is optional for other web servers.
Client IP Insertion	Enable/disable insertion of the client's IP address into the HTTP request header. You can specify a name for the header field in the adjacent text box. When a web server managed by an appliance receives a mapped IP address or a subnet IP address, the server identifies it as the client’s IP address. Some applications need the client’s IP address for logging purposes or to dynamically determine the content to be served by the web server. You can enable insertion of the actual client IP address into the HTTP header request sent from the client to one, some, or all servers managed by the appliance. You can then access the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter Type	Specifies
	inserted address through a minor modification to the server (using an Apache module, ISAPI interface, or NSAPI interface).
Cookie Version	The HTTP cookie version to use when COOKIEINSERT persistence is configured on a virtual server. The default, version 0, is the most common type on the Internet. Alternatively, you can specify version 1.
Requests/Responses	Options for handling certain types of requests, and enable/disable logging of HTTP error responses.
Server Header Insertion	Insert a server header in NetScaler-generated HTTP responses.

To configure HTTP parameters by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Settings, click Change HTTP parameters.
3. In the Configure HTTP parameters dialog box, specify values for some or all of the parameters that appear under the headings listed in the table above.
4. Click OK.

To set the FTP port range by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Settings, click Change global system settings.
3. Under FTP Port Range, in the Start Port and End Port text boxes, type the lowest and highest port numbers, respectively, for the range you want to specify (for example, 5000 and 6000).
4. Click OK.

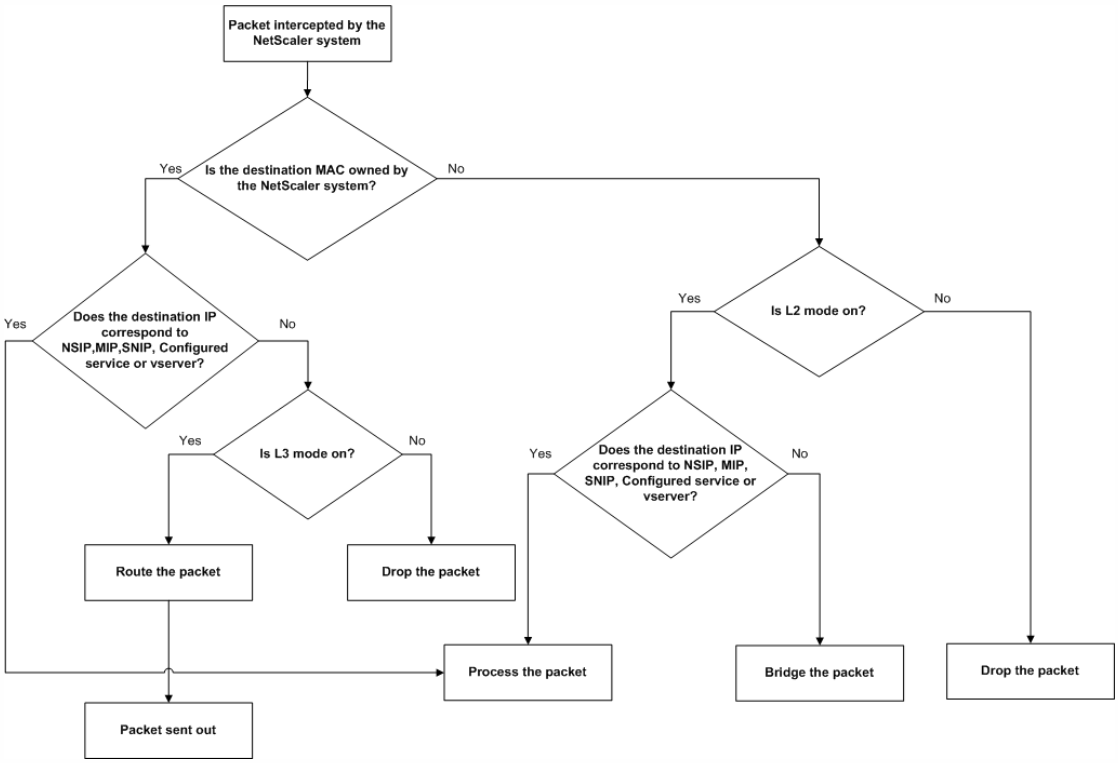
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Modes of Packet Forwarding

Jun 24, 2013

The NetScaler appliance can either route or bridge packets that are not destined for an IP address owned by the appliance (that is, the IP address is not the NSIP, a MIP, a SNIP, a configured service, or a configured virtual server). By default, L3 mode (routing) is enabled and L2 mode (bridging) is disabled, but you can change the configuration. The following flow chart shows how the appliance evaluates packets and either processes, routes, bridges, or drops them.

Figure 1. Interaction between Layer 2 and Layer 3 Modes



An appliance can use the following modes to forward the packets it receives:

- Layer 2 (L2) Mode
- Layer 3 (L3) Mode
- MAC-Based Forwarding Mode

Enabling and Disabling Layer 2 Mode

Updated: 2013-09-13

Layer 2 mode controls the Layer 2 forwarding (bridging) function. You can use this mode to configure a NetScaler appliance to behave as a Layer 2 device and bridge the packets that are not destined for it. When this mode is enabled, packets are not forwarded to any of the MAC addresses, because the packets can arrive on any interface of the appliance and each interface has its own MAC address.

With Layer 2 mode disabled (which is the default), the appliance drops packets that are not destined for one of its MAC address. If another Layer 2 device is installed in parallel with the appliance, Layer 2 mode must be disabled to prevent

© 1999-2017 Citrix Systems, Inc. All rights reserved.

bridging (Layer 2) loops. You can use the configuration utility or the command line to enable Layer 2 mode.

Note: The appliance does not support spanning tree protocol. To avoid loops, if you enable L2 mode, do not connect two interfaces on the appliance to the same broadcast domain.

To enable or disable Layer 2 mode by using the command line interface

At the command prompt, type the following commands to enable/disable Layer 2 mode and verify that it has been enabled/disabled:

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

Examples

```
> enable ns mode l2
 Done
> show ns mode

    Mode              Acronym        Status
    -------           -------        ------
 1)  Fast Ramp          FR            ON
 2)  Layer 2 mode       L2            ON
 .
 .
 .
 Done
>

> disable ns mode l2
 Done
> show ns mode

    Mode              Acronym        Status
    -------           -------        ------
 1)  Fast Ramp          FR            ON
 2)  Layer 2 mode       L2            OFF
 .
 .
 .
 Done
>
```

To enable or disable Layer 2 mode by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure modes.
3. In the Configure Modes dialog box, to enable Layer 2 mode, select the Layer 2 Mode check box. To disable Layer 2 mode, clear the check box.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Click OK. The Enable/Disable Mode(s)? message appears in the details pane.
5. Click Yes.

Enabling and Disabling Layer 3 Mode

Updated: 2013-09-13

Layer 3 mode controls the Layer 3 forwarding function. You can use this mode to configure a NetScaler appliance to look at its routing table and forward packets that are not destined for it. With Layer 3 mode enabled (which is the default), the appliance performs route table lookups and forwards all packets that are not destined for any appliance-owned IP address. If you disable Layer 3 mode, the appliance drops these packets.

To enable or disable Layer 3 mode by using the command line interface

At the command prompt, type the following commands to enable/disable Layer 3 mode and verify that it has been enabled/disabled:

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

Examples

```
> enable ns mode l3
 Done
> show ns mode

   Mode                    Acronym        Status
   -------                 -------        ------
 1)  Fast Ramp                FR            ON
 2)  Layer 2 mode              L2            OFF
 .
 .
 .
 9)  Layer 3 mode (ip forwarding)  L3            ON
 .
 .
 .
 Done
>

> disable ns mode l3
 Done
> show ns mode

   Mode                    Acronym        Status
   -------                 -------        ------
 1)  Fast Ramp                FR            ON
 2)  Layer 2 mode              L2            OFF
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
.
.
.
9)  Layer 3 mode (ip forwarding)  L3            OFF
.
.
.
Done
>
```

To enable or disable Layer 3 mode by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure modes.
3. In the Configure Modes dialog box, to enable Layer 3 mode, select the Layer 3 Mode (IP Forwarding) check box. To disable Layer 3 mode, clear the check box.
4. Click OK. The Enable/Disable Mode(s)? message appears in the details pane.
5. Click Yes.

Enabling and Disabling MAC-Based Forwarding Mode

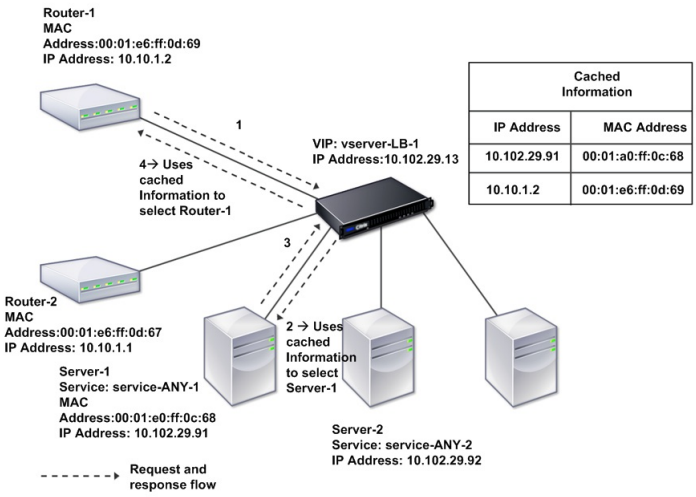
Updated: 2013-09-13

You can use MAC-based forwarding to process traffic more efficiently and avoid multiple-route or ARP lookups when forwarding packets, because the NetScaler appliance remembers the MAC address of the source. To avoid multiple lookups, the appliance caches the source MAC address of every connection for which it performs an ARP lookup, and it returns the data to the same MAC address.

MAC-based forwarding is useful when you use VPN devices because the appliance ensures that all traffic flowing through a particular VPN passes through the same VPN device.

The following figure shows the process of MAC-based forwarding.

Figure 2. MAC-Based Forwarding Process



© 1999-2017 Citrix Systems, Inc. All rights reserved.

When MAC-based forwarding is enabled, the appliance caches the MAC address of:

- The source (a transmitting device such as router, firewall, or VPN device) of the inbound connection.
- The server that responds to the requests.

When a server responds through an appliance, the appliance sets the destination MAC address of the response packet to the cached address, ensuring that the traffic flows in a symmetric manner, and then forwards the response to the client. The process bypasses the route table lookup and ARP lookup functions. However, when an appliance initiates a connection, it uses the route and ARP tables for the lookup function. To enable MAC-based forwarding, use the configuration utility or the command line.

Some deployments require the incoming and outgoing paths to flow through different routers. In these situations, MAC-based forwarding breaks the topology design. For a global server load balancing (GSLB) site that requires the incoming and outgoing paths to flow through different routers, you must disable MAC-based forwarding and use the appliance's default router as the outgoing router.

With MAC-based forwarding disabled and Layer 2 or Layer 3 connectivity enabled, a route table can specify separate routers for outgoing and incoming connections. To disable MAC-based forwarding, use the configuration utility or the command line.

To enable or disable MAC-based forwarding by using the command line interface

At the command prompt, type the following commands to enable/disable MAC-based forwarding mode and verify that it has been enabled/disabled:

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

  Example

```
> enable ns mode mbf
 Done
> show ns mode

   Mode                    Acronym        Status
   -------                 -------        ------
 1)  Fast Ramp               FR             ON
 2)  Layer 2 mode             L2            OFF
 .
 .
 .
 6)  MAC-based forwarding      MBF           ON
 .
 .
 .
 Done
>

> disable ns mode mbf
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done
> show ns mode

   Mode                    Acronym        Status
   -------                 -------        ------
1)  Fast Ramp              FR             ON
2)  Layer 2 mode           L2             OFF
.
.
.
6)  MAC-based forwarding   MBF            OFF
.
.
.
Done
>
```

To enable or disable MAC-based forwarding by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features group, click Configure modes.
3. In the Configure Modes dialog box, to enable MAC-based forwarding mode, select the MAC Based Forwarding check box. To disable MAC-based forwarding mode, clear the check box.
4. Click OK. The Enable/Disable Mode(s)? message appears in the details pane.
5. Click Yes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Network Interfaces

Aug 23, 2013

NetScaler interfaces are numbered in slot/port notation. In addition to modifying the characteristics of individual interfaces, you can configure virtual LANs to restrict traffic to specific groups of hosts. You can also aggregate links into high-speed channels.

Virtual LANs

The NetScaler supports (Layer 2) port and IEEE802.1Q tagged virtual LANs (VLANs). VLAN configurations are useful when you need to restrict traffic to certain groups of stations. You can configure a network interface to belong to multiple VLANs by using IEEE 802.1q tagging.

You can bind your configured VLANs to IP subnets. The NetScaler (if it is configured as the default router for the hosts on the subnets) then performs IP forwarding between these VLANs. A NetScaler supports the following types of VLANs.

Default VLAN

By default, the network interfaces on a NetScaler are included in a single, port-based VLAN as untagged network interfaces. This default VLAN has a VID of 1 and exists permanently. It cannot be deleted, and its VID cannot be changed.

Port-Based VLANs

A set of network interfaces that share a common, exclusive, Layer 2 broadcast domain define the membership of a port-based VLAN. You can configure multiple port-based VLANs. When you add an interface to a new VLAN as an untagged member, it is automatically removed from the default VLAN.

Tagged VLAN

A network interface can be a tagged or untagged member of a VLAN. Each network interface is an untagged member of only one VLAN (its native VLAN). The untagged network interface forwards the frames for the native VLAN as untagged frames. A tagged network interface can be a part of more than one VLAN. When you configure tagging, be sure that both ends of the link have matching VLAN settings. You can use the configuration utility to define a tagged VLAN (nsvlan) that can have any ports bound as tagged members of the VLAN. Configuring this VLAN requires a reboot of the NetScaler and therefore must be done during initial network configuration.

Link Aggregate Channels

Link aggregation combines incoming data from multiple ports into a single high speed link. Configuring the link aggregate channel increases the capacity and availability of the communication channel between a NetScaler and other connected devices. An aggregated link is also referred to as a channel.

When a network interface is bound to a channel, the channel parameters have precedence over the network interface parameters. A network interface can be bound to only one channel. Binding a network interface to a link aggregate channel changes the VLAN configuration. That is, binding network interfaces to a channel removes them from the VLANs that they originally belonged to and adds them to the default VLAN. However, you can bind the channel back to the old VLAN, or to a new one. For example, if you have bound network interfaces 1/2 and 1/3 to a VLAN with ID 2, and then you bind them to link aggregate channel LA/1, the network interfaces are moved to the default VLAN, but you can bind them to VLAN 2.

Note: You can also use Link Aggregation Control Protocol (LACP) to configure link aggregation. For more information, see ""Configuring Link Aggregation by Using the Link Aggregation Control Protocol."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Clock Synchronization

Aug 23, 2013

You can configure your NetScaler appliance to synchronize its local clock with a Network Time Protocol (NTP) server. This ensures that its clock has the same date and time settings as the other servers on your network. NTP uses User Datagram Protocol (UDP) port 123 as its transport layer. You have to add NTP servers in the NTP configuration file so that the appliance periodically gets updates from these servers.

If you do not have a local NTP server, you can find a list of public, open access, NTP servers at the official NTP site at http://www.ntp.org.

To configure clock synchronization on your appliance

1. Log on to the command line and enter the shell command.
2. At the shell prompt, copy the ntp.conf file from the /etc directory to the /nsconfig directory. If the file already exists in the /nsconfig directory, make sure that you remove the following entries from the ntp.conf file:
   restrict localhost

   restrict 127.0.0.2

   These entries are required only if you want to run the device as a time server. However, this feature is not supported on the NetScaler.

3. Edit /nsconfig/ntp.conf by typing the IP address for the desired NTP server under the file's server and restrict entries.
4. Create a file named rc.netscaler in the /nsconfig directory, if the file does not already exist in the directory.
5. Edit /nsconfig/rc.netscaler by adding the following entry: /usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ntpd.log &
   This entry starts the ntpd service, checks the ntp.conf file, and logs messages in the /var/log directory.

   Note: If the time difference between the NetScaler and the time server is more than 1000 sec, the ntpd service terminates with a message to the NetScaler log. To avoid this, you need to start ntpd with the -g option, which forcibly syncs the time. Add the following entry in /nsconfig/rc.netscaler:
   /usr/sbin/ntpd -g -c /nsconfig/ntp.conf -l /var/log/ntpd.log &

   If you do not want to forcibly sync the time when there is a large difference, you can set the date manually and then start ntpd again. You can check the time difference between the appliance and the time server by running the following command in the shell:

   ntpdate -q <IP address or domain name of the NTP server>
6. Reboot the appliance to enable clock synchronization.
   Note: If you want to start time synchronization before you restart the appliance, enter the following command (which you added to the rc.netscaler file in step 5) at the shell prompt:
   /usr/sbin/ntpd -c /nsconfig/ntp.conf -l /var/log/ ntpd.log &

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DNS

Sep 04, 2013

You can configure a NetScaler appliance to function as an Authoritative Domain Name Server (ADNS), DNS proxy server, End Resolver, or Forwarder. You can add DNS resource records such as SRV Records, AAAA Records, A Records, MX Records, NS Records, CNAME Records, PTR Records, and SOA Records. Also, the appliance can balance the load on external DNS servers.

A common practice is to configure an appliance as a forwarder. For this configuration, you need to add external name servers. After you have added the external servers, you should verify that your configuration is correct.

You can add, remove, enable, and disable external name servers. You can create a name server by specifying its IP address, or you can configure an existing virtual server as the name server.

When adding name servers, you can specify IP addresses or virtual IP addresses (VIPs). If you use IP addresses, the appliance load balances requests to the configured name servers in a round robin manner. If you use VIPs, you can specify any load balancing method.

To add a name server by using the command line interface

At the command prompt, type the following commands to add a name server and verify the configuration:

- add dns nameServer <IP>
- show dns nameServer <IP>

  Example

  ```
  > add dns nameServer 10.102.29.10
   Done
  > show dns nameServer 10.102.29.10
  1)      10.102.29.10  -  State: DOWN
   Done
  >
  ```

To add a name server by using the configuration utility

1. Navigate to Traffic Management > DNS > Name Servers.
2. In the details pane, click Add.
3. In the Create Name Server dialog box, select IP Address.
4. In the IP Address text box, type the IP address of the name server (for example, 10.102.29.10). If you are adding an external name server, clear the Local check box.
5. Click Create, and then click Close.
6. Verify that the name server you added appears in the Name Servers pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

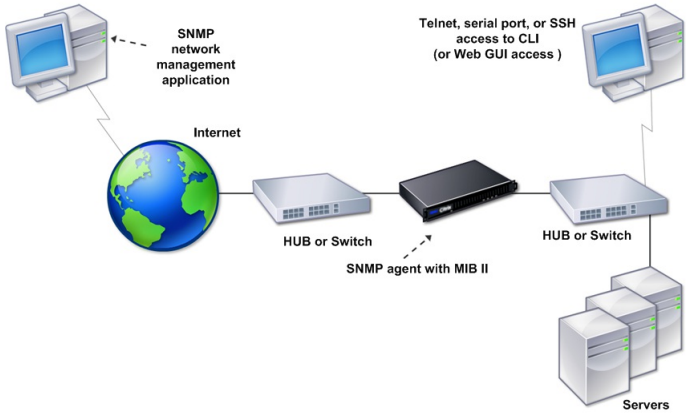
Configuring SNMP

May 14, 2012

The Simple Network Management Protocol (SNMP) network management application, running on an external computer, queries the SNMP agent on the NetScaler. The agent searches the management information base (MIB) for data requested by the network management application and sends the data to the application.

SNMP monitoring uses traps messages and alarms. SNMP traps messages are asynchronous events that the agent generates to signal abnormal conditions, which are indicated by alarms. For example, if you want to be informed when CPU utilization is above 90 percent, you can set up an alarm for that condition. The following figure shows a network with a NetScaler that has SNMP enabled and configured.

Figure 1. SNMP on the NetScaler



The SNMP agent on a NetScaler supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3). Because it operates in bilingual mode, the agent can handle SNMPv2 queries, such as Get-Bulk, and SNMPv1 queries. The SNMP agent also sends traps compliant with SNMPv2 and supports SNMPv2 data types, such as counter64. SNMPv1 managers (programs on other servers that request SNMP information from the NetScaler) use the NS-MIB-smiv1.mib file when processing SNMP queries. SNMPv2 managers use the NS-MIB-smiv2.mib file.

The NetScaler supports the following enterprise-specific MIBs:

A subset of standard MIB-2 groups

Provides MIB-2 groups SYSTEM, IF, ICMP, UDP, and SNMP.

A system enterprise MIB

Provides system-specific configuration and statistics.

To configure SNMP, you specify which managers can query the SNMP agent, add SNMP trap listeners that will receive the SNMP trap messages, and configure SNMP Alarms.

Adding SNMP Managers

You can configure a workstation running a management application that complies with SNMP version 1, 2, or 3 to access an

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appliance. Such a workstation is called an SNMP manager. If you do not specify an SNMP manager on the appliance, the appliance accepts and responds to SNMP queries from all IP addresses on the network. If you configure one or more SNMP managers, the appliance accepts and responds to SNMP queries from only those specific IP addresses. When specifying the IP address of an SNMP manager, you can use the netmask parameter to grant access from entire subnets. You can add a maximum of 100 SNMP managers or networks.

To add an SNMP manager by using the command line interface

At the command prompt, type the following commands to add an SNMP manager and verify the configuration:

- add snmp manager <IPAddress> ... [-netmask <netmask>]
- show snmp manager <IPAddress>

  Example

```
> add snmp manager 10.102.29.5 -netmask 255.255.255.255
 Done
> show snmp manager 10.102.29.5
1)     10.102.29.5        255.255.255.255
 Done
>
```

To add an SNMP manager by using the configuration utility

1. In the navigation pane, expand System, expand SNMP, and then click Managers.
2. In the details pane, click Add.
3. In the Add SNMP Manager dialog box, in the IP Address text box, type the IP address of the workstation running the management application (for example, 10.102.29.5).
4. Click Create, and then click Close.
5. Verify that the SNMP manager you added appears in the Details section at the bottom of the pane.

Adding SNMP Traps Listeners

Updated: 2013-09-13

After configuring the alarms, you need to specify the trap listener to which the appliance will send the trap messages. Apart from specifying parameters like IP address and the destination port of the trap listener, you can specify the type of trap (either generic or specific) and the SNMP version.

You can configure a maximum of 20 trap listeners for receiving either generic or specific traps.

To add an SNMP trap listener by using the command line interface

At the command prompt, type the following command to add an SNMP trap and verify that it has been added:

- add snmp trap specific <IP>
- show snmp trap

  Example

```
> add snmp trap specific 10.102.29.3
 Done
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> show snmp trap
Type     DestinationIP   DestinationPort  Version  SourceIP      Min-Severity  Community
----     -------------   ---------------  -------  --------      ------------  ---------
generic  10.102.29.9     162              V2       NetScaler IP  N/A           public
generic  10.102.29.5     162              V2       NetScaler IP  N/A           public
generic  10.102.120.101  162              V2       NetScaler IP  N/A           public
.
.
.
specific 10.102.29.3     162              V2       NetScaler IP  -             public
 Done
>
```

To add an SNMP trap listener by using the configuration utility

1. In the navigation pane, expand System, expand SNMP, and then click Traps.
2. In the details pane, click Add.
3. In the Create SNMP Trap Destination dialog box, in the Destination IP Address text box, type the IP address (for example, 10.102.29.3).
4. Click Create and then click Close.
5. Verify that the SNMP trap you added appears in the Details section at the bottom of the pane.

Configuring SNMP Alarms

Updated: 2013-09-13

You configure alarms so that the appliance generates a trap message when an event corresponding to one of the alarms occurs. Configuring an alarm consists of enabling the alarm and setting the severity level at which a trap is generated. There are five severity levels: Critical, Major, Minor, Warning, and Informational. A trap is sent only when the severity of the alarm matches the severity specified for the trap.

Some alarms are enabled by default. If you disable an SNMP alarm, the appliance will not generate trap messages when corresponding events occur. For example, if you disable the Login-Failure SNMP alarm, the appliance will not generate a trap message when a login failure occurs.

To enable or disable an alarm by using the command line interface

At the command prompt, type the following commands to enable or disable an alarm and verify that it has been enabled or disabled:

- set snmp alarm <trapName> [-state ENABLED | DISABLED]
- show snmp alarm <trapName>

Example

```
> set snmp alarm LOGIN-FAILURE -state ENABLED
 Done
> show snmp alarm LOGIN-FAILURE
  Alarm          Alarm Threshold  Normal Threshold  Time  State    Severity   Logging
  -----          ---------------  ----------------  ----  --------  ---------  --------
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
1) LOGIN-FAILURE  N/A          N/A          N/A   ENABLED   -       ENABLED
Done
>
```

To set the severity of the alarm by using the command line interface

At the command prompt, type the following commands to set the severity of the alarm and verify that the severity has been set correctly:

- set snmp alarm <trapName> [-severity <severity>]
- show snmp alarm <trapName>

Example

```
> set snmp alarm LOGIN-FAILURE -severity Major
Done
> show snmp alarm LOGIN-FAILURE
  Alarm          Alarm Threshold  Normal Threshold  Time  State    Severity   Logging
  -----          ---------------  ----------------  ----  --------  ---------  --------
1) LOGIN-FAILURE  N/A          N/A          N/A   ENABLED   Major     ENABLED
Done
>
```

To configure alarms by using the configuration utility

1. In the navigation pane, expand System, expand SNMP, and then click Alarms.
2. In the details pane, select an alarm (for example, LOGIN-FAILURE), and then click Open.
3. In the Configure SNMP Alarm dialog box, to enable the alarm, select Enabled in the State drop-down list. To disable the alarm, select Disabled.
4. In the Severity drop-down list, select a severity option (for example, Major).
5. Click OK, and then click Close.
6. Verify that the parameters for the SNMP alarm you configured are correctly configured by viewing the Details section at the bottom of the pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Verifying the Configuration

Nov 08, 2013

After you finish configuring your system, complete the following checklists to verify your configuration.

Configuration Checklist

- The build running is:
- There are no incompatibility issues. (Incompatibility issues are documented in the build's release notes.)
- The port settings (speed, duplex, flow control, monitoring) are the same as the switch's port.
- Enough mapped IP addresses have been configured to support all server-side connections during peak times.
  - The number of configured mapped IP addresses is: _____
  - The expected number of simultaneous server connections is:
    [ ] 62,000 [ ] 124,000 [ ] Other_____

Topology Configuration Checklist

- The routes have been used to resolve servers on other subnets.

  The routes entered are:

  _____

- If the NetScaler is in a public-private topology, reverse NAT has been configured.
- The failover (high availability) settings configured on the NetScaler resolve in a one arm or two-arm configuration. All unused network interfaces have been disabled:

  _____

- If the NetScaler is placed behind an external load balancer, then the load balancing policy on the external load balancer is not "least connection."
  The load balancing policy configured on the external load balancer is:

  _____

- If the NetScaler is placed in front of a firewall, the session time-out on the firewall is set to a value greater than or equal to 300 seconds.
  Note: The TCP idle connection timeout on a NetScaler appliance is 360 seconds. If the timeout on the firewall is also set to 300 seconds or more, then the appliance can perform TCP connection multiplexing effectively because connections will not be closed earlier.
  The value configured for the session time-out is: _____

Server Configuration Checklist

- "Keep-alive" has been enabled on all the servers.
  The value configured for the keep-alive time-out is: _____

- The default gateway has been set to the correct value. (The default gateway should either be a NetScaler or upstream router.) The default gateway is:

  _____

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- The server port settings (speed, duplex, flow control, monitoring) are the same as the switch port settings.

  _____

- If the Microsoft® Internet Information Server is used, buffering is enabled on the server.
- If an Apache Server is used, the MaxConn (maximum number of connections) parameter is configured on the server and on the NetScaler.
  The MaxConn (maximum number of connections) value that has been set is:

  _____

- If a Netscape® Enterprise Server™ is used, the maximum requests per connection parameter is set on the NetScaler. The maximum requests per connection value that has been set is:

  _____

Software Features Configuration Checklist

- Does the Layer 2 mode feature need to be disabled? (Disable if another Layer 2 device is working in parallel with a NetScaler.)
  Reason for enabling or disabling:

  _____

- Does the MAC-based forwarding feature need to be disabled? (If the MAC address used by return traffic is different, it should be disabled.)
  Reason for enabling or disabling:

  _____

- Does host-based reuse need to be disabled? (Is there virtual hosting on the servers?)
  Reason for enabling or disabling:

  _____

- Do the default settings of the surge protection feature need to be changed?
  Reason for changing or not changing:

  _____

Access Checklist

- The system IPs can be pinged from the client-side network.
- The system IPs can be pinged from the server-side network.
- The managed server(s) can be pinged through the NetScaler.
- Internet hosts can be pinged from the managed servers.
- The managed server(s) can be accessed through the browser.
- The Internet can be accessed from managed server(s) using the browser.
- The system can be accessed using SSH.
- Admin access to all managed server(s) is working.

Note: When you are using the ping utility, ensure that the pinged server has ICMP ECHO enabled, or your ping will not succeed.

Firewall Checklist

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following firewall requirements have been met:

- UDP 161 (SNMP)
- UDP 162 (SNMP trap)
- TCP/UDP 3010 (GUI)
- HTTP 80 (GUI)
- TCP 22 (SSH)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing Traffic on a NetScaler Appliance

Jun 24, 2013

The load balancing feature distributes client requests across multiple servers to optimize resource utilization. In a real-world scenario with a limited number of servers providing service to a large number of clients, a server can become overloaded and degrade the performance of the server farm. A Citrix NetScaler appliance uses load balancing criteria to prevent bottlenecks by forwarding each client request to the server best suited to handle the request when it arrives.

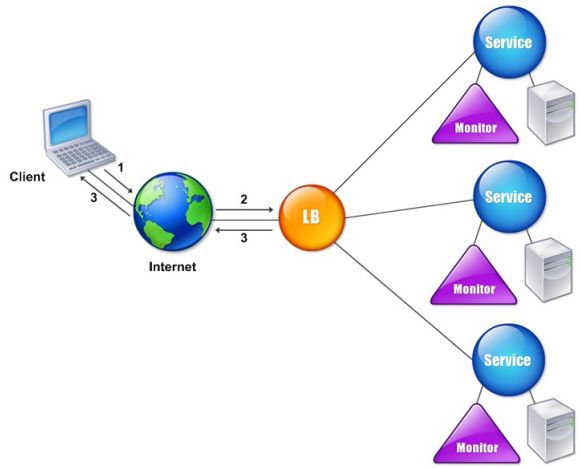
To configure load balancing, you define a virtual server to proxy multiple servers in a server farm and balance the load among them.

When a client initiates a connection to the server, a virtual server terminates the client connection and initiates a new connection with the selected server, or reuses an existing connection with the server, to perform load balancing. The load balancing feature provides traffic management from Layer 4 (TCP and UDP) through Layer 7 (FTP, HTTP, and HTTPS).

The NetScaler appliance uses a number of algorithms, called load balancing methods, to determine how to distribute the load among the servers. The default load balancing method is the Least Connections method.

A typical load balancing deployment consists of the entities described in the following figure.

Figure 1. Load Balancing Architecture



The entities function as follows:

- **Virtual server**. An entity that is represented by an IP address, a port, and a protocol. The virtual server IP address (VIP) is usually a public IP address. The client sends connection requests to this IP address. The virtual server represents a bank of servers.
- **Service**. A logical representation of a server or an application running on a server. Identifies the server's IP address, a port, and a protocol. The services are bound to the virtual servers.
- **Server object**. An entity that is represented by an IP address. The server object is created when you create a service. The IP address of the service is taken as the name of the server object. You can also create a server object and then create

© 1999-2017 Citrix Systems, Inc. All rights reserved.

services by using the server object.

- **Monitor**. An entity that tracks the health of the services. The appliance periodically probes the servers using the monitor bound to each service. If a server does not respond within a specified response timeout, and the specified number of probes fails, the service is marked DOWN. The appliance then performs load balancing among the remaining services.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Load Balancing

Jun 24, 2013

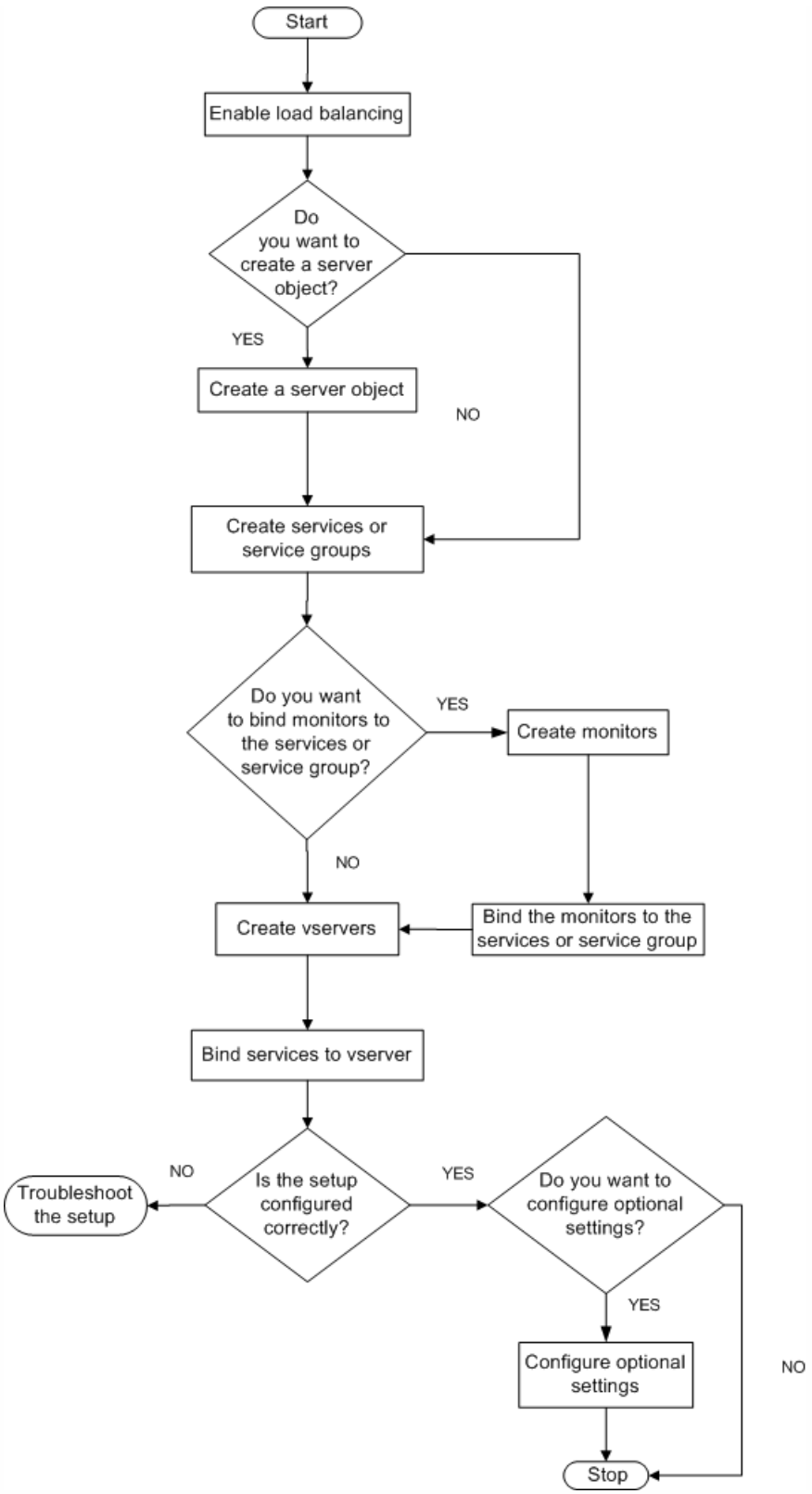
To configure load balancing, you must first create services. Then, you create virtual servers and bind the services to the virtual servers. By default, the NetScaler appliance binds a monitor to each service. After binding the services, verify your configuration by making sure that all of the settings are correct.

Note: After you deploy the configuration, you can display statistics that show how the entities in the configuration are performing. Use the statistical utility or the stat lb vserver <vserverName> command.
Optionally, you can assign weights to a service. The load balancing method then uses the assigned weight to select a service. For getting started, however, you can limit optional tasks to configuring some basic persistence settings, for sessions that must maintain a connection to a particular server, and some basic configuration-protection settings.

The following flow chart illustrates the sequence of the configuration tasks.

Figure 1. Sequence of Tasks to Configure Load Balancing

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Enabling Load Balancing

Before configuring load balancing, make sure that the load balancing feature is enabled.

To enable load balancing by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to enable load balancing and verify that it is enabled:

- enable feature lb
- show feature

Example

```
> enable feature lb
 Done
> show feature

   Feature              Acronym        Status
   -------              -------        ------
 1)  Web Logging          WL            OFF
 2)  Surge Protection     SP            OFF
 3)  Load Balancing       LB            ON
 .
 .
 .
 9)  SSL Offloading       SSL           ON
 .
 .
 .
Done
```

To enable load balancing by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Load Balancing check box, and then click OK.
4. In the Enable/Disable Feature(s)? message, click Yes.

Configuring Services and a Virtual Server

When you have identified the services you want to load balance, you can implement your initial load balancing configuration by creating the service objects, creating a load balancing virtual server, and binding the service objects to the virtual server.

To implement the initial load balancing configuration by using the command line interface

At the command prompt, type the following commands to implement and verify the initial configuration:

- add service <name> <IPaddress> <serviceType> <port>
- add lb vserver <vServerName> <serviceType> [<IPaddress> <port>]
- bind lb vserver <name> <serviceName>
- show service bindings <serviceName>

Example

```
> add service service-HTTP-1 10.102.29.5 HTTP 80
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 Done
> add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
 Done
> bind lb vserver vserver-LB-1 service-HTTP-1
 Done
> show service bindings service-HTTP-1
  service-HTTP-1 (10.102.29.5:80) - State : DOWN

    1)     vserver-LB-1 (10.102.29.60:80) - State : DOWN
 Done
```

To implement the initial load balancing configuration by using the configuration utility

1. Navigate to Traffic Management > Load Balancing.
2. In the details pane, under Getting Started, click Load Balancing wizard, and follow the instructions to create a basic load balancing setup.
3. Return to the navigation pane, expand Load Balancing, and then click Virtual Servers.
4. Select the virtual server that you configured and verify that the parameters displayed at the bottom of the page are correctly configured.
5. Click Open.
6. Verify that each service is bound to the virtual server by confirming that the Active check box is selected for each service on the Services tab.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Choosing and Configuring Persistence Settings

Sep 04, 2013

You must configure persistence on a virtual server if you want to maintain the states of connections on the servers represented by that virtual server (for example, connections used in e-commerce). The appliance then uses the configured load balancing method for the initial selection of a server, but forwards to that same server all subsequent requests from the same client.

If persistence is configured, it overrides the load balancing methods once the server has been selected. If the configured persistence applies to a service that is down, the appliance uses the load balancing methods to select a new service, and the new service becomes persistent for subsequent requests from the client. If the selected service is in an Out Of Service state, it continues to serve the outstanding requests but does not accept new requests or connections. After the shutdown period elapses, the existing connections are closed. The following table lists the types of persistence that you can configure.

Table 1. Limitations on Number of Simultaneous Persistent Connections

Persistence Type	Persistent Connections
Source IP, SSL Session ID, Rule, DESTIP, SRCIPDESTIP	250K
CookieInsert, URL passive, Custom Server ID	Memory limit. In case of CookieInsert, if time out is not 0, any number of connections is allowed until limited by memory.

If the configured persistence cannot be maintained because of a lack of resources on an appliance, the load balancing methods are used for server selection. Persistence is maintained for a configured period of time, depending on the persistence type. Some persistence types are specific to certain virtual servers. The following table shows the relationship.

Table 2. Persistence Types Available for Each Type of Virtual Server

Persistence TypeHeader 1	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge
Source IP	YES	YES	YES	YES	YES
CookieInsert	YES	YES	NO	NO	NO
SSL Session ID	NO	YES	NO	NO	YES
URL Passive	YES	YES	NO	NO	NO
Custom Server ID	YES	YES	NO	NO	NO
Rule	YES	YES	NO	NO	NO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Type	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge
SRCIPDESTIP	N/A	N/A	YES	YES	N/A
Header 1					
DESTIP	N/A	N/A	YES	YES	N/A

You can also specify persistence for a group of virtual servers. When you enable persistence on the group, the client requests are directed to the same selected server regardless of which virtual server in the group receives the client request. When the configured time for persistence elapses, any virtual server in the group can be selected for incoming client requests.

Two commonly used persistence types are persistence based on cookies and persistence based on server IDs in URLs.

Configuring Persistence Based on Cookies

Updated: 2013-08-23

When you enable persistence based on cookies, the NetScaler adds an HTTP cookie into the Set-Cookie header field of the HTTP response. The cookie contains information about the service to which the HTTP requests must be sent. The client stores the cookie and includes it in all subsequent requests, and the NetScaler uses it to select the service for those requests. You can use this type of persistence on virtual servers of type HTTP or HTTPS.

The NetScaler inserts the cookie <NSC_XXXX>= <ServiceIP> <ServicePort>

where:

- <NSC_XXXX> is the virtual server ID that is derived from the virtual server name.
- <ServiceIP> is the hexadecimal value of the IP address of the service.
- <ServicePort> is the hexadecimal value of the port of the service.

The NetScaler encrypts ServiceIP and ServicePort when it inserts a cookie, and decrypts them when it receives a cookie.

Note: If the client is not allowed to store the HTTP cookie, the subsequent requests do not have the HTTP cookie, and persistence is not honored.
By default, the NetScaler sends HTTP cookie version 0, in compliance with the Netscape specification. It can also send version 1, in compliance with RFC 2109.

You can configure a timeout value for persistence that is based on HTTP cookies. Note the following:

- If HTTP cookie version 0 is used, the NetScaler inserts the absolute Coordinated Universal Time (GMT) of the cookie's expiration (the expires attribute of the HTTP cookie), calculated as the sum of the current GMT time on a NetScaler, and the timeout value.
- If an HTTP cookie version 1 is used, the NetScaler inserts a relative expiration time (Max-Age attribute of the HTTP cookie). In this case, the client software calculates the actual expiration time.

Note: Most client software currently installed (Microsoft Internet Explorer and Netscape browsers) understand HTTP cookie version 0; however, some HTTP proxies understand HTTP cookie version 1.
If you set the timeout value to 0, the NetScaler does not specify the expiration time, regardless of the HTTP cookie version used. The expiration time then depends on the client software, and such cookies are not valid if that software is shut down. This persistence type does not consume any system resources. Therefore, it can accommodate an unlimited number of persistent clients.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

An administrator can use the procedure in the following table to change the HTTP cookie version.

To change the HTTP cookie version by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, click Change HTTP Parameters.
3. In the Configure HTTP Parameters dialog box, under Cookie, select Version 0 or Version 1.

Note: For information about the parameters, see "Configuring Persistence Based on Cookies."

To configure persistence based on cookies by using the command line interface

At the command prompt, type the following commands to configure persistence based on cookies and verify the configuration:

- set lb vserver <name> -persistenceType COOKIEINSERT
- show lb vserver <name>

  Example

```
> set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
 Done
> show lb vserver vserver-LB-1
   vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
   .
   .
   .
   Persistence: COOKIEINSERT (version 0)   Persistence Timeout: 2 min
   .
   .
   .
 Done
>
```

To configure persistence based on cookies by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure persistence (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Method and Persistence tab, in the Persistence list, select COOKIEINSERT.
4. In the Time-out (min) text box, type the time-out value (for example, 2).
5. Click OK.
6. Verify that the virtual server for which you configured persistence is correctly configured by selecting the virtual server and viewing the Details section at the bottom of the pane.

Configuring Persistence Based on Server IDs in URLs

Updated: 2013-08-23

The NetScaler can maintain persistence based on the server IDs in the URLs. In a technique called URL passive persistence, the NetScaler extracts the server ID from the server response and embeds it in the URL query of the client request. The

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

server ID is an IP address and port specified as a hexadecimal number. The NetScaler extracts the server ID from subsequent client requests and uses it to select the server.

URL passive persistence requires configuring either a payload expression or a policy infrastructure expression specifying the location of the server ID in the client requests. For more information about expressions, see "Policy Configuration and Reference."

Note: If the server ID cannot be extracted from the client requests, server selection is based on the load balancing method.

Example: Payload Expression

The expression, URLQUERY contains sid= configures the system to extract the server ID from the URL query of a client request, after matching token sid=. Thus, a request with the URL http://www.citrix.com/index.asp?&sid;=c0a864100050 is directed to the server with the IP address10.102.29.10 and port 80.

The timeout value does not affect this type of persistence, which is maintained as long as the server ID can be extracted from the client requests. This persistence type does not consume any system resources, so it can accommodate an unlimited number of persistent clients.

Note: For information about the parameters, see "Load Balancing."

To configure persistence based on server IDs in URLs by using the command line interface

At the command prompt, type the following commands to configure persistence based on server IDs in URLs and verify the configuration:

- set lb vserver <name> -persistenceType URLPASSIVE
- show lb vserver <name>

  Example

```
> set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
 Done
> show lb vserver vserver-LB-1
   vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
   .
   .
   .
   Persistence: URLPASSIVE Persistence Timeout: 2 min
   .
   .
   .
 Done
>
```

To configure persistence based on server IDs in URLs by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure persistence (for example, vserver-LB-1), and then click Open.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Configure Virtual Server (Load Balancing) dialog box, on the Method and Persistence tab, in the Persistence list, select URLPASSIVE.
4. In the Time-out (min) text box, type the time-out value (for example, 2).
5. In the Rule text box, enter a valid expression. Alternatively, click Configure next to the Rule text box and use the Create Expression dialog box to create an expression.
6. Click OK.
7. Verify that the virtual server for which you configured persistence is correctly configured by selecting the virtual server and viewing the Details section at the bottom of the pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Features to Protect the Load Balancing Configuration

Jun 24, 2013

You can configure URL redirection to provide notifications of virtual server malfunctions, and you can configure backup virtual servers to take over if a primary virtual server becomes unavailable.

Configuring URL Redirection

You can configure a redirect URL to communicate the status of the appliance in the event that a virtual server of type HTTP or HTTPS is down or disabled. This URL can be a local or remote link. The appliance uses HTTP 302 redirect.

Redirects can be absolute URLs or relative URLs. If the configured redirect URL contains an absolute URL, the HTTP redirect is sent to the configured location, regardless of the URL specified in the incoming HTTP request. If the configured redirect URL contains only the domain name (relative URL), the HTTP redirect is sent to a location after appending the incoming URL to the domain configured in the redirect URL.

Note: If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. In this case, a redirect is used when both the primary and backup virtual servers are down.

To configure a virtual server to redirect client requests to a URL by using the command line interface

At the command prompt, type the following commands to configure a virtual server to redirect client requests to a URL and verify the configuration:

- set lb vserver <name> -redirectURL <URL>
- show lb vserver <name>

  Example

```
> set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
 Done
> show lb vserver vserver-LB-1
   vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
   State: DOWN
   Last state change was at Wed Jun 17 08:56:34 2009 (+666 ms)
   .
   .
   .
   Redirect URL: http://www.newdomain.com/mysite/maintenance
   .
   .
   .
 Done
>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure a virtual server to redirect client requests to a URL by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure URL redirection (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Advanced tab, in the Redirect URL text box, type the URL (for example, http://www.newdomain.com/mysite/maintenance), and then click OK.
4. Verify that the redirect URL you configured for the server appears in the Details section at the bottom of the pane.

Configuring Backup Virtual Servers

If the primary virtual server is down or disabled, the appliance can direct the connections or client requests to a backup virtual server that forwards the client traffic to the services. The appliance can also send a notification message to the client regarding the site outage or maintenance. The backup virtual server is a proxy and is transparent to the client.

You can configure a backup virtual server when you create a virtual server or when you change the optional parameters of an existing virtual server. You can also configure a backup virtual server for an existing backup virtual server, thus creating a cascaded backup virtual server. The maximum depth of cascading backup virtual servers is 10. The appliance searches for a backup virtual server that is up and accesses that virtual server to deliver the content.

You can configure URL redirection on the primary for use when the primary and the backup virtual servers are down or have reached their thresholds for handling requests.

Note: If no backup virtual server exists, an error message appears, unless the virtual server is configured with a redirect URL. If both a backup virtual server and a redirect URL are configured, the backup virtual server takes precedence.

To configure a backup virtual server by using the command line interface

At the command prompt, type the following commands to configure a backup server and verify the configuration:

- set lb vserver <name> [-backupVserver <string>]
- show lb vserver <name>

  Example

```
> set lb vserver vserver-LB-1 -backupVserver vserver-LB-2
 Done
> show lb vserver vserver-LB-1
   vserver-LB-1 (10.102.29.60:80) - HTTP   Type: ADDRESS
   State: DOWN
   Last state change was at Wed Jun 17 08:56:34 2009 (+661 ms)
   .
   .
   .
   Backup: vserver-LB-2
   .
   .
   .
 Done
>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To set up a backup virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the backup virtual server (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Advanced tab, in the Backup Virtual Server list, select the backup virtual server (for example, vserver-LB-2, and then click OK.
4. Verify that the backup virtual server you configured appears in the Details section at the bottom of the pane.
   Note: If the primary server goes down and then comes back up, and you want the backup virtual server to function as the primary server until you explicitly reestablish the primary virtual server, select the Disable Primary When Down check box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

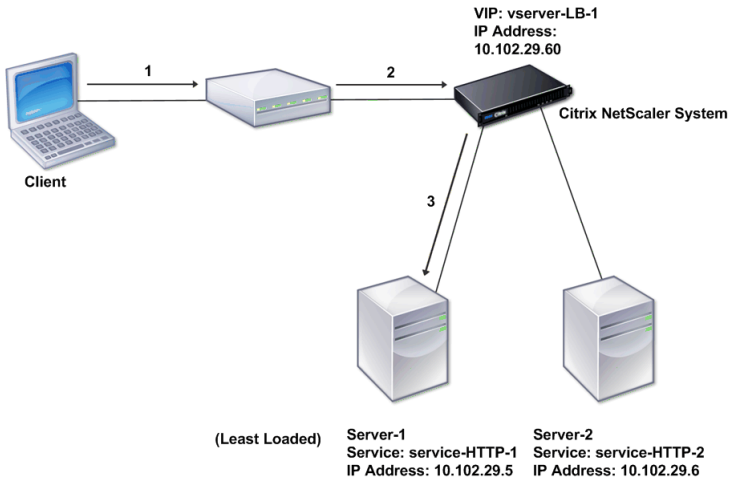
A Typical Load Balancing Scenario

Aug 23, 2013

In a load balancing setup, the NetScaler appliances are logically located between the client and the server farm, and they manage traffic flow to the servers.

The following figure shows the topology of a basic load balancing configuration.

Figure 1. Basic Load Balancing Topology



The virtual server selects the service and assigns it to serve client requests. Consider the scenario in the preceding figure, where the services service-HTTP-1 and service-HTTP-2 are created and bound to the virtual server named virtual server-LB-1. Virtual server-LB-1 forwards the client request to either service-HTTP-1 or service-HTTP-2. The system selects the service for each request by using the Least Connections load balancing method. The following table lists the names and values of the basic entities that must be configured on the system.

Table 1. LB Configuration Parameter Values

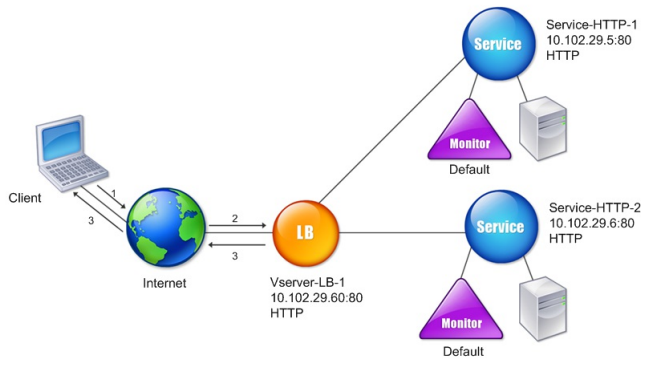
Entity Type	Required parameters and sample values			
	Name	**IP Address**	**Port**	**Protocol**
Virtual Server	vserver-LB-1	10.102.29.60	80	HTTP
Services	service-HTTP-1	10.102.29.5	8083	HTTP
	service-HTTP-2	10.102.29.6	80	HTTP
Monitors	Default	None	None	None

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity Type	Required parameters and sample values			
	Name	IP Address	Port	Protocol

The following figure shows the load balancing sample values and required parameters that are described in the preceding table.

Figure 2. Load Balancing Entity Model



The following tables list the commands used to configure this load balancing setup by using the command line interface.

Table 2. Initial Configuration Tasks

Task	Command
To enable load balancing	enable feature lb
To create a service named service-HTTP-1	add service service-HTTP-1 10.102.29.5 HTTP 80
To create a service named service-HTTP-2	add service service-HTTP-2 10.102.29.6 HTTP 80
To create a virtual server named vserver-LB-1	add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
To bind a service named service-HTTP-1 to a virtual server named vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-1
To bind a service named service-HTTP-2 to a virtual server named vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-2

For more information about the initial configuration tasks, see "Enabling Load Balancing" and "Configuring Services and a Vserver."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 3. Verification Tasks

Task	Command
To view the properties of a virtual server named vserver-LB-1	show lb vserver vserver-LB-1
To view the statistics of a virtual server named vserver-LB-1	stat lb vserver vserver-LB-1
To view the properties of a service named service-HTTP-1	show service service-HTTP-1
To view the statistics of a service named service-HTTP-1	stat service service-HTTP-1
To view the bindings of a service named service-HTTP-1	show service bindings service-HTTP-1

Table 4. Customization Tasks

Task	Command
To configure persistence on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType SOURCEIP -persistenceMask 255.255.255.255 -timeout 2
To configure COOKIEINSERT persistence on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
To configure URLPassive persistence on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
To configure a virtual server to redirect the client request to a URL on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
To set a backup virtual server on a virtual server named vserver-LB-1	set lb vserver vserver-LB-1 -backupVserver vserver-LB-2

For more information about configuring persistence, see "Choosing and Configuring Persistence Settings." For information about configuring a virtual server to redirect a client request to a URL and setting up a backup virtual server, see "Configuring Features to Protect the Load Balancing Configuration."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Accelerating Load Balanced Traffic by Using Compression

Aug 22, 2013

Compression is a popular means of optimizing bandwidth usage, and most web browsers support compressed data. If you enable the compression feature, the NetScaler appliance intercepts requests from clients and determines whether the client can accept compressed content. After receiving the HTTP response from the server, the appliance examines the content to determine whether it is compressible. If the content is compressible, the appliance compresses it, modifies the response header to indicate the type of compression performed, and forwards the compressed content to the client.

NetScaler compression is a policy-based feature. A policy filters requests and responses to identify responses to be compressed, and specifies the type of compression to apply to each response. The appliance provides several built-in policies to compress common MIME types such as text/html, text/plain, text/xml, text/css, text/rtf, application/msword, application/vnd.ms-excel, and application/vnd.ms-powerpoint. You can also create custom policies. The appliance does not compress compressed MIME types such as application/octet-stream, binary, bytes, and compressed image formats such as GIF and JPEG.

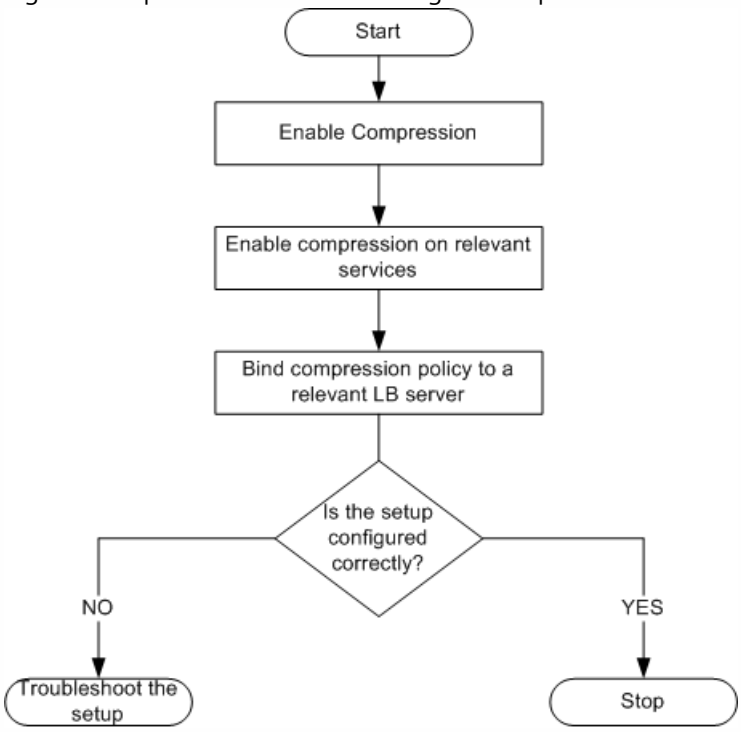
To configure compression, you must enable it globally and on each service that will provide responses that you want compressed. If you have configured virtual servers for load balancing or content switching, you should bind the polices to the virtual servers. Otherwise, the policies apply to all traffic that passes through the appliance.

Compression Configuration Task Sequence

Updated: 2013-08-22

The following flow chart shows the sequence of tasks for configuring basic compression in a load balancing setup.

Figure 1. Sequence of Tasks to Configure Compression



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: The steps in the above figure assume that load balancing has already been configured.

Enabling Compression

By default, compression is not enabled. You must enable the compression feature to allow compression of HTTP responses that are sent to the client.

To enable compression by using the command line interface

At the command prompt, type the following commands to enable compression and verify the configuration:

- enable ns feature CMP
- show ns feature

Example

```
> enable ns feature CMP
Done
> show ns feature

    Feature                Acronym        Status
    -------                -------        ------
1)  Web Logging            WL             ON
2)  Surge Protection       SP             OFF
.
7)  Compression Control    CMP            ON
8)  Priority Queuing       PQ             OFF
.
Done
```

To enable compression by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Compression check box, and then click OK.
4. In the Enable/Disable Feature(s)? dialog box, click Yes.

Configuring Services to Compress Data

Updated: 2013-08-22

In addition to enabling compression globally, you must enable it on each service that will deliver files to be compressed.

To enable compression on a service by using the command line

At the command prompt, type the following commands to enable compression on a service and verify the configuration:

- set service <name> -CMP YES
- show service <name>

Example

```
> show service SVC_HTTP1
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
SVC_HTTP1 (10.102.29.18:80) - HTTP
State: UP
Last state change was at Tue Jun 16 06:19:14 2009 (+737 ms)
Time since last state change: 0 days, 03:03:37.200
Server Name: 10.102.29.18
Server ID : 0   Monitor Threshold : 0
Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
Use Source IP: NO
Client Keepalive(CKA): NO
Access Down Service: NO
TCP Buffering(TCPB): NO
HTTP Compression(CMP): YES
Idle timeout: Client: 180 sec   Server: 360 sec
Client IP: DISABLED
Cacheable: NO
SC: OFF
SP: OFF
Down state flush: ENABLED

1)      Monitor Name: tcp-default
State: DOWN     Weight: 1
Probes: 1095    Failed [Total: 1095 Current: 1095]
Last response: Failure - TCP syn sent, reset received.
Response Time: N/A
Done
```

To enable compression on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, select the service for which you want to configure compression (for example, service-HTTP-1), and then click Open.
3. On the Advanced tab, under Settings, select the Compression check box, and then click OK.
4. Verify that, when the service is selected, HTTP Compression(CMP): ON appears in the **Details** section at the bottom of the pane.

Binding a Compression Policy to a Virtual Server

Updated: 2013-09-04

If you bind a policy to a virtual server, the policy is evaluated only by the services associated with that virtual server. You can bind compression policies to a virtual server either from the Configure Virtual Server (Load Balancing) dialog box or from the Compression Policy Manager dialog box. This topic includes instructions to bind compression policies to a load balancing virtual server by using the Configure Virtual Server (Load Balancing) dialog box. For information about how you can bind a compression policy to a load balancing virtual server by using the Compression Policy Manager dialog box, see "Configuring and Binding Policies with the Policy Manager."

To bind or unbind a compression policy to a virtual server by using the command line

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to bind or unbind a compression policy to a load balancing virtual server and verify the configuration:

- (bind | unbind) lb vserver <name> -policyName <string>
- show lb vserver <name>

Example

```
> bind lb vserver lbvip -policyName ns_cmp_msapp
Done
> show lb vserver lbvip
lbvip (8.7.6.6:80) - HTTP      Type: ADDRESS
State: UP
Last state change was at Thu May 28 05:37:21 2009 (+685 ms)
Time since last state change: 19 days, 04:26:50.470
Effective State: UP
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Port Rewrite : DISABLED
No. of Bound Services :  1 (Total)      1 (Active)
Configured Method: LEASTCONNECTION
Current Method: Round Robin, Reason: Bound service's state changed to UP
Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED  Push VServer:
Push Multi Clients: NO
Push Label Rule:

Bound Service Groups:
1)     Group Name: Service-Group-1

1) Service-Group-1 (10.102.29.252: 80) - HTTP State: UP Weight:                    1

1)     Policy : ns_cmp_msapp Priority:0
Done
```

To bind or unbind a compression policy to a load balancing virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server to which you want to bind or unbind a compression policy (for example, Vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Policies tab, click Compression.
4. Do one of the following:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- To bind a compression policy, click Insert Policy, and then select the policy you want to bind to the virtual server.
- To unbind a compression policy, click the name of the policy you want to unbind from the virtual server, and then click Unbind Policy.

5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Jumbo Frames on NetScaler VPX Appliances

Jan 31, 2011

NetScaler VPX appliances support receiving and transmitting jumbo frames containing up to 9216 bytes of IP data. Jumbo frames can transfer large files more efficiently than it is possible with the standard IP MTU size of 1500 bytes.

A NetScaler appliance can use jumbo frames in the following deployment scenarios:

- Jumbo to Jumbo. The appliance receives data as jumbo frames and sends it as jumbo frames.
- Non-Jumbo to Jumbo. The appliance receives data as regular frames and sends it as jumbo frames.
- Jumbo to Non-Jumbo. The appliance receives data as jumbo frames and sends it as regular frames.

Jumbo Frames support is available on NetScaler VPX appliances running on the following virtualization platforms:

- VMware ESX
- Linux-KVM Platform

Jumbo frames on VPX appliances works similar to Jumbo frames on MPX appliances. For more information on Jumbo Frames and its use cases, see Configuring Jumbo Frames on MPX appliances. The use cases of Jumbo Frames on MPX appliances also apply to VPX appliances.

Configuring Jumbo Frames for VPX running on VMware ESX

Perform the following tasks for configuring Jumbo Frames on a NetScaler VPX appliance running on VMware ESX server:

- Set the MTU of the interface or channel of the VPX appliance to a value in the range 1501-9000. Use the NetScaler command line interface or the configuration utility of the VPX appliance to set the MTU size. Note that NetScaler VPX appliances running on VMware ESX support receiving and transmitting jumbo frames containing up to only 9000 bytes of IP data.
- Set the same MTU size on the corresponding physical interfaces of the VMware ESX server by using its management applications. For more information about setting the MTU size on the physical interfaces of VMware ESX, see http://vmware.com/.

Configuring Jumbo Frames for VPX running on Linux-KVM Server

Perform the following tasks for configuring Jumbo Frames on a NetScaler VPX appliance running on a Linux-KVM Server:

- Set the MTU of the interface or channel of the VPX appliance to a value in the range 1501-9216. Use the NetScaler command line interface or the configuration utility of the VPX appliance to set the MTU size.
- Set the same MTU size on the corresponding physical interfaces of a Linux-KVM Server by using its management applications. For more information about setting the MTU size on the physical interfaces of Linux-KVM, see http://www.linux-kvm.org/.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Securing Load Balanced Traffic by Using SSL

Jan 31, 2011

The Citrix NetScaler SSL offload feature transparently improves the performance of web sites that conduct SSL transactions. By offloading CPU-intensive SSL encryption and decryption tasks from the local web server to the appliance, SSL offloading ensures secure delivery of web applications without the performance penalty incurred when the server processes the SSL data. Once the SSL traffic is decrypted, it can be processed by all standard services. The SSL protocol works seamlessly with various types of HTTP and TCP data and provides a secure channel for transactions using such data.

To configure SSL, you must first enable it. Then, you configure HTTP or TCP services and an SSL virtual server on the appliance, and bind the services to the virtual server. You must also add a certificate-key pair and bind it to the SSL virtual server. If you use Outlook Web Access servers, you must create an action to enable SSL support and a policy to apply the action. An SSL virtual server intercepts incoming encrypted traffic and decrypts it by using a negotiated algorithm. The SSL virtual server then forwards the decrypted data to the other entities on the appliance for appropriate processing.

This document includes the following:

- SSL Configuration Task Sequence
- Enabling SSL Offload
- Creating HTTP Services
- Adding an SSL-Based Virtual Server
- Binding Services to the SSL Virtual Server
- Adding a Certificate Key Pair
- Binding an SSL Certificate Key Pair to the Virtual Server
- Configuring Support for Outlook Web Access

SSL Configuration Task Sequence

To configure SSL, you must first enable it. Then, you must create an SSL virtual server and HTTP or TCP services on the NetScaler. Finally, you must bind a valid SSL certificate and the configured services to the SSL virtual server.

An SSL virtual server intercepts incoming encrypted traffic and decrypts it using a negotiated algorithm. The SSL virtual server then forwards the decrypted data to the other entities on the NetScaler for appropriate processing.

The following flow chart shows the sequence of tasks for configuring a basic SSL offload setup.

Figure 1. Sequence of Tasks to Configure SSL Offloading

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling SSL Offload

You should enable the SSL feature before configuring SSL offload. You can configure SSL-based entities on the appliance without enabling the SSL feature, but they will not work until you enable SSL.

To enable SSL by using the command line interface

At the command prompt, type the following commands to enable SSL Offload and verify the configuration:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- enable ns feature SSL
- show ns feature

Example

```
> enable ns feature ssl
Done
> show ns feature
Feature Acronym Status
------- ------- ------
1) Web Logging WL ON
2) SurgeProtection SP OFF
3) Load Balancing LB ON . . .
 9) SSL Offloading SSL ON
10) Global Server Load Balancing GSLB ON . .
Done >
```

To enable SSL by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Change basic features.
3. Select the SSL Offloading check box, and then click OK.
4. In the Enable/Disable Feature(s)? message box, click Yes.

Creating HTTP Services

Updated: 2013-08-23

A service on the appliance represents an application on a server. Once configured, services are in the disabled state until the appliance can reach the server on the network and monitor its status. This topic covers the steps to create an HTTP service.

Note: For TCP traffic, perform the procedures in this and the following topics, but create TCP services instead of HTTP services.

To add an HTTP service by using the command line interface

At the command prompt, type the following commands to add a HTTP service and verify the configuration:

- add service <name> (<IP> | <serverName>) <serviceType> <port>
- show service <name>

```
> add service SVC_HTTP1 10.102.29.18 HTTP 80
 Done
> show service SVC_HTTP1
    SVC_HTTP1 (10.102.29.18:80) - HTTP
    State: UP
    Last state change was at Wed Jul 15 06:13:05 2009
    Time since last state change: 0 days, 00:00:15.350
    Server Name: 10.102.29.18
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Server ID : 0   Monitor Threshold : 0
    Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
    TCP Buffering(TCPB): NO
    HTTP Compression(CMP): YES
    Idle timeout: Client: 180 sec   Server: 360 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: OFF
    Down state flush: ENABLED

1)    Monitor Name: tcp-default
        State: UP       Weight: 1
        Probes: 4       Failed [Total: 0 Current: 0]
        Last response: Success - TCP syn+ack received.
        Response Time: N/A
Done
```

To add an HTTP service by using the configuration utility

1. Navigate to Traffic Management > SSL Offload > Services.
2. In details pane, click Add.
3. In the Create Service dialog box, in the Service Name, Server, and Port text boxes, type the name of the service, IP address, and port (for example, SVC_HTTP1, 10.102.29.18, and 80).
4. In the Protocol list, select the type of the service (for example, HTTP).
5. Click Create, and then click Close. The HTTP service you configured appears in the Services page.
6. Verify that the parameters you configured are correctly configured by selecting the service and viewing the Details section at the bottom of the pane.

Adding an SSL-Based Virtual Server

In a basic SSL offloading setup, the SSL virtual server intercepts encrypted traffic, decrypts it, and sends the clear text messages to the services that are bound to the virtual server. Offloading CPU-intensive SSL processing to the appliance allows the back-end servers to process a greater number of requests.

To add an SSL-based virtual server by using the command line interface

At the command prompt, type the following commands to create an SSL-based virtual server and verify the configuration:

- add lb vserver <name> <serviceType> [<IPAddress> <port>]
- show lb vserver <name>

  Example

```
> add lb vserver vserver-SSL-1 SSL 10.102.29.50 443
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done
> show lb vserver vserver-SSL-1
vserver-SSL-1 (10.102.29.50:443) - SSL Type: ADDRESS
State: DOWN[Certkey not bound] Last state change was at Tue Jun 16 06:33:08 2009 (+176 ms)
Time since last state change: 0 days, 00:03:44.120
Effective State: DOWN Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
No. of Bound Services : 0 (Total) 0 (Active)
Configured Method: LEASTCONNECTION Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule: Done
```

Caution: To ensure secure connections, you must bind a valid SSL certificate to the SSL-based virtual server before you enable it.

To add an SSL-based virtual server by using the configuration utility

1. Navigate to Traffic Management > SSL Offload > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (SSL Offload) dialog box, in the Name, IP Address, and Port text boxes, type the name of the virtual server, IP address, and port (for example, Vserver-SSL-1, 10.102.29.50, and 443).
4. In the Protocol list, select the type of the virtual server, for example, SSL.
5. Click Create, and then click Close.
6. Verify that the parameters you configured are correctly configured by selecting the virtual server and viewing the Details section at the bottom of the pane. The virtual server is marked as DOWN because a certificate-key pair and services have not been bound to it.

Caution: To ensure secure connections, you must bind a valid SSL certificate to the SSL-based virtual server before you enable it.

Binding Services to the SSL Virtual Server

Updated: 2013-08-23

After decrypting the incoming data, the SSL virtual server forwards the data to the services that you have bound to the virtual server.

Data transfer between the appliance and the servers can be encrypted or in clear text. If the data transfer between the appliance and the servers is encrypted, the entire transaction is secure from end to end. For more information about configuring the system for end-to-end security, see "SSL Offload and Acceleration."

To bind a service to a virtual server by using the command line interface

At the command prompt, type the following commands to bind service to the SSL virtual server and verify the configuration:

- bind lb vserver <name> <serviceName>
- show lb vserver <name>

Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> bind lb vserver vserver-SSL-1 SVC_HTTP1
Done
> show lb vserver vserver-SSL-1 vserver-SSL-1 (10.102.29.50:443) - SSL Type:
ADDRESS State: DOWN[Certkey not bound]
Last state change was at Tue Jun 16 06:33:08 2009 (+174 ms)
Time since last state change: 0 days, 00:31:53.70
Effective State: DOWN Client Idle
Timeout: 180 sec
Down state flush: ENABLED Disable Primary Vserver On Down :
DISABLED No. of Bound Services : 1 (Total) 0 (Active)
Configured Method: LEASTCONNECTION Mode: IP Persistence: NONE Vserver IP and
Port insertion: OFF Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule:

1) SVC_HTTP1 (10.102.29.18: 80) - HTTP
State: DOWN Weight: 1
Done
```

To bind a service to a virtual server by using the configuration utility

1. Navigate to Traffic Management > SSL Offload > Virtual Servers.
2. In the details pane, select a virtual server, and then click Open.
3. On the Services tab, in the Active column, select the check boxes next to the services that you want to bind to the selected virtual server.
4. Click OK.
5. Verify that the Number of Bound Services counter in the Details section at the bottom of the pane is incremented by the number of services that you bound to the virtual server.

Adding a Certificate Key Pair

An SSL certificate is an integral element of the SSL Key-Exchange and encryption/decryption process. The certificate is used during SSL handshake to establish the identity of the SSL server. You can use a valid, existing SSL certificate that you have on the NetScaler appliance, or you can create your own SSL certificate. The appliance supports RSA/DSA certificates of up to 4096 bits.

Note: Citrix recommends that you use a valid SSL certificate that has been issued by a trusted certificate authority. Invalid certificates and self-created certificates are not compatible with all SSL clients.
Before a certificate can be used for SSL processing, you must pair it with its corresponding key. The certificate key pair is then bound to the virtual server and used for SSL processing.

To add a certificate key pair by using the command line interface

At the command prompt, type the following commands to create a certificate key pair and verify the configuration:

- add ssl certKey <certkeyName> -cert <string> [-key <string>]
- show sslcertkey <name>

  Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> add ssl certKey CertKey-SSL-1 -cert ns-root.cert -key ns-root.key
 Done
> show sslcertkey CertKey-SSL-1
  Name: CertKey-SSL-1 Status: Valid,
  Days to expiration:4811 Version: 3
  Serial Number: 00 Signature Algorithm: md5WithRSAEncryption Issuer: C=US,ST=California,L=San
  Jose,O=Citrix ANG,OU=NS Internal,CN=de fault
  Validity Not Before: Oct 6 06:52:07 2006 GMT Not After : Aug 17 21:26:47 2022 GMT
  Subject: C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,CN=d efault Public Key
  Algorithm: rsaEncryption Public Key
  size: 1024
 Done
```

To add a certificate key pair by using the configuration utility

1. Navigate to Traffic Management > SSL > Certificates.
2. In the details pane, click Add.
3. In the Install Certificate dialog box, in the Certificate-Key Pair Name text box, type a name for the certificate key pair you want to add, for example, Certkey-SSL-1.
4. Under Details, in Certificate File Name, click Browse (Appliance) to locate the certificate. Both the certificate and the key are stored in the /nsconfig/ssl/ folder on the appliance. To use a certificate present on the local system, select Local.
5. Select the certificate you want to use, and then click Select.
6. In Private Key File Name, click Browse (Appliance) to locate the private key file. To use a private key present on the local system, select Local.
7. Select the key you want to use and click Select. To encrypt the key used in the certificate key pair, type the password to be used for encryption in the Password text box.
8. Click Install.
9. Double-click the certificate key pair and, in the Certificate Details window, verify that the parameters have been configured correctly and saved.

Binding an SSL Certificate Key Pair to the Virtual Server

After you have paired an SSL certificate with its corresponding key, you must bind the certificate key pair to the SSL virtual server so that it can be used for SSL processing. Secure sessions require establishing a connection between the client computer and an SSL-based virtual server on the appliance. SSL processing is then carried out on the incoming traffic at the virtual server. Therefore, before enabling the SSL virtual server on the appliance, you need to bind a valid SSL certificate to the SSL virtual server.

To bind an SSL certificate key pair to a virtual server by using the command line interface

At the command prompt, type the following commands to bind an SSL certificate key pair to a virtual server and verify the configuration:

- bind ssl vserver <vServerName> -certkeyName <string>
- show ssl vserver <name>

  Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 > bind ssl vserver Vserver-SSL-1 -certkeyName CertKey-SSL-1
Done
> show ssl vserver Vserver-SSL-1

    Advanced SSL configuration for VServer Vserver-SSL-1:
    DH: DISABLED
    Ephemeral RSA: ENABLED Refresh Count: 0
    Session Reuse: ENABLED Timeout: 120 seconds
    Cipher Redirect: ENABLED
    SSLv2 Redirect: ENABLED
    ClearText Port: 0
    Client Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1) CertKey Name: CertKey-SSL-1 Server Certificate
1) Cipher Name: DEFAULT
  Description: Predefined Cipher Alias
Done
```

To bind an SSL certificate key pair to a virtual server by using the configuration utility

1. Navigate to Traffic Management > SSL Offload > Virtual Servers.
2. Select the virtual server to which you want to bind the certificate key pair, for example, Vserver-SSL-1, and click Open.
3. In the Configure Virtual Server (SSL Offload) dialog box, on the SSL Settings tab, under Available, select the certificate key pair that you want to bind to the virtual server (for example, Certkey-SSL-1), and then click Add.
4. Click OK.
5. Verify that the certificate key pair that you selected appears in the Configured area.

Configuring Support for Outlook Web Access

If you use Outlook Web Access (OWA) servers on your NetScaler appliance, you must configure the appliance to insert a special header field, FRONT-END-HTTPS: ON, in HTTP requests directed to the OWA servers, so that the servers generate URL links as https:// instead of http://.

Note: You can enable OWA support for HTTP-based SSL virtual servers and services only. You cannot apply it for TCP-based SSL virtual servers and services.
To configure OWA support, do the following:

- Create an SSL action to enable OWA support.
- Create an SSL policy.
- Bind the policy to the SSL virtual server.

Creating an SSL Action to Enable OWA Support

Before you can enable Outlook Web Access (OWA) support, you must create an SSL action. SSL actions are bound to SSL

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

policies and triggered when incoming data matches the rule specified by the policy.

To create an SSL action to enable OWA support by using the command line interface

At the command prompt, type the following commands to create an SSL action to enable OWA support and verify the configuration:

- add ssl action <name> -OWASupport ENABLED
- show SSL action <name>

```
> add ssl action Action-SSL-OWA -OWASupport enabled
Done
> show SSL action Action-SSL-OWA
  Name: Action-SSL-OWA
  Data Insertion Action: OWA
  Support: ENABLED
Done
```

To create an SSL action to enable OWA support by using the configuration utility

1. Navigate to Traffic Management > SSL > Policies.
2. In the details pane, on the Actions tab, click Add.
3. In the Create SSL Action dialog box, in the Name text box, type Action-SSL-OWA.
4. Under Outlook Web Access, select Enabled.
5. Click Create, and then click Close.
6. Verify that Action-SSL-OWA appears in the **SSL Actions** page.

Creating SSL Policies

Updated: 2013-09-04

SSL policies are created by using the policy infrastructure. Each SSL policy has an SSL action bound to it, and the action is carried out when incoming traffic matches the rule that has been configured in the policy.

To create an SSL policy by using the command line interface

At the command prompt, type the following commands to configure an SSL policy and verify the configuration:

- add ssl policy <name> -rule <expression> -reqAction <string>
- show ssl policy <name>

Example

```
> add ssl policy Policy-SSL-1 -rule ns_true -reqaction Action-SSL-OWA
Done
> show ssl policy Policy-SSL-1
Name: Policy-SSL-1      Rule: ns_true
Action: Action-SSL-OWA  Hits: 0
Policy is bound to following entities
1)     PRIORITY : 0
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To create an SSL policy by using the configuration utility

1. Navigate to Traffic Management > SSL > Policies.
2. In the details pane, click Add.
3. In the Create SSL Policy dialog box, in the Name text box, type the name of the SSL Policy (for example, Policy-SSL-1).
4. In Request Action, select the configured SSL action that you want to associate with this policy (for example, Action-SSL-OWA). The ns_true general expression applies the policy to all successful SSL handshake traffic. However, if you need to filter specific responses, you can create policies with a higher level of detail. For more information about configuring granular policy expressions, see "Understanding Policies and Expressions."
5. In Named Expressions, choose the built-in general expression ns_true and click Add Expression. The expression ns_true now appears in the Expression text box.
6. Click Create, and then click Close.
7. Verify that the policy is correctly configured by selecting the policy and viewing the Details section at the bottom of the pane.

Binding the SSL Policy to an SSL Virtual Server

After you configure an SSL policy for Outlook Web Access, bind the policy to a virtual server that will intercept incoming Outlook traffic. If the incoming data matches any of the rules configured in the SSL policy, the policy is triggered and the action associated with it is carried out.

To bind an SSL policy to an SSL virtual server by using the command line interface

At the command prompt, type the following commands to bind an SSL policy to an SSL virtual server and verify the configuration:

- bind ssl vserver <vServerName> -policyName <string>
- show ssl vserver <name>

Example

```
> bind ssl vserver Vserver-SSL-1 -policyName Policy-SSL-1
 Done
> show ssl vserver Vserver-SSL-1
      Advanced SSL configuration for VServer Vserver-SSL-1:
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 0
      Session Reuse: ENABLED        Timeout: 120 seconds
      Cipher Redirect: ENABLED
      SSLv2 Redirect: ENABLED
      ClearText Port: 0
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

  1)    CertKey Name: CertKey-SSL-1 Server Certificate

  1)    Policy Name: Policy-SSL-1
                  Priority: 0
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
1)     Cipher Name: DEFAULT
       Description: Predefined Cipher Alias
 Done
>
```

To bind an SSL policy to an SSL virtual server by using the configuration utility

1. Navigate to Traffic Management > SSL Offload > Virtual Servers.
2. In the details pane, select the virtual server (for example, Vserver-SSL-1), and then click Open.
3. In the Configure Virtual Server (SSL Offload) dialog box, click Insert Policy, and then select the policy that you want to bind to the SSL virtual server. Optionally, you can double-click the Priority field and type a new priority level.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting Up vPath on NetScaler VPX

Jul 05, 2016

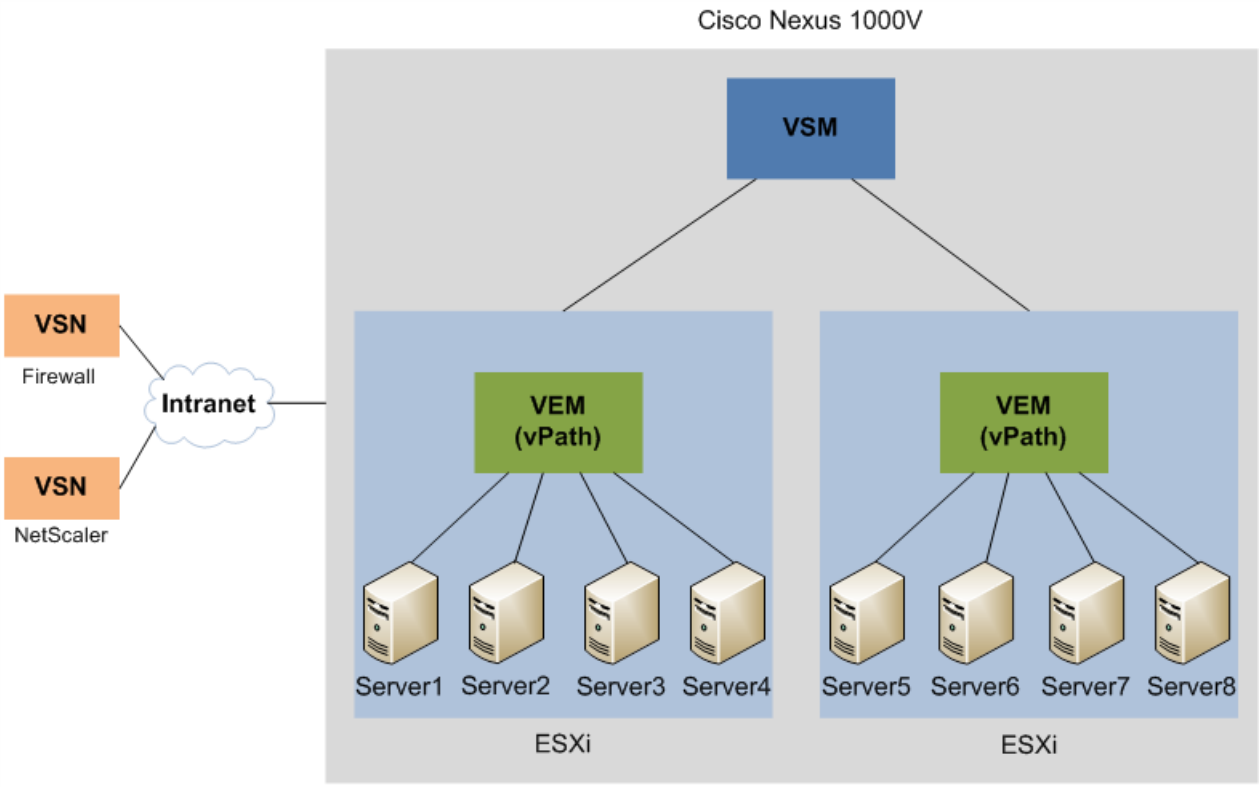
After installing the NetScaler virtual appliance, you must set it up to communicate with the servers.

In a NetScaler deployment, the virtual appliance communicates with servers through the Virtual Ethernet Modules (VEMs). A VEM can only interpret packets that are encapsulated with vPath service intelligence. Therefore, you must set up the virtual appliance to apply vPath encapsulation to all packets that are being sent to the server.

> **Note**
>
> The vPath feature in no longer available in NetScaler version 11.1.

vPath uses overlay tunnels to steer traffic to a VSN (for example, a NetScaler virtual appliance), which can be either Layer 2 or Layer 3 adjacent. For detailed information on vPath, see "Cisco vPath and vServices Reference Guide for VMware vSphere."

Figure 1. NetScaler 1000V with Nexus 1000V



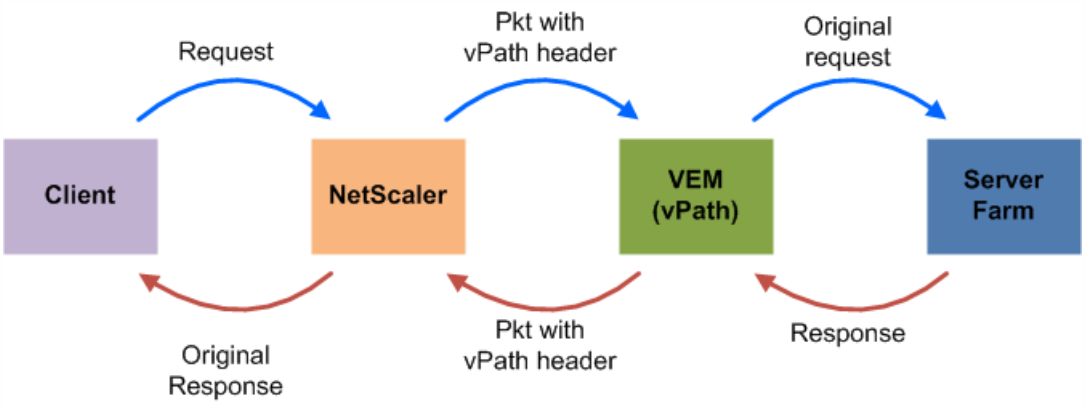
VSM – Virtual Supervisor Module, VEM – Virtual Ethernet Module, VSN – Virtual Service Node

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How vPath Works

Oct 06, 2014

Figure 1. Packet Flow Using vPath



The above figure illustrates the flow of traffic using vPath:

1. Client sends request to the NetScaler virtual appliance.
2. The NetScaler virtual appliance encapsulates the client request with a vPath header and sends the updated packet to the server that is selected by the load balancing algorithm.
3. The VEM (in which the vPath module is embedded) intercepts and decapsulates the packet and forwards it to the server.
4. Server responds with the required information.
5. The VEM encapsulates the server response with a vPath header and forwards the packet to the NetScaler virtual appliance.
6. The NetScaler virtual appliance decapsulates the packet and sends the response to the client.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Step 1: Configuring vPath on a NetScaler

Oct 06, 2014

All data transmitted between the NetScaler virtual appliance and the server is vPath encapsulated. By default, vPath is disabled on the NetScaler virtual appliance. Therefore, to configure vPath on a NetScaler, you must first enable vPath, and then configure a SNIP address as the source of the vPath packet when the packet is forwarded to the switch.

If, in the return flow, the vPath packet is received at an IP address other than the specified SNIP address, the appliance drops the packet.

To configure vPath on a NetScaler by using the command line interface

At the command prompt, do the following:

1. Enable vPath on the NetScaler virtual appliance.
   enable ns feature vPath

2. Specify the SNIP address to be used as the source IP address of the vPath packet. You can also specify whether the NetScaler must offload to the VEM, sessions for which the NetScaler has no matching configurations and hence not interested in.
   set vPathParam -srcIP <ip_addr> -offload <ENABLED | DISABLED>

   Note:
   - When the offload parameter is enabled, the NetScaler adds an extra 24 bytes to the vPath header.
   - By default, the NetScaler IP (NSIP) address is configured as the vPath source IP address. However, the show vPathParam command shows the source IP address as 0.0.0.0.

3. If you have a server that is not configured as a service on the NetScaler, you must explicitly enable vPath encapsulation as follows:
   add vpath <name> (<destIP> [<netmask>] [<gateway>]) -encapMode L3

4. Save the configurations.
   save ns config

To configure vPath on a NetScaler by using the graphical user interface

1. Navigate to Configuration > System > Settings.
2. In the details pane, under Modes and Features, click Configure advanced features and select the vPath checkbox.
3. Navigate to Configuration > System > Network.
4. In the details pane, under Settings, click Configure VPath Parameters and select the appropriate SNIP address as the source address.
5. To enable vPath encapsulation on a server that is not configured on the NetScaler, navigate to Configuration > System > Network > vPath.
6. In the details pane, click Add and provide the required details.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Step 2: Configuring Load Balancing of Backend Servers

Oct 06, 2014

When deployed in front of application servers, NetScaler ensures optimal distribution of traffic by the way in which it directs client requests. Administrators can segment application traffic according to information in the body of an HTTP or TCP request, and on the basis of L4-L7 header information such as URL, application data type, or cookie.

Numerous load balancing algorithms and extensive server health checks improve application availability by ensuring that client requests are directed to the appropriate servers.

To configure load balancing of servers, do the following:

1. Enable the load balancing feature and the use source IP (USIP) mode of the NetScaler.
   Navigate to Configuration > System > Settings and under Modes and Features do the following:

   1. Click Configure basic features and select the Load Balancing checkbox.
   2. Click Configure modes and select the Use Source IP checkbox.
      Note: With vPath integration, Source NAT is not required and server return traffic is redirected to NetScaler by vPath service attached to the server VM port. The original source IP is preserved for all connections.
2. Add the required servers as services on the NetScaler .
   Navigate to Configuration > Traffic Management > Load Balancing > Services, click Add and configure the details (IP address, port, protocol) of each of the servers as services on the NetScaler .

   Note: NetScaler is tightly integrated with the vPath architecture, and will not work without a vPath port-profile attached to the servers. Therefore, till the port profile configuration (provided in step 3) is done, the service state may appear as **Down**.
3. Create a virtual server that will bind these services to the virtual server IP address.
   Navigate to Configuration > Traffic Management > Load Balancing > Virtual Servers, click Add and configure the name, virtual IP address (VIP), protocol, load balancing method, and the services to be bound to the virtual server.

4. Save the configurations. Click Save in the upper right hand corner of the interface.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Step 3: Binding Backend Servers to a Port Profile

Oct 06, 2014

After performing the vPath configurations on the NetScaler and then configuring the load balancing virtual server, you must define the NetScaler as a Virtual Service Node (VSN) and associate it with a port profile. The port profile, which is defined on the Virtual Supervisor Module (VSM), specifies that all traffic reaching the virtual port of the server must be redirected to the NetScaler virtual appliance. On the vCenter Server, you must then bind the port profile to the virtual port that is associated with the virtual machine.

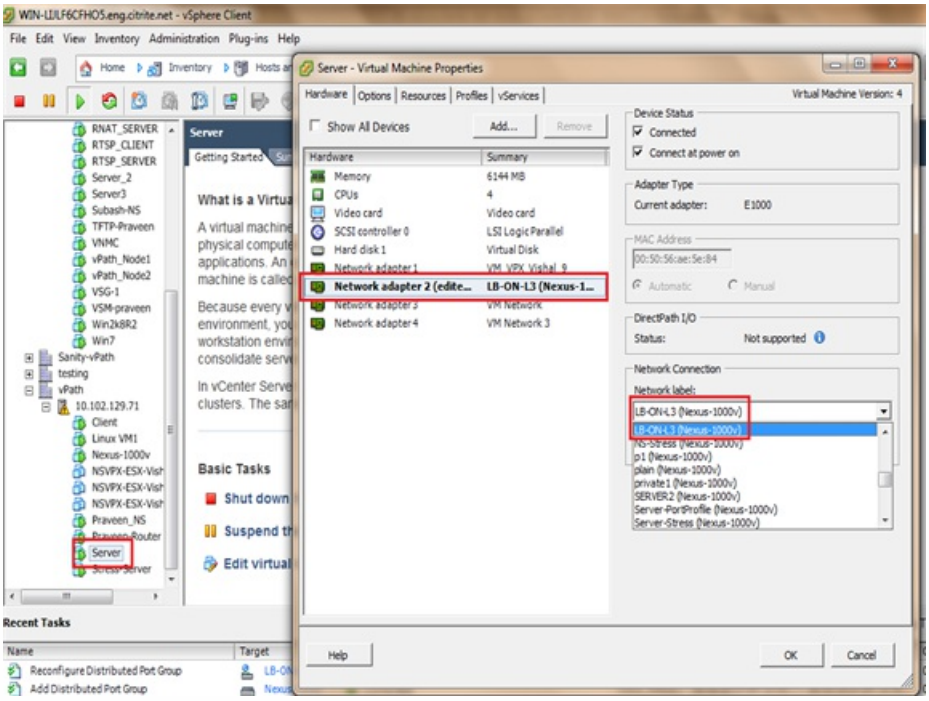
Note: Every virtual NIC of a virtual machine has a corresponding virtual port on the Nexus virtual switch. Each virtual port must be associated with a port profile that specifies the properties of the device.

To bind backend servers to a port profile

On the Nexus Virtual Supervisor Module (VSM), do the following

1. Configure the NetScaler virtual appliance as a VSN. **Example:** Create a VSN named "NS1" for a NetScaler with IP address 10.102.38.220.
   # vservice node **NS1** type **adc**
   ip address **10.102.38.220**
   adjacency l3
   fail-mode open
2. Create a port profile for the NetScaler virtual appliance. **Example:** Create a port profile named "LB-ON-L3" to be used for the NetScaler services.
   # port-profile type vethernet **LB-ON-L3**
   vmware port-group
   switchport mode access
   switchport access vlan 1
   vservice node **NS1**
   no shutdown
   system vlan 1
   state enabled
3. On the vCenter Server, bind the port profile to the virtual machine as shown in the following image:

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Note: Repeat this step to bind the required servers to the port profile.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Behavioral Aspects of NetScaler with vPath

Oct 06, 2014

Some points to note in a NetScaler deployment with vPath configured:

- The maximum value for the Maximum Segment Size (MSS) of the default TCP profile (nstcp_default_profile) is 1380.
- The MSS used by services and virtual servers is determined as follows:
  - A service uses the MSS configured for the default TCP profile (nstcp_default_profile) regardless of the MSS of the TCP profile that is bound to the service.
  - A virtual server uses the MSS that is the lower of the MSS defined for the default TCP profile (nstcp_default_profile) and the TCP profile that is bound to the virtual server.
- Supports pre-fragmentation of vPath encapsulated packets. Even packets with Do not Fragment (DF) bit set are pre-fragmented.
- When encapsulating a full-size packet with vPath information, if the packet exceeds the MTU, then, if the icmpErrGenerate parameter is set to ENABLED, the NetScaler generates an ICMP (Type 3,code 4) fragment needed error message.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Features at a Glance

Sep 04, 2013

Citrix NetScaler features can be configured independently or in combinations to address specific needs. Although some features fit more than one category, the numerous NetScaler features can generally be categorized as application switching and traffic management features, application acceleration features, and application security and firewall features, and an application visibility feature.

To understand the order in which the features perform their processing, see "Processing Order of Features."

This document includes the following:

- Application Switching and Traffic Management Features
- Application Acceleration Features
- Application Security and Firewall Features
- Application Visibility Feature
- Cloud Integration Feature

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Switching and Traffic Management Features

Aug 31, 2016

SSL Offloading

Transparently offloads SSL encryption and decryption from web servers, freeing server resources to service content requests. SSL places a heavy burden on an application's performance and can render many optimization measures ineffective. SSL offload and acceleration allow all the benefits of Citrix Request Switching technology to be applied to SSL traffic, ensuring secure delivery of web applications without degrading end-user performance.
For more information, see "SSL Offload and Acceleration."

Access Control Lists

Compares incoming packets to Access Control Lists (ACLs). If a packet matches an ACL rule, the action specified in the rule is applied to the packet. Otherwise, the default action (ALLOW) is applied and the packet is processed normally. For the appliance to compare incoming packets to the ACLs, you have to apply the ACLs. All ACLs are enabled by default, but you have to apply them in order for the NetScaler to compare incoming packets against them. If an ACL is not required to be a part of the lookup table, but still needs to be retained in the configuration, it should be disabled before the ACLs are applied. A NetScaler does not compare incoming packets to disabled ACLs.
For more information, see "Access Control List."

Load Balancing

Load balancing decisions are based on a variety of algorithms, including round robin, least connections, weighted least bandwidth, weighted least packets, minimum response time, and hashing based on URL, domain source IP, or destination IP. Both the TCP and UDP protocols are supported, so the NetScaler can load balance all traffic that uses those protocols as the underlying carrier (for example, HTTP, HTTPS, UDP, DNS, NNTP, and general firewall traffic). In addition, the NetScaler can maintain session persistence based on source IP, cookie, server, group, or SSL session. It allows users to apply custom Extended Content Verification (ECV) to servers, caches, firewalls and other infrastructure devices to ensure that these systems are functioning properly and are providing the right content to users. It can also perform health checks using ping, TCP, or HTTP URL, and the user can create monitors based on Perl scripts. To provide high-scale WAN optimization, the CloudBridge appliances deployed at data centers can be load balanced through NetScaler appliances. The bandwidth and number of concurrent sessions can be improved significantly.
For more information, see "Load Balancing."

Traffic Domains

Traffic domains provide a way to create logical ADC partitions within a single NetScaler appliance. They enable you to segment network traffic for different applications. You can use traffic domains to create multiple isolated environments whose resources do not interact with each other. An application belonging to a specific traffic domain communicates only with entities, and processes traffic, within that domain. Traffic belonging to one traffic domain cannot cross the boundary of another traffic domain. Therefore, you can use duplicate IP addresses on the appliance as long as an addresses is not duplicated within the same domain.
For more information, see "Traffic Domains."

Network Address Translation

Network address translation (NAT) involves modification of the source and/or destination IP addresses, and/or the TCP/UDP port numbers, of IP packets that pass through the NetScaler appliance. Enabling NAT on the appliance enhances

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the security of your private network, and protects it from a public network such as the Internet, by modifying your network's source IP addresses when data passes through the NetScaler.

The NetScaler appliance supports the following types of network address translation:

INAT—In Inbound NAT (INAT), an IP address (usually public) configured on the NetScaler appliance listens to connection requests on behalf of a server. For a request packet received by the appliance on a public IP address, the NetScaler replaces the destination IP address with the private IP address of the server. In other words, the appliance acts as a proxy between clients and the server. INAT configuration involves INAT rules, which define a 1:1 relationship between the IP address on the NetScaler appliance and the IP address of the server.

RNAT—In Reverse Network Address Translation (RNAT), for a session initiated by a server, the NetScaler appliance replaces the source IP address in the packets generated by the server with an IP address (type SNIP) configured on the appliance. The appliance thereby prevents exposure of the server's IP address in any of the packets generated by the server. An RNAT configuration involves an RNAT rule, which specifies a condition. The appliance performs RNAT processing on those packets that match the condition.

Stateless NAT46 Translation—Stateless NAT46 enables communication between IPv4 and IPv6 networks, by way of IPv4 to IPv6 packet translation and vice versa, without maintaining any session information on the NetScaler appliance. A stateless NAT46 configuration involves an IPv4-IPv6 INAT rule and an NAT46 IPv6 prefix.

Stateful NAT64 Translation—The stateful NAT64 feature enables communication between IPv4 clients and IPv6 servers through IPv6 to IPv4 packet translation, and vice versa, while maintaining session information on the NetScaler appliance. A stateful NAT64 configuration involves an NAT64 rule and an NAT64 IPv6 prefix.

For more information, see "Configuring Network Address Translation."

Multipath TCP Support

NetScaler appliances support Multipath TCP (MPTCP). MPTCP is a TCP/IP protocol extension that identifies and uses multiple paths available between hosts to maintain the TCP session. You must enable MPTCP on a TCP profile and bind it to a virtual server. When MPTCP is enabled, the virtual server functions as an MPTCP gateway and converts MPTCP connections with the clients to TCP connections that it maintains with the servers.
For more information, see "MPTCP (Multi-Path TCP)."

Content Switching

Determines the server to which to send the request on the basis of configured content switching policies. Policy rules can be based on the IP address, URL, and HTTP headers. This allows switching decisions to be based on user and device characteristics such as who the user is, what type of agent is being used, and what content the user requested.
For more information, see "Content Switching."

Global Server Load Balancing (GSLB)

Extends the traffic management capabilities of a NetScaler to include distributed Internet sites and global enterprises. Whether installations are spread across multiple network locations or multiple clusters in a single location, the NetScaler maintains availability and distributes traffic across them. It makes intelligent DNS decisions to prevent users from being sent to a site that is down or overloaded. When the proximity-based GSLB method is enabled, the NetScaler can make load balancing decisions based on the proximity of the client's local DNS server (LDNS) in relation to different sites. The main benefit of the proximity-based GSLB method is faster response time resulting from the selection of the closest available site.
For more information, see "Global Server Load Balancing."

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Dynamic Routing

Enables routers to obtain topology information, routes, and IP addresses from neighboring routers automatically. When dynamic routing is enabled, the corresponding routing process listens to route updates and advertises routes. The routing processes can also be placed in passive mode. Routing protocols enable an upstream router to load balance traffic to identical virtual servers hosted on two standalone NetScaler units using the Equal Cost Multipath technique.
For more information, see "Configuring Dynamic Routes."

Link Load Balancing

Load balances multiple WAN links and provides link failover, further optimizing network performance and ensuring business continuity. Ensures that network connections remain highly available, by applying intelligent traffic control and health checks to distribute traffic efficiently across upstream routers. Identifies the best WAN link to route both incoming and outbound traffic based on policies and network conditions, and protects applications against WAN or Internet link failure by providing rapid fault detection and failover.
For more information, see " Link Load Balancing."

TCP Optimization

You can use TCP profiles to optimize TCP traffic. TCP profiles define the way that NetScaler virtual servers process TCP traffic. Administrators can use the built-in TCP profiles or configure custom profiles. After defining a TCP profile, you can bind it to a single virtual server or to multiple virtual servers.
Some of the key optimization features that can be enabled by TCP profiles are:

- TCP keep-alive—Checks the operational status of the peers at specified time intervals to prevent the link from being broken.
- Selective Acknowledgment (SACK)— Improves the performance of data transmission, especially in long fat networks (LFNs).
- TCP window scaling— Allows efficient transfer of data over long fat networks (LFNs).

For more information on TCP Profiles, see "Configuring TCP Profiles."

Web Interface on NetScaler

Provides access to XenApp and XenDesktop resources, which include applications, content, and desktops. Users access resources through a standard Web browser or by using the Citrix XenApp plug-in. The Web Interface runs as a service on port 8080 on the NetScaler appliance. To create Web Interface sites, Java is executed on Apache Tomcat Web server version 6.0.26 on the NetScaler appliance.
Note: Web Interface is supported only on NetScaler nCore releases.
For more information, see "Web Interface."

CloudBridge Connector

The Citrix NetScaler CloudBridge Connector feature, a fundamental part of the Citrix OpenCloud framework, is a tool used to build a cloud-extended data center. The OpenCloud Bridge enables you to connect one or more NetScaler appliances or NetScaler virtual appliances on the cloud-to your network without reconfiguring your network. Cloud hosted applications appear as though they are running on one contiguous enterprise network. The primary purpose of the OpenCloud Bridge is to enable companies to move their applications to the cloud while reducing costs and the risk of application failure. In addition, the OpenCloud Bridge increases network security in cloud environments. An OpenCloud Bridge is a Layer-2 network bridge that connects a NetScaler appliance or NetScaler virtual appliance on a cloud instance to a NetScaler appliance or NetScaler virtual appliance on your LAN. The connection is made through a tunnel that uses the Generic Routing Encapsulation (GRE) protocol. The GRE protocol provides a mechanism for encapsulating packets from a wide variety of network protocols to be forwarded over another protocol. Then Internet Protocol security (IPsec) protocol suite is used to secure the communication between the peers in the OpenCloud Bridge.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For more information, see "CloudBridge."

DataStream

The NetScaler DataStream feature provides an intelligent mechanism for request switching at the database layer by distributing requests on the basis of the SQL query being sent.

When deployed in front of database servers, a NetScaler ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and on the basis of database names, user names, character sets, and packet size.

You can configure load balancing to switch requests according to load balancing algorithms, or you can elaborate the switching criteria by configuring content switching to make a decision based on SQL query parameters, such as user name, database names, and command parameters. You can further configure monitors to track the states of database servers.

The advanced policy infrastructure on the NetScaler appliance includes expressions that you can use to evaluate and process the requests. The advanced expressions evaluate traffic associated with MySQL database servers. You can use request-based expressions (expressions that begin with MYSQL.CLIENT and MYSQL.REQ) in advanced policies to make request switching decisions at the content switching virtual server bind point and response-based expressions (expressions that begin with MYSQL.RES) to evaluate server responses to user-configured health monitors.

Note: DataStream is supported for MySQL and MS SQL databases.
For more information, see "DataStream."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Acceleration Features

Sep 06, 2013

AppCompress

Uses the gzip compression protocol to provide transparent compression for HTML and text files. The typical 4:1 compression ratio yields up to 50% reduction in bandwidth requirements out of the data center. It also results in significantly improved end-user response time, because it reduces the amount of data that must be delivered to the user's browser.

For more information, see "Compression."

Cache Redirection

Manages the flow of traffic to a reverse proxy, transparent proxy, or forward proxy cache farm. Inspects all requests, and identifies non-cacheable requests and sends them directly to the origin servers over persistent connections. By intelligently redirecting non-cacheable requests back to the origin web servers, the NetScaler appliance frees cache resources and increases cache hit rates while reducing overall bandwidth consumption and response delays for these requests.

For more information, see "Cache Redirection."

AppCache

Helps optimize web content and application data delivery by providing a fast in-memory HTTP/1.1 and HTTP/1.0 compliant web caching for both static and dynamic content. This on-board cache stores the results of incoming application requests even when an incoming request is secured or the data compressed, and then reuses the data to fulfill subsequent requests for the same information. By serving data directly from the on-board cache, the appliance can reduce page regeneration times by eliminating the need to funnel static and dynamic content requests to the server.

For more information, see "Integrated Caching."

TCP Buffering

Buffers the server's response and delivers it to the client at the client's speed, thus offloading the server faster and thereby improving the performance of web sites.

For more information, see "TCP Buffering."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Security and Firewall Features

Oct 30, 2013

Denial of Service Attack (DoS) Defense

Detects and stops malicious distributed denial-of-service (DDoS) attacks and other types of malicious attacks before they reach your servers, preventing them from affecting network and application performance. The NetScaler appliance identifies legitimate clients and elevates their priority, leaving suspect clients unable to consume a disproportionate percentage of resources and cripple your site. The appliance provides application-level protection from the following types of malicious attacks:

- SYN flood attacks
- Pipeline attacks
- Teardrop attacks
- Land attacks
- Fraggle attacks
- Zombie connection attacks

The appliance aggressively defends against these types of attacks by preventing the allocation of server resources for these connections. This insulates servers from the overwhelming flood of packets associated with these events.

The appliance also protects network resources from ICMP based attacks by using ICMP rate limiting and aggressive ICMP packet inspection. It performs strong IP reassembly, drops a variety of suspicious and malformed packets, and applies Access Control Lists (ACLs) to site traffic for further protection.

For more information, see "HTTP Denial-of-Service Protection."

Content Filtering

Provides protection from malicious attacks for web sites at the Layer 7 level. The appliance inspects each incoming request according to user-configured rules based on HTTP headers, and performs the action the user configured. Actions can include resetting the connection, dropping the request, or sending an error message to the user's browser. This allows the appliance to screen unwanted requests and reduces your servers' exposure to attacks.
This feature can also analyze HTTP GET and POST requests and filter out known bad signatures, allowing it to defend your servers against HTTP-based attacks.

For more information, see "Content Filtering."

Responder

Functions like an advanced filter and can be used to generate responses from the appliance to the client. Some common uses of this feature are generation of redirect responses, user defined responses, and resets.
For more information, see "Responder."

Rewrite

Modifies HTTP headers and body text. You can use the rewrite feature to add HTTP headers to an HTTP request or response, make modifications to individual HTTP headers, or delete HTTP headers. It also enables you to modify the HTTP body in requests and responses.
When the appliance receives a request or sends a response, it checks for rewrite rules, and if applicable rules exist, it applies them to the request or response before passing it on to the web server or client computer.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For more information, see "Rewrite."

Priority Queuing

Prioritizes user requests to ensure that the most important traffic is serviced first during surges in request volume. You can establish priority based on request URLs, cookies, or a variety of other factors. The appliance places requests in a three-tier queue based on their configured priority, enabling business-critical transactions to flow smoothly even during surges or site attacks.
For more information, see "Priority Queuing."

Surge Protection

Regulates the flow of user requests to servers and controls the number of users that can simultaneously access the resources on the servers, queuing any additional requests once your servers have reached their capacity. By controlling the rate at which connections can be established, the appliance blocks surges in requests from being passed on to your servers, thus preventing site overload.
For more information, see "Surge Protection."

NetScaler Gateway

NetScaler Gateway is a secure application access solution that provides administrators granular application-level policy and action controls to secure access to applications and data while allowing users to work from anywhere. It gives IT administrators a single point of control and tools to help ensure compliance with regulations and the highest levels of information security across and outside the enterprise. At the same time, it empowers users with a single point of access—optimized for roles, devices, and networks—to the enterprise applications and data they need. This unique combination of capabilities helps maximize the productivity of today's mobile workforce.

For more information, see "NetScaler Gateway."

Application Firewall

Protects applications from misuse by hackers and malware, such as cross site scripting attacks, buffer overflow attacks, SQL injection attacks, and forceful browsing, by filtering traffic between each protected web server and users that connect to any web site on that web server. The application firewall examines all traffic for evidence of attacks on web server security or misuse of web server resources, and takes the appropriate action to prevent these attacks from succeeding.
For more information, see "Application Firewall."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Visibility Feature

Sep 04, 2013

NetScaler Insight Center

NetScaler Insight Center is a high performance collector that provides end-to-end user experience visibility across Web and HDX (ICA) traffic. It collects HTTP and ICA AppFlow records generated by NetScaler ADC appliances and populates analytical reports covering Layer 3 to Layer 7 statistics. NetScaler Insight Center provides in-depth analysis for the last five minutes of real-time data, and for historical data collected for the last one hour, one day, one week, and one month.
HDX (ICA) analytic dashboard enables you to drill down from HDX Users, Applications, Desktops, and even from gateway-level information. Similarly, HTTP analytics provide a bird's eye view of Web Applications, URLs Accessed, Client IP Addresses and Server IP Addresses, and other dashboards. The administrator can drill down and identify the pain points from any of these dashboards, as appropriate for the use case.

Enhanced Application Visibility Using AppFlow

The Citrix NetScaler appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. AppFlow transmits this information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information.
Using UDP as the transport protocol, AppFlow transmits the collected data, called *flow records*, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor.

To limit the types of flows to monitor, by sampling and filtering the application traffic, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

For more information, see "AppFlow."

Stream Analytics

The performance of your web site or application depends on how well you optimize the delivery of the most frequently requested content. Techniques such as caching and compression help accelerate the delivery of services to clients, but you need to be able to identify the resources that are requested most frequently, and then cache or compress those resources. You can identify the most frequently used resources by aggregating real-time statistics about website or application traffic. Statistics such as how frequently a resource is accessed relative to other resources and how much bandwidth is consumed by those resources help you determine whether those resources need to be cached or compressed to improve server performance and network utilization. Statistics such as response times and the number of concurrent connections to the application help you determine whether you must enhance server-side resources.

If the web site or application does not change frequently, you can use products that collect statistical data, and then manually analyze the statistics and optimize the delivery of content. However, if you do not want to perform manual optimizations, or if your web site or application is dynamic in nature, you need infrastructure that can not only collect

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

statistical data but can also automatically optimize the delivery of resources on the basis of the statistics. On the NetScaler appliance, this functionality is provided by the Stream Analytics feature. The feature operates on a single NetScaler appliance and collects run-time statistics on the basis of criteria that you define. When used with NetScaler policies, the feature also provides you with the infrastructure that you need for automatic, real-time traffic optimization.

For more information, see "Stream Analytics."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cloud Integration Feature

Aug 23, 2013

AutoScale

All applications have specific usage patterns that comprise peaks and troughs. These load variations can be dynamic in nature and difficult to predict, given that they depend on several factors that are intrinsic to the use case. Cloud users have to constantly monitor the load on their application fleet and make sure that these variations have minimum impact on end users. During periods of peak usage, when the application fleet is overloaded and end users experience significant latency, they have to deploy additional application instances. During trough periods, the expanded fleet is underutilized. So they might have to remove additional instances or bear unnecessary cost overheads. In most cases, they have to perform these tasks manually.

If your organization uses Citrix CloudPlatform to deploy and manage the cloud environment, users can use the *AutoScale* feature in CloudPlatform, in conjunction with a Citrix NetScaler appliance, to automatically scale their applications as needed. The AutoScale feature is part of the elastic load balancing feature in CloudPlatform. A CloudPlatform user can use the AutoScale feature to specify thresholds for various conditions for automatically scaling the application fleet upward and downward. CloudPlatform, in turn, configures the NetScaler appliance (by using the NetScaler NITRO API) to load balance traffic to the application virtual machines (VMs), monitor application thresholds and performance, and trigger scale-up and scale-down actions to add or remove VMs to or from the application fleet.

As the NetScaler administrator, you do not have to perform any tasks for configuring AutoScale on the NetScaler appliance. However, you might have to be aware of certain prerequisites, and you might have to troubleshoot the configuration if issues arise in the AutoScale configuration. To troubleshoot the configuration, you have to be aware of how CloudPlatform works and what configuration CloudPlatform pushes to the NetScaler appliance. You also need a working knowledge of how to troubleshoot issues on a NetScaler appliance.

For more information about AutoScale, see "AutoScale: Automatic Scaling in the Citrix CloudPlatform Environment."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Hardware Installation

Aug 02, 2017

> **Note**
>
> For the latest content, see NetScaler MPX.

The following sections describe the hardware installation and initial configuration for all NetScaler hardware platforms.

Hardware Platforms	Describes the NetScaler hardware platforms and provides detailed information about each platform and its components.
Preparing for Installation	Describes how to unpack the NetScaler appliance and prepare the site and rack for installing the appliance. Lists the cautions and warnings that you should review before you install the appliance.
Installing the Hardware	Describes the steps to install the rails, mount the hardware, connect the cables, and turn on the appliance.
Initial Configuration	Describes how to perform initial configuration of your NetScaler appliance and assign management and network IP addresses.
Lights Out Management Port of the NetScaler Appliance	Describes the different operations you can perform on your NetScaler appliance by using the Lights Out Management Port.

For information about NetScaler hardware and software compatibility and the supported upgrade and downgrade paths, see http://support.citrix.com/article/CTX113357.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Licensing, Upgrading, and Downgrading

Oct 04, 2016

The following topics describe the migration instructions for setting up a new version of a NetScaler appliance with a list of all new and deprecated commands, parameters, and SNMP OIDs.

This document includes the following information:

- New and Deprecated Commands, Parameters, and SNMP OIDs
- NetScaler Licensing Overview
- NetScaler Gateway Universal License
- Upgrading or Downgrading the System Software
- Troubleshooting

New and Deprecated Commands, Parameters, and SNMP OIDs

New Commands

The following table lists all the new commands in release 11.0

Command Group	Command
Authentication	add authentication loginSchema rm authentication loginSchema add authentication loginSchemaPolicy rm authentication loginSchemaPolicy set authentication loginSchemaPolicy show authentication loginSchemaPolicy rename authentication loginSchemaPolicy
Cache	stat cache contentGroup
DNS	set dns nameServer
ICA	add ica policy rm ica policy set ica policy rename ica policy show ica policy rm ica action

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
	rename ica action bind ica global unbind ica global show ica global
vPath	set vPathParam show vPathParam
APP ALG	set appAlgParam show appAlgParam
NetScaler	query ns config import ns extension rm ns extension add ns extension update ns extension bind ns extension set ns extension show ns extension
RDP	add rdp clientprofile show rdp clientprofile rm rdp clientprofile add rdp serverprofile show rdp serverprofile rm rdp serverprofile
SSL	show ssl certChain add ssl profile rm ssl profile set ssl profile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
System	rm system file show system file create system kek export system kek import system kek
SNMP	add smpp user rm smpp user set smpp user show smpp user
VPN	check vpn vserver kill vpn icaConnection
WebFront	install wf package uninstall wf package show wf package add wf site rm wf site set wf site show wf site
LLDP	set lldp param show lldp param
Media Classification	show mediaclassification stats

New Parameters

The following table lists all the new parameters in release 11.0.

Command Group	Command
Basic	add service [-processLocal]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
	add service [-dnsProfileName] set service [-processLocal] set service [-dnsProfileName] unset service [-processLocal] unset service [-dnsProfileName]
AAA	bind aaa user [-intranetIP6] unbind aaa user [-intranetIP6] bind aaa group [-intranetIP6] unbind aaa group [-intranetIP6] add aaa preauthenticationaction [-defaultEPAGroup] set aaa preauthenticationaction [-defaultEPAGroup] add aaa kcdAccount [-enterpriseRealm] add aaa kcdAccount [-serviceSPN] set aaa kcdAccount [-enterpriseRealm] set aaa kcdAccount [-serviceSPN] unset aaa kcdAccount [-enterpriseRealm] unset aaa kcdAccount [-serviceSPN]
Application Firewall	add appfw profile [-contentTypeAction] add appfw profile [-inspectContentTypes] add appfw profile [-SQLInjectionType] add appfw profile [SQLInjectionCheckSQLWildChars] add appfw profile [-doSecureCreditCardLogging] add appfw profile [trace] add appfw profile [-XMLSQLInjectionType] add appfw profile [-XMLSQLInjectionCheckSQLWildChars] add appfw profile [-URLDecodeRequestCookies] set appfw profile [-URLDecodeRequestCookies]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
	bind appfw profile [-CreditCardNumber] unbind appfw profile [-CreditCardNumber] set appfw settings [-GeoLocationLogging] import appfw signatures [-log]
Application Firewall	add appfw profile [-contentTypeAction] add appfw profile [-inspectContentTypes] add appfw profile [-SQLInjectionType] add appfw profile [SQLInjectionCheckSQLWildChars] add appfw profile [-doSecureCreditCardLogging] add appfw profile [trace] add appfw profile [-XMLSQLInjectionType] add appfw profile [-XMLSQLInjectionCheckSQLWildChars] add appfw profile [-URLDecodeRequestCookies] set appfw profile [-URLDecodeRequestCookies] bind appfw profile [-CreditCardNumber] unbind appfw profile [-CreditCardNumber] set appfw settings [-GeoLocationLogging] import appfw signatures [-log]
Audit	add audit syslogAction [-alg] add audit syslogAction [-transport] add audit syslogAction [-tcpProfileName] add audit syslogAction [-maxLogDataSizeToHold] add audit syslogAction [-dns] add audit syslogAction [-lbVserverName] set audit syslogAction [-alg] set audit syslogAction [-transport] set audit syslogAction [-tcpProfileName]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
	set audit syslogAction [-maxLogDataSizeToHold] set audit syslogAction [-dns] set audit syslogAction: [-lbVserverName] unset audit syslogAction [-alg] unset audit syslogAction [-transport] unset audit syslogAction [-tcpProfileName] unset audit syslogAction [-maxLogDataSizeToHold] unset audit syslogAction [-dns] add audit nslogAction [-alg] set audit nslogAction [-alg] unset audit nslogAction [-alg] set audit syslogParams [-alg] set audit syslogParams [-alg] unset audit syslogParams [-alg] unset audit syslogParams [-dns] set audit nslogParams [-alg] unset audit nslogParams [-alg]
Authentication	add authentication ldapAction [-referralDNSLookup] add authentication ldapAction [-msSRVRecordlocation] add authentication ldapAction [-Attribute1] add authentication ldapAction [-Attribute2] add authentication ldapAction [-Attribute3] add authentication ldapAction [-Attribute4] add authentication ldapAction [-Attribute5] add authentication ldapAction [-Attribute6] add authentication ldapAction [-Attribute7] add authentication ldapAction [-Attribute8] add authentication ldapAction [-Attribute9]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
	add authentication ldapAction [-Attribute10] add authentication ldapAction [-Attribute11] add authentication ldapAction [-Attribute12] add authentication ldapAction [-Attribute13] add authentication ldapAction [-Attribute14] add authentication ldapAction [-Attribute15] add authentication ldapAction [-Attribute16] set authentication ldapAction [-referralDNSLookup] set authentication ldapAction [-msSRVRecordlocation] set authentication ldapAction [-Attribute1] set authentication ldapAction [-Attribute2] set authentication ldapAction [-Attribute3] set authentication ldapAction [-Attribute4] set authentication ldapAction [-Attribute5] set authentication ldapAction [-Attribute6] set authentication ldapAction [-Attribute7] set authentication ldapAction [-Attribute8] set authentication ldapAction [-Attribute9] set authentication ldapAction [-Attribute10] set authentication ldapAction [-Attribute11] set authentication ldapAction [-Attribute12] set authentication ldapAction [-Attribute13] set authentication ldapAction [-Attribute14] set authentication ldapAction [-Attribute15] set authentication ldapAction [-Attribute16] add authentication samlAction [-requestedAuthnContext] add authentication samlAction [-authnCtxClassRef] add authentication samlAction [-samlBinding]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
	add authentication samlAction [-attributeConsumingServiceIndex] add authentication samlAction [-sendThumbprint] add authentication samlAction [-enforceUserName] add authentication samlAction [-logoutURL] add authentication samlAction [-artifactResolutionServiceURL] set authentication samlAction [-requestedAuthnContext] set authentication samlAction [-authnCtxClassRef] set authentication samlAction [-samlBinding] set authentication samlAction [-attributeConsumingServiceIndex] set authentication samlAction [-sendThumbprint] set authentication samlAction [-enforceUserName] set authentication samlAction [-logoutURL] set authentication samlAction [-artifactResolutionServiceURL] add authentication dfaAction [-clientID] add authentication dfaAction [-serverURL] add authentication dfaAction [-passPhrase] set authentication dfaAction [-clientID] set authentication dfaAction [-serverURL] set authentication dfaAction [-passPhrase]
Content Adaptation	add ca action [-accumResSize] add ca action [-lbvserver] add ca action [-type] set ca action [-accumResSize] set ca action [-type] set ca action [-lbvserver]
Cluster	add cluster instance [-quorumType] set cluster instance [-quorumType]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
	set cluster instance [-nodegroup] add cluster node [-name] add cluster node [-cfgflags] set cluster node [-name] set cluster node [-cfgflags] rm cluster node [-clearNodegroupConfig] add cluster nodegroup [-sticky] add cluster nodegroup [-boundedConfigEntitiesCnt] add cluster nodegroup [-state] add cluster nodegroup [-priority] set cluster nodegroup [-boundedConfigEntitiesCnt] set cluster nodegroup [-state] set cluster nodegroup [-priority] bind cluster nodegroup [-gslbSite] bind cluster nodegroup [-service] unbind cluster nodegroup [-gslbSite] unbind cluster nodegroup [-service]
	add cr action [-name] add cr action [-crType] add cr action [-builtin] add cr action [-comment] rm cr action [-name] set cr action [-name] set cr action[-crType] set cr action[-comment] show cr action [-name] rename cr action [-name]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
Content Switching	add cs vserver [-RHIstate] add cs vserver [-dnsProfileName] set cs vserver [-RHIstate] set cs vserver [-dnsProfileName] unset cs vserver [-dnsProfileName] add cs action [-targetVserver] set cs action [-targetVserver]
DNS	add dns nameServer [-dnsProfileName] set dns nameServer [-dnsProfileName] add dns policy [-logAction] set dns policy [-logAction] add dns action [-dnsProfileName] set dns action [-dnsProfileName]
Front End Optimization	set feo parameter [-logLevel] set feo parameter [-builtin]
IPsec	add ipsec profile [-replayWindowSize] add ipsec profile [-perfectForwardSecrecy] set ipsec parameter [-replayWindowSize] set ipsec parameter [-perfectForwardSecrecy] unset ipsec parameter [-replayWindowSize] unset ipsec parameter [-perfectForwardSecrecy]
Load Balancing	add lb monitor [-storefrontcheckbackendservices] set lb monitor [-storefrontcheckbackendservices] set lb monitor [-storedb] set lb monitor [-trofscode] set lb monitor [-trofsstring] add lb vserver [-RHIstate]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
	add lb vserver [-RecursionAvailable] add lb vserver [-processLocal] add lb vserver [-dnsProfileName] set lb vserver [-RHIstate] set lb vserver [-RecursionAvailable] set lb vserver [-processLocal] set lb vserver [-dnsProfileName] unset lb vserver [-dnsProfileName] set lb sipParameters [-rnatsecuresrcport] set lb sipParameters [-rnatsecuredstport] set lb parameter [-useSecuredPersistenceCookie] set lb parameter [-cookiePassphrase]
Responder	add responder action [-responseStatusCode] add responder action [-reasonPhrase] set responder action [-responseStatusCode] set responder action [-reasonPhrase]
Rewrite	add rewrite policy [-isDefault] add rewrite policylabel [-builtin]
SNMP	set snmp option [-partitionNameInTrap]
SSL	add ssl certKey [-gslbServiceFlag] set ssl service [-sslProfile] set ssl service [-gslbServiceFlag] bind ssl service [-eccCurveName] unbind ssl service [-eccCurveName] set ssl serviceGroup [-sslProfile] set ssl vserver [-sslProfile]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command Group	Command
	bind ssl vserver [-eccCurveName] unbind ssl vserver [-eccCurveName] show ssl certChain [-CertKeyName] add ssl profile [-name] add ssl profile [-gslbServiceFlag] rm ssl profile [-name] set ssl profile [-name] show ssl profile [-name]

Deprecated Parameters

The following parameters are deprecated in release 11.0:

- start nstrace -tcpdump
- start nstrace -doruntimemerge
- fw profile -SQLInjectionOnlyCheckFieldsWithSQLChars
- fw profile -XMLSQLInjectionOnlyCheckFieldsWithSQLChars
- set appfw profile -SQLInjectionOnlyCheckFieldsWithSQLChars
- set appfw profile -XMLSQLInjectionOnlyCheckFieldsWithSQLChars
- authentication negotiateAction -OU
- set authentication negotiateAction -OU
- set dns parameter -cacheRecords
- policy patset -indexType
- policy dataset -indexType
- write action -bypassSafetyCheck
- set rewrite action -bypassSafetyCheck
- vpn sessionAction -spoofIIP
- set vpn sessionAction -spoofIIP
- set vpn parameter -spoofIIP
- setlsnparameter -memlimit

New SNMP OIDs

The following lists all the new SNMP OIDs in release 11.0:

- mediaclassification,1.3.6.1.4.1.5951.4.1.1.21.17
- sslTotTransactionsRate,1.3.6.1.4.1.5951.4.1.1.47.333
- sslTotSSLv2TransactionsRate,1.3.6.1.4.1.5951.4.1.1.47.334
- sslTotSSLv3TransactionsRate,1.3.6.1.4.1.5951.4.1.1.47.335
- sslTotTLSv1TransactionsRate,1.3.6.1.4.1.5951.4.1.1.47.336
- httpTotRequestsRate,1.3.6.1.4.1.5951.4.1.1.48.76
- httpTotResposesRate,1.3.6.1.4.1.5951.4.1.1.48.77
- dtlsStaValidation,1.3.6.1.4.1.5951.4.1.1.66.47
- dtlsStaValidationError,1.3.6.1.4.1.5951.4.1.1.66.48

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- dtlsStaReconnectTktReq,1.3.6.1.4.1.5951.4.1.1.66.49
- dtlsStaPostResponseSuccess,1.3.6.1.4.1.5951.4.1.1.66.50
- dtlsStaPostResponseError,1.3.6.1.4.1.5951.4.1.1.66.51
- nsLsnGlobalStatsGroup,1.3.6.1.4.1.5951.4.1.1.83.1
- lsnTotRxPkts,1.3.6.1.4.1.5951.4.1.1.83.1.1
- lsnTotRxBytes,1.3.6.1.4.1.5951.4.1.1.83.1.2
- lsnTotTxPkts,1.3.6.1.4.1.5951.4.1.1.83.1.3
- lsnTotTxBytes,1.3.6.1.4.1.5951.4.1.1.83.1.4
- lsnCurSessions,1.3.6.1.4.1.5951.4.1.1.83.1.5
- lsnCurSubscribers,1.3.6.1.4.1.5951.4.1.1.83.1.6
- nsLsnGroupTable,1.3.6.1.4.1.5951.4.1.1.83.2
- nsLsnGroupEntry,1.3.6.1.4.1.5951.4.1.1.83.2.1
- lsnGrpName,1.3.6.1.4.1.5951.4.1.1.83.2.1.1
- lsnGrpTotTranslPkts,1.3.6.1.4.1.5951.4.1.1.83.2.1.2
- lsnGrpTotTranslBytes,1.3.6.1.4.1.5951.4.1.1.83.2.1.3
- lsnGrpCurSessions,1.3.6.1.4.1.5951.4.1.1.83.2.1.4
- lsnGrpCurSubscribers,1.3.6.1.4.1.5951.4.1.1.83.2.1.5
- nsPPTPStatsGroup,1.3.6.1.4.1.5951.4.1.1.83.3
- pptpCurCtrlSessions,1.3.6.1.4.1.5951.4.1.1.83.3.1
- pptpTotCtrlSessions,1.3.6.1.4.1.5951.4.1.1.83.3.2
- pptpTotGRESessions,1.3.6.1.4.1.5951.4.1.1.83.3.3
- pptpCurGRESessions,1.3.6.1.4.1.5951.4.1.1.83.3.4
- pptpStrtCtrlConnReply,1.3.6.1.4.1.5951.4.1.1.83.3.5
- pptpStpCtrlConnReply,1.3.6.1.4.1.5951.4.1.1.83.3.6
- pptpOutCallReply,1.3.6.1.4.1.5951.4.1.1.83.3.7
- pptpCtlErrNoCallidMap,1.3.6.1.4.1.5951.4.1.1.83.3.8
- pptpInCallReq,1.3.6.1.4.1.5951.4.1.1.83.3.9
- pptpInCallConn,1.3.6.1.4.1.5951.4.1.1.83.3.10
- pptpCallDisNotify,1.3.6.1.4.1.5951.4.1.1.83.3.11
- pptpWanErrNotify,1.3.6.1.4.1.5951.4.1.1.83.3.12
- pptpSetLinkInfo,1.3.6.1.4.1.5951.4.1.1.83.3.13
- pptpEchoReq,1.3.6.1.4.1.5951.4.1.1.83.3.14
- pptpEchoRep,1.3.6.1.4.1.5951.4.1.1.83.3.15
- pptpStrtCtrlConnReq,1.3.6.1.4.1.5951.4.1.1.83.3.16
- pptpStpCtrlConnReq,1.3.6.1.4.1.5951.4.1.1.83.3.17
- pptpOutCallReq,1.3.6.1.4.1.5951.4.1.1.83.3.18
- pptpInCallRep,1.3.6.1.4.1.5951.4.1.1.83.3.19
- pptpCallClrReq,1.3.6.1.4.1.5951.4.1.1.83.3.20
- pptpCntlOcreq,1.3.6.1.4.1.5951.4.1.1.83.3.21
- pptpCntlIcreq,1.3.6.1.4.1.5951.4.1.1.83.3.22
- nsLsnDSLiteGlobalStatsGroup,1.3.6.1.4.1.5951.4.1.1.83.4
- lsnTotDSLiteRxPkts,1.3.6.1.4.1.5951.4.1.1.83.4.1
- lsnTotDSLiteRxBytes,1.3.6.1.4.1.5951.4.1.1.83.4.2
- lsnTotDSLiteTxPkts,1.3.6.1.4.1.5951.4.1.1.83.4.3
- lsnTotDSLiteTxBytes,1.3.6.1.4.1.5951.4.1.1.83.4.4
- lsnCurDSLiteSessions,1.3.6.1.4.1.5951.4.1.1.83.4.5

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- lsnDSLiteCurSubscribers,1.3.6.1.4.1.5951.4.1.1.83.4.6
- nsmcStatsGroup,1.3.6.1.4.1.5951.4.1.1.84
- mcTotaudio,1.3.6.1.4.1.5951.4.1.1.84.1
- mcTotvideo,1.3.6.1.4.1.5951.4.1.1.84.2
- mcTotother,1.3.6.1.4.1.5951.4.1.1.84.3
- mcMsSmthStrmVid,1.3.6.1.4.1.5951.4.1.1.84.4
- mcMsSmthStrmVidBytes,1.3.6.1.4.1.5951.4.1.1.84.5
- mcMsSmthStrVidpl,1.3.6.1.4.1.5951.4.1.1.84.6
- mcMsSmthStrmPlVidBytespl,1.3.6.1.4.1.5951.4.1.1.84.7
- mccApleLiveStrmngVid,1.3.6.1.4.1.5951.4.1.1.84.8
- mcAppleLiveStreamingVidBytes,1.3.6.1.4.1.5951.4.1.1.84.9
- mccAppleLiveStrmngVidpl,1.3.6.1.4.1.5951.4.1.1.84.10
- mcAppleLiveStreamingPlaylistVidBytespl,1.3.6.1.4.1.5951.4.1.1.84.11
- mcADTSaudio,1.3.6.1.4.1.5951.4.1.1.84.12
- mcADTSaudioBytes,1.3.6.1.4.1.5951.4.1.1.84.13
- mcAACaudio,1.3.6.1.4.1.5951.4.1.1.84.14
- mcAACaudiobytes,1.3.6.1.4.1.5951.4.1.1.84.15
- mcFLVVid,1.3.6.1.4.1.5951.4.1.1.84.16
- mcFLVVidBytes,1.3.6.1.4.1.5951.4.1.1.84.17
- mcMP4Vid,1.3.6.1.4.1.5951.4.1.1.84.18
- mcMP4VidBytes,1.3.6.1.4.1.5951.4.1.1.84.19
- mc3PVid,1.3.6.1.4.1.5951.4.1.1.84.20
- mc3GPVidBytes,1.3.6.1.4.1.5951.4.1.1.84.21
- mcTotResponse,1.3.6.1.4.1.5951.4.1.1.84.22
- mcAndroid,1.3.6.1.4.1.5951.4.1.1.84.23
- mcLaptopDesktp,1.3.6.1.4.1.5951.4.1.1.84.24
- mcIos,1.3.6.1.4.1.5951.4.1.1.84.25
- mcUnidentified,1.3.6.1.4.1.5951.4.1.1.84.26
- mcOther,1.3.6.1.4.1.5951.4.1.1.84.27
- lsnSubscrIP,1.3.6.1.4.1.5951.4.1.10.2.53
- lsnSubscrTD,1.3.6.1.4.1.5951.4.1.10.2.54
- protocol,1.3.6.1.4.1.5951.4.1.10.2.55
- vsvrCurSoValue,1.3.6.1.4.1.5951.4.1.10.2.56
- vsvrSoMethod,1.3.6.1.4.1.5951.4.1.10.2.57
- vsvrSoThresh,1.3.6.1.4.1.5951.4.1.10.2.58
- lsnSessLimitExceededBy,1.3.6.1.4.1.5951.4.1.10.2.59
- nsPartitionName,1.3.6.1.4.1.5951.4.1.10.2.60
- operation,1.3.6.1.4.1.5951.4.1.10.2.61
- fromPartition,1.3.6.1.4.1.5951.4.1.10.2.62
- toPartition,1.3.6.1.4.1.5951.4.1.10.2.63
- clPeerID,1.3.6.1.4.1.5951.4.1.10.2.64

Deprecated SNMP OIDs

The following lists all the deprecated SNMP OIDs in release 11.0:

- gslbPoliciesTable,1.3.6.1.4.1.5951.4.1.1.51.2.2
- gslbPoliciesEntry,1.3.6.1.4.1.5951.4.1.1.51.2.2.1

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- gslbPolicyName,1.3.6.1.4.1.5951.4.1.1.51.2.2.1.1
- totalHits,1.3.6.1.4.1.5951.4.1.1.51.2.2.1.2

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler Licensing Overview

Jan 25, 2016

The process of allocating your NetScaler licenses has been greatly simplified. The new licensing framework allows you to focus on getting maximum value from Citrix products.

In the NetScaler configuration utility (GUI), you can use your hardware serial number (HSN) or your license activation code (LAC) to allocate your licenses. Alternatively, if a license is already present on your local computer, you can upload it to the appliance.

For all other functionality, such as returning or reallocating your license, you must use the licensing portal. Optionally, you can still use the licensing portal for license allocation. For more information about the licensing portal, see "http://support.citrix.com/article/CTX131110".

Note: You must purchase separate licenses for each appliance in a high availability (HA) pair. Make sure that the same type of licenses are installed on both the appliances. For example, if you purchase a platinum license for one appliance, you must purchase another platinum license for the other appliance.
This document includes the following information:

- Prerequisites
- Allocating your License by using the Configuration Utility
- Installing the License
- Verifying the Licensed Features
- Enabling or Disabling a Feature

Prerequisites

To use the hardware serial number or license activation code to allocate your licenses:

- You must be able to access public domains through the appliance. For example, the appliance should be able to access www.citrix.com. The license allocation software internally accesses the Citrix licensing portal for your license. To access a public domain, you can either use a proxy server or set up a DNS server and, on your NetScaler appliance, configure a NetScaler IP (NSIP) address or a subnet IP (SNIP) address.
- Your license must be linked to your hardware, or you must have a valid license activation code (LAC). Citrix sends your LAC by email when you purchase a license.

Allocating your License by using the Configuration Utility

If your license is already linked to your hardware, the license allocation process can use the hardware serial number. Otherwise, you must type the license activation code (LAC).

You can partially allocate licenses as required for your deployment. For example, if your license file contains ten licenses, but your current requirement is for only six licenses, you can allocate six licenses now, and allocate additional licenses later. You cannot allocate more than the total number of licenses present in your license file.

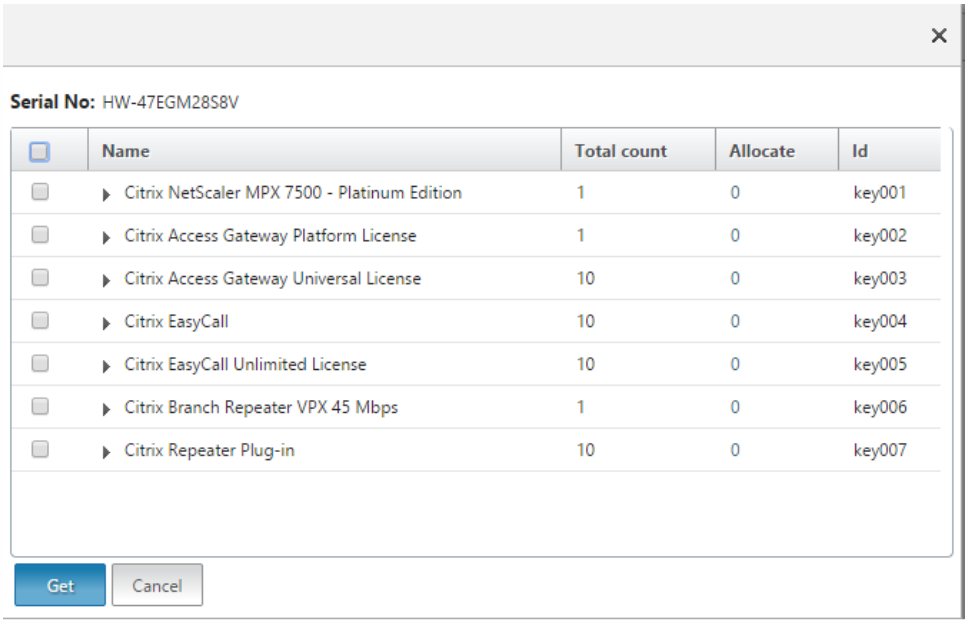
To allocate your license

1. In a web browser, type the IP address of the NetScaler appliance (for example, http://192.168.100.1).
2. In User Name and Password, type the administrator credentials.
3. On the Configuration tab, navigate to System > Licenses.
4. In the details pane, click Manage Licenses, click Add New License, and then select one of the following options:

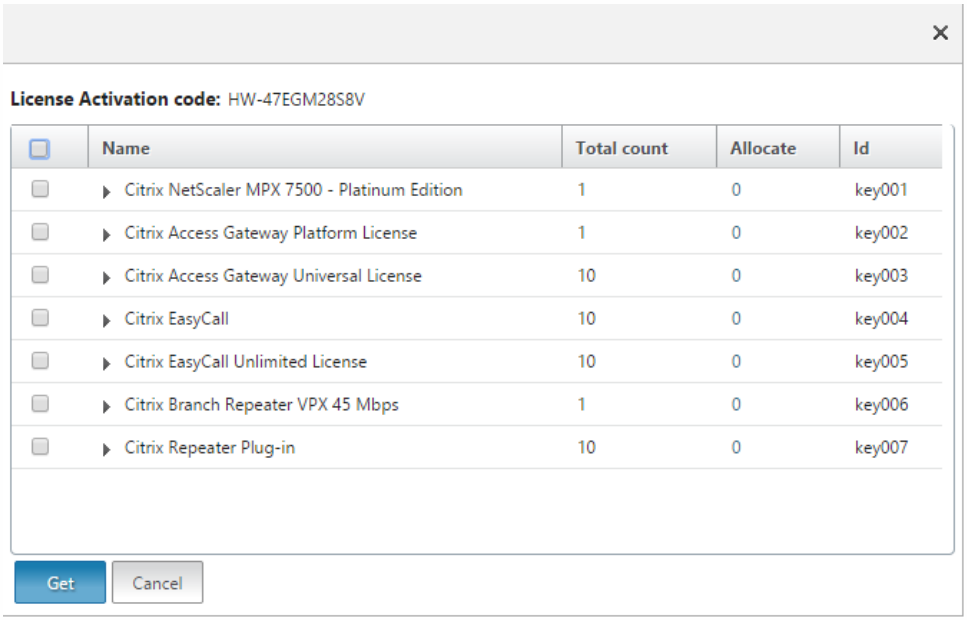
© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Use Serial Number.** The software internally fetches the serial number of your appliance and uses this number to display your license(s).
- **Use License Activation Code.** Citrix emails the LAC for the license that you purchased. Enter the LAC in the text box. If you do not want to configure internet connectivity on the NetScaler appliance, you can use a proxy server. Select the **Connect through Proxy Server** check box and specify the IP address and port of your proxy server.

5. Click Get Licenses. Depending on the option that you selected, one of the following dialog boxes appears.
   - The following dialog box appears if you selected Hardware Serial Number.

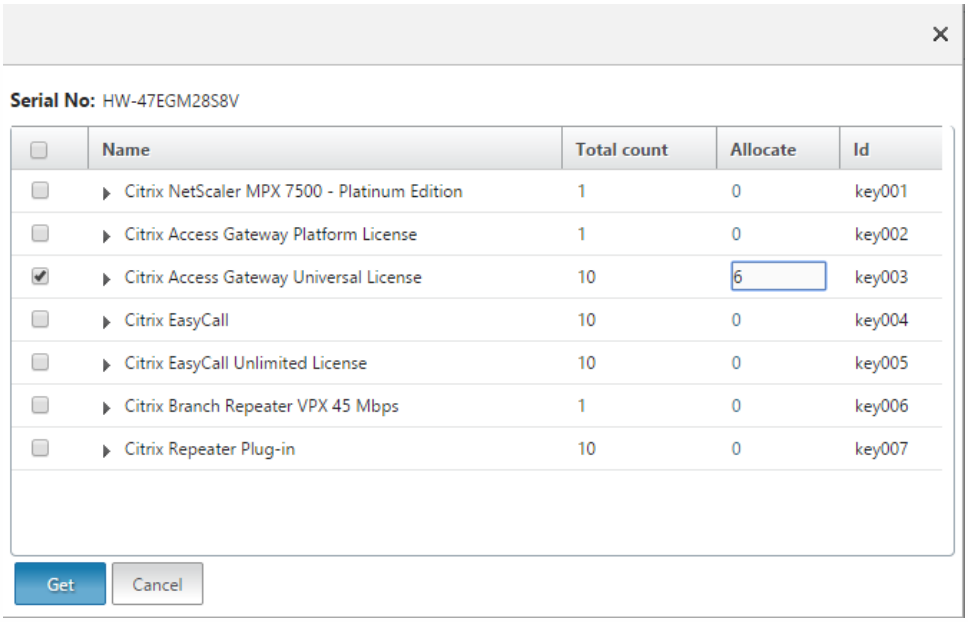


   - The following dialog box appears if you selected License Activation Code.

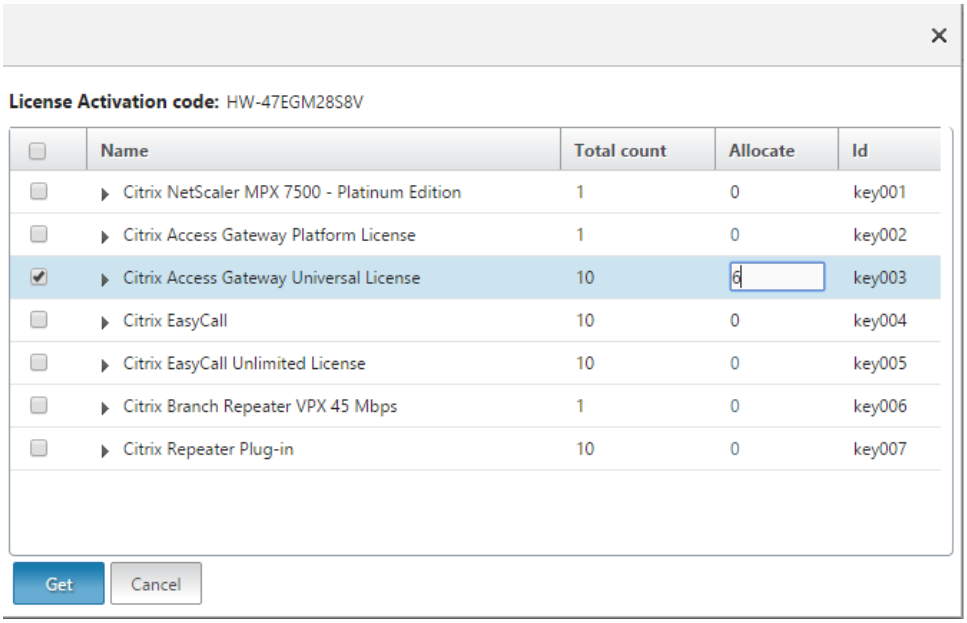


6. Select the license file that you want to use to allocate your licenses.
7. In the **Allocate** column, enter the number of licenses to be allocated. Then click **Get**.
   - If you selected **Hardware Serial Number**, enter the number of licenses, as shown in the following screen shot.

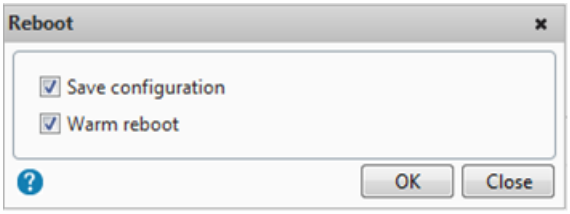
© 1999-2017 Citrix Systems, Inc. All rights reserved.



- If you selected **License Activation Code**, enter the number of licenses, as shown in the following screen shot.



8. Click Reboot for the license to take effect.
9. In the Reboot dialog box, click OK to proceed with the changes, or click Close to cancel the changes.



Installing the License

If you downloaded your license file to your local computer by accessing the licensing portal, you must upload the license to the appliance.

To install a license file by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. In a web browser, type the IP address of the NetScaler (for example, http://192.168.100.1).
2. In User Name and Password, type the administrator credentials.
3. On the Configuration tab, navigate to System > Licenses .
4. In the details pane, click Manage Licenses.
5. Click Add New License, then select **Upload license files from a local computer**.
6. Click Browse. Navigate to the location of the license files, select the license file, and then click Open.
7. Click Reboot to apply the license.
8. In the Reboot dialog box, click OK to proceed with the changes, or click Close to cancel the changes.

To install the licenses by using the command line interface

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
2. Log on to the NetScaler by using the administrator credentials.
3. Switch to the shell prompt, create a license subdirectory in the nsconfig directory, if it does not exist, and copy the new license file(s) to this directory.
   Example

   ```
   login: nsroot
   Password: nsroot
   Last login: Mon Aug  4 03:37:27 2008 from 10.102.29.9
   Done
   > shell
   Last login: Mon Aug  4 03:51:42 from 10.103.25.64
   root@ns# mkdir /nsconfig/license
   root@ns# cd /nsconfig/license
   ```

   Copy the new license file(s) to this directory.
   Note: The NetScaler appliance does not prompt for a reboot option when you use the command line interface to install the licenses. Run the reboot -w command to warm reboot the system, or run the reboot command to reboot the system normally.

Verifying the Licensed Features

Before using a feature, make sure that your license supports the feature.

To verify the licensed features by using the command line interface

1. Open an SSH connection to the NetScaler by using an SSH client, such as PuTTY.
2. Log on to the NetScaler by using the administrator credentials.
3. At the command prompt, enter the sh ns license command to display the features supported by the license.
   Example

   ```
   sh ns license
       License status:
                   Web Logging: YES
               Surge Protection: YES
                 .......

                 HTML Injection: YES
   Done
   ```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To verify the licensed features by using the configuration utility

1. In a Web browser, type the IP address of the NetScaler, such as http://192.168.100.1.
2. In User Name and Password, type the administrator credentials.
3. Provide the User name and Password and click Login.
4. In the navigation pane, expand System, and then click Licenses. You will see a green check mark next to the licensed features.

Enabling or Disabling a Feature

When you use the NetScaler appliance for the first time, you need to enable a feature before you can use its functionality. If you configure a feature before it is enabled, a warning message appears. The configuration is saved but it will apply only after the feature is enabled.

To enable a feature by using the command line interface

At the NetScaler command prompt, type the following commands to enable a feature and verify the configuration:

- enable feature <FeatureName>
- show feature

Example

```
enable feature lb cs
done
 >show feature

     Feature                Acronym        Status
     -------                -------        ------
 1)  Web Logging            WL             OFF
 2)  Surge Protection       SP             ON
 3)  Load Balancing         LB             ON
 4)  Content Switching      CS             ON
 5)  Cache Redirection      CR             ON
 .
 .
 .
 24) NetScaler Push         push           OFF
 Done
```

The example shows how to enable load balancing (lb) and content switching (cs).

If the license key is not available for a particular feature, the following error message appears for that feature:

ERROR: feature(s) not licensed

Note: To enable an optional feature, you need a feature-specific license. For example, if you have purchased and installed the Citrix NetScaler Enterprise Edition license and need to enable the Integrated Caching feature, you first need to purchase and install the AppCache license.

To disable a feature by using the command line interface

At the NetScaler command prompt, type the following commands to disable a feature and verify the configuration:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- disable feature <FeatureName>
- show feature

Example

The following example shows how to disable load balancing (LB).

```
> disable feature lb
 Done
> show feature

      Feature                Acronym        Status
      -------                -------        ------
 1)   Web Logging            WL             OFF
 2)   Surge Protection       SP             ON
 3)   Load Balancing         LB             OFF
 4)   Content Switching      CS             ON
 .
 .
 .
 24)  NetScaler Push         push           OFF
 Done
>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler Gateway Universal License

Jul 30, 2017

The NetScaler Gateway universal license limits the number of concurrent user sessions to the number of licenses purchased. If you purchase 100 licenses, you can have 100 concurrent sessions at any time. When a user ends a session, that license is released for the next user. A user who logs on to the NetScaler Gateway from more than one computer occupies a license for each session.

If all licenses are occupied, no additional connections can be opened until a user ends a session or the administrator terminates the session using the configuration utility. When a connection is closed, the license is released and can be used for a new user.

This document includes the following information:

- Obtaining the Universal License
- Installing the Universal License
- Verifying Installation of the Universal License

Obtaining the Universal License

You need the following information before going to the Citrix Web site for the universal license.

Your user ID and password for My Citrix

Register at My Citrix (www.mycitrix.com) to receive your user ID and password.
Note: If you cannot locate either the license code or your user ID and password, contact Citrix Customer Service.

The host name of the NetScaler Gateway

The entry field for this name on My Citrix is case-sensitive, so make sure that you copy the host name exactly as it is configured on the NetScaler.

The number of licenses you want to include in the license file

You do not have to download all of the licenses you are entitled to at once. For example, if your company purchased 100 licenses, you can choose to download 50. You can allocate the rest in another license file at a later time. Multiple license files can be installed on the NetScaler Gateway.

Note: Before obtaining your licenses, make sure you configure the host name of the NetScaler using the Setup Wizard and then restart the NetScaler.

To obtain your universal license

1. In a Web browser, go to http://www.citrix.com/ and click My Citrix.
2. Enter your user name and password. If this is the first time you are logging on to the site, you are asked for additional background information.
3. Under My Tools, point to Choose a toolbox, and click Activation System/Manage Assets.
4. In the Current Tool drop-down menu, select Activate/Allocate and follow the directions to obtain your license file.

Installing the Universal License

To install the license, see "Installing the License". After installation, verify that the license was installed correctly.

Verifying Installation of the Universal License

Before proceeding, verify that your universal license is installed correctly.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To verify installation of the universal license by using the command line interface

1. Open an SSH connection to the NetScaler appliance by using an SSH client, such as PuTTY.
2. Log on to the NetScaler appliance by using the administrator credentials.
3. Use the show license command to verify that “SSL VPN = YES” and that Maximum Users has increased from 5 to the expected number of concurrent users.

To verify installation of the universal license by using the configuration utility

1. In a Web browser, type the IP address of the NetScaler appliance, such as http://192.168.100.1.
2. In User Name and Password, type the administrator credentials.
3. In the navigation pane, expand System, and then click Licenses.
4. In the Licenses pane, you will see a green check mark next to **NetScaler** Gateway. The field Maximum NetScaler Gateway Users Allowed displays the number of concurrent user sessions licensed on the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Upgrading or Downgrading the System Software

May 17, 2017

NetScaler 11.0 offers new and updated features with increased functionality. A comprehensive list of enhancements is listed in the release notes accompanying the release announcement. Take a moment to read this document before you upgrade your software.

It is important to understand the licensing framework and types of licenses before you upgrade your software. A software edition upgrade may require new licenses, such as upgrading from the standard edition to the enterprise edition, the standard edition to the platinum edition, or the enterprise edition to the platinum edition.

Note: For upgrading or downgrading the nodes in a cluster setup, see "Upgrading or Downgrading the Cluster Software". Upgrading from release 10.1 build 121.10 or any earlier releases to release 10.1 build 122.17 and later involves some location changes of user monitor script files. For details, see Directory Locations of Script Files for User Monitors.

This document includes the following information:

- Upgrading to Release 11.0
- Downgrading from Release 11.0

> **Note**
>
> For best practices for upgrading NetScaler appliances, read the article CTX126793.

Directory locations of script files for user monitors

In release 10.1 build 122.17, the script files for user monitors are at a new location. If you upgrade an appliance or virtual appliance to release 10.1 build 122.17 or later, the changes are as follows:

- A new directory named conflicts is created in /nsconfig/monitors/ and all the built-in scripts of the previous builds are moved to this directory.
- All new built-in scripts are available in the /netscaler/monitors/ directory. All custom scripts are available in the /nsconfig/monitors/ directory.
- You must save a new custom script in the /nsconfig/monitors/ directory.
- After the upgrade is completed, if a custom script is created and saved in the /nsconfig/monitors/ directory with the same name as that of a built-in script, the script in the /netscaler/monitors/ directory takes priority. That is, the custom script is not run.

If you provision a virtual appliance running release 10.1 build 122.17 or later, the changes are as follows:

- All built-in scripts are available in the /netscaler/monitors/ directory
- The directory /nsconfig/monitors/ is empty.
- If you create a new custom script, you must save it in the /nsconfig/monitors/ directory.

For more information about user monitors, see "Understanding User Monitors."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Before You Begin

May 28, 2017

Before you start the upgrade or downgrade process, make sure you check the guidelines given at this link.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Upgrading to Release 11.0

May 17, 2017

You follow the same basic procedure to upgrade either a standalone appliance or each appliance in a high availability pair, although additional considerations apply to upgrading a high availability pair.

Note: You cannot upgrade to NetScaler 11.0 from the following builds by using the Upgrade Wizard of the NetScaler GUI:

- All builds of NetScaler 9.3
- All builds of NetScaler 10.1
- Any build before Build 57.x of NetScaler 10.5

Workaround: Use the command line interface to upgrade the NetScaler appliance.

This document includes the following information:

- Upgrading a Standalone NetScaler
- Upgrading a High Availability Pair

Upgrading a Standalone NetScaler

Before upgrading the system software, make sure that you have the required licenses. For more information, see NetScaler Licensing Overview. You do not need a new license for the following upgrades:

- 8.x to 9.x
- 8.x or 9.x to 10.x
- 8.x, 9.x, or 10.x to 10.y

Note: When upgrading from release 8.0, 8.1, 9.0, 9.1, 9.2, 9.3, 10.0, 10.1, or 10.5, you have the option to use the configuration utility or the command line interface. Citrix recommends to perform the upgrade by using the command line interface. When using the upgrade wizard in the configuration utility to upgrade from release 8.0, do not use the Device option to upload your software.

Note: You cannot upgrade to NetScaler 11.0 from the following builds by using the Upgrade Wizard of the NetScaler GUI:

- All builds of NetScaler 9.3
- All builds of NetScaler 10.1
- Any build before Build 57.x of NetScaler 10.5

In the following procedure, <release> and <releasenumber> represent the release version you are upgrading to, and <targetbuildnumber> represents the build number that you are upgrading to. The procedure includes optional steps to avoid losing any updates that are pushed to the /etc directory during the upgrade.

To upgrade a standalone NetScaler appliance running release 8.0, 8.1, 9.0, 9.1, 9.2, 9.3, 10.0, 10.1, or 10.5 by using the command line interface

1. Use an SSH client, such as PuTTy, to open an SSH connection to the appliance.
2. Log on to the appliance by using the administrator credentials. Save the running configuration. At the prompt, type:save config
3. Create a copy of the ns.conf file. At the shell prompt, type:
   1. cd /nsconfig
   2. cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You should backup the configuration file to another computer.

4. (Optional) If you have modified some of the following files in the /etc directory, and copied them to /nsconfig to maintain persistency, any updates that are pushed to the /etc directory during the upgrade might be lost:
   - ttys
   - resolv.conf
   - sshd_config
   - host.conf
   - newsyslog.conf
   - host.conf
   - httpd.conf
   - rc.conf
   - syslog.conf
   - crontab
   - monitrc

   To avoid losing these updates, create a /var/nsconfig_backup directory, and move the customized files to this directory. That is, move any files that you modified in /etc directory and copied to /nsconfig by running the following command:

   cp /nsconfig/<filename> /var/nsconfig_backup

   Example:
   cp /nsconfig/syslog.conf /var/nsconfig_backup
5. Create a location for the installation package. At the shell prompt type
   1. cd /var/nsinstall
   2. mkdir <releasenumber>nsinstall
   3. cd <releasenumber>nsinstall
   4. mkdir build_<targetbuildnumber>
   5. cd build_<targetbuildnumber>
6. Download or copy the installation package (build-<release>-<targetbuildnumber>_nc.tgz) to the directory that you created for it. To download the installation package from the Citrix Web site, do the following:
   1. Go to MyCitrix.com, log on with your credentials, and click Downloads.
   2. In Select a Product, select NetScaler ADC.
   3. Under Firmware, click the release and build number to download.
   4. Click Get Firmware.
7. Extract the contents of the installation package. Example:
   tar –xvzf build-11.0-38.3_nc.tgz
8. Run the installns script to install the new version of the system software. The script updates the /etc directory. Example:
   ./installns
   Note:
   To install a FIPS appliance, run the installns script with the -F option.

   Warning: When upgrading to the NetScaler nCore build, the installation script prompts you to delete the /var directory if the swap partition is smaller than 32 gigabytes (GB). If this prompt appears, type N, save any important files located in /var to a backup location, and then re-run the installation script.
   If the free space available on the flash drive is insufficient to install the new build, the appliance prompts you to clean up the flash drive.
9. When prompted, restart the NetScaler.
10. (Optional) If you performed step 4, do the following:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

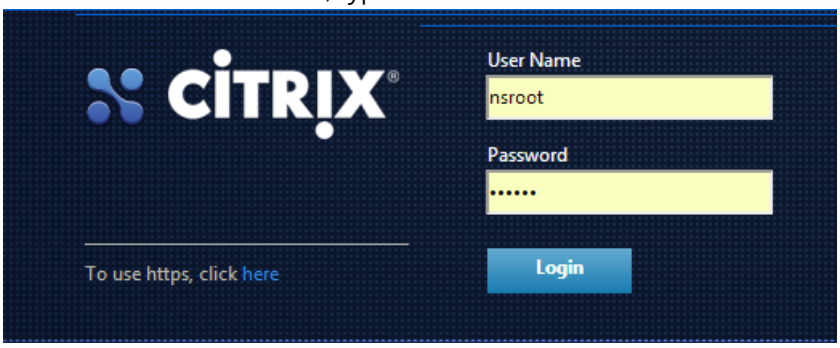
1. Manually compare the files in /var/nsconfig_backup and /etc and make appropriate changes in /etc.
2. To maintain persistency, move the updated files in /etc to /nsconfig.
3. Restart the appliance to put the changes into effect.

Example

```
login: nsroot
Password: nsroot
Last login: Fri Jun 12 19:22:51 2015 from 10.252.243.134
Done
> save config
> shell
Last login: Fri Jun 12 19:22:51 2015 from 10.252.243.134
root@NSnnn# cd /var/nsinstall
root@NSnnn# cd 11nsinstall
root@NSnnn# mkdir build_52
root@NSnnn# cd build_52
root@NSnnn# ftp <FTP server IP address>
ftp> mget build-11.0-52_nc.tgz
ftp> bye
root@NSnnn# tar xzvf build-11.0-52_nc.tgz
root@NSnnn# ./installns
installns version (11.0-52) kernel (ns-11.0-52_nc.gz)
...
...
...
Copying ns-11.0-52_nc.gz to /flash/ns-11.0-52_nc.gz ...

...
Installation has completed.

Reboot NOW? [Y/N] Y
```

To upgrade a standalone NetScaler running release 8.0, 8.1, 9.0, 9.1, 9.2, 9.3, 10.0, 10.1, or 10.5 by using the configuration utility

1. In a Web browser, type the IP address of the NetScaler, such as http://10.102.29.50.
2. In User Name and Password, type the administrator credentials and then click **Login**, as shown in the following figure.



3. In the configuration utility, in the navigation pane, click System.
4. In the System Overview page, click System Upgrade.
5. Follow the instructions to upgrade the software.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

6. When prompted, select Reboot.
   Note: After the upgrade, close all browser instances and clear your computer's cache before accessing the appliance.

Upgrading a High Availability Pair

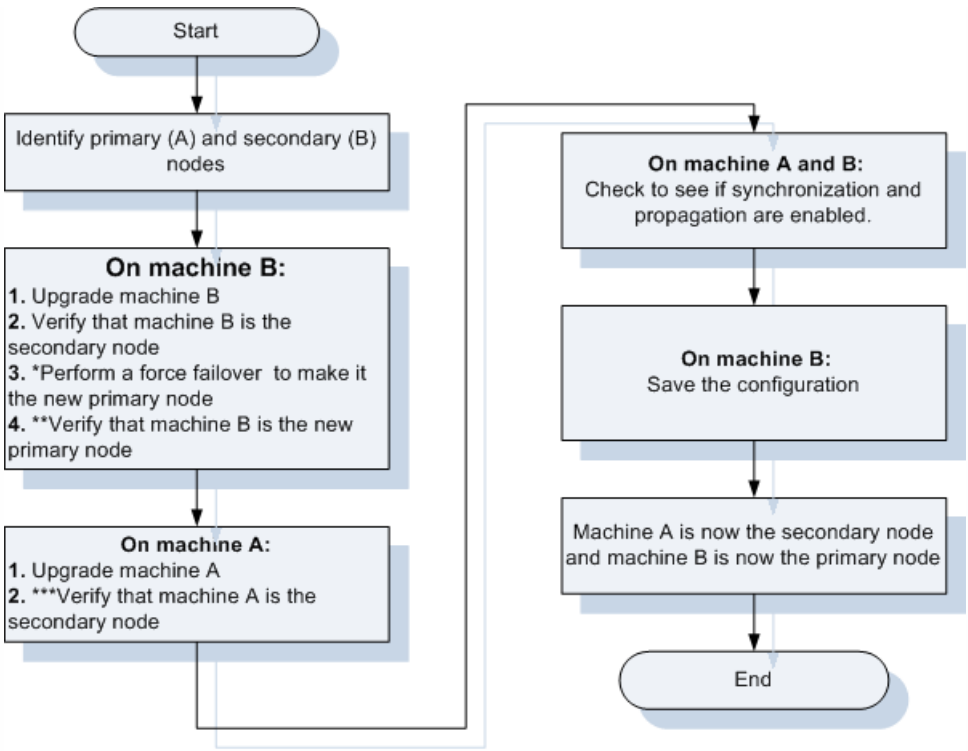
To upgrade the system software on NetScaler units in a high availability (HA) pair, first upgrade the secondary node, and then the primary node.

Points to Note

1. If the two nodes in an HA configuration are running different NetScaler software releases, the following information does not get synchronized on the primary and secondary nodes:
   -Configuration propagation and synchronization
   -States of the services
   -Connection failover sessions
   -Persistence sessions
   The above information might not get synchronized on the primary and secondary nodes if the two nodes are running different builds of the same release. Refer to the Known Issues section of the release notes to check if your NetScaler build has this issue.
2. Synchronization of the files in the All mode of the Sync HA files command works successfully if the two nodes in an HA configuration are running different NetScaler software releases, or the two nodes are running different builds of the same release. For more information, see Synchronising Configuration Files in High Availability Setup.

Figure 1. Upgrading a High Availability Pair

© 1999-2017 Citrix Systems, Inc. All rights reserved.



*After upgrading machine B, it becomes the primary node. If machine B does not function as expected, enter the force failover command on the new primary node (machine B) forcing it to again become the secondary node, and contact Citrix Customer Service before proceeding. After you test that machine B properly functions as the new primary node, proceed with upgrading the former primary node (machine A).

**After machine B is upgraded successfully, both synchronization and propagation are automatically disabled until you upgrade machine A.

***After both the nodes are upgraded successfully, synchronization and propagation are automatically enabled.

In the following procedure, machine A is the primary node and machine B is the secondary node before the upgrade.

To upgrade NetScaler units in a high availability pair running release 8.0, 8.1, 9.0, 9.1, 9.2, 9.3, 10.0, 10.1, or 10.5 by using the command line interface

On machine B (original secondary node)

1. Follow the procedure for upgrading a standalone node as described in "Upgrading a Standalone NetScaler Appliance". The procedure includes optional steps to avoid losing any updates that are pushed to the /etc directory during the upgrade.
2. After the appliance restarts, log on with the administrator credentials and enter the show ha node command to verify that the appliance is a secondary node.
3. Test the new build by entering the force failover command on the secondary node (machine B). At the command prompt type force failover.
   When you do so, machine B becomes the primary node. If machine B does not function as expected, enter the force failover command on the new primary node (machine B) forcing it to again become the secondary node, and contact Citrix Customer Service before proceeding.
4. Enter the show ha node command to verify that machine B is the new primary node.
   Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
login: nsroot
Password: nsroot
Last login: Mon Mar  26 08:37:26 2008 from 10.102.29.9
Done
show ha node
    2 nodes:
1)   Node ID:    0
    IP:     10.0.4.2
    Node State: UP
    Master State: Primary
    ...
    Sync State: AUTO DISABLED
    Propagation: AUTO DISABLED
    ...
Done
```

Note: After machine B is upgraded successfully, both synchronization and propagation are automatically disabled until you upgrade machine A.

On machine A (original primary node)

5. Follow the procedure for upgrading a standalone node as described in "Upgrading a Standalone NetScaler Appliance." The procedure includes optional steps to avoid losing any updates that are pushed to the /etc directory during the upgrade.
6. After the appliance restarts, log on by using the administrator credentials, and enter the show ha node command to verify that the appliance is a secondary node and that synchronization is disabled.
   Note: After both nodes are upgraded successfully, synchronization and propagation are automatically enabled.
   On machine A and machine B
7. After successfully upgrading both the nodes, run the show ha node command to verify that synchronization and propagation are enabled on the primary node and synchronization is successful and propagation is enabled on the secondary node.
   Example

   On Primary node (Machine B)

```
show ha node
    Node ID:    0
    IP:  10.0.4.2
    Node State: UP
    Master State: Primary
...
...
    INC State: DISABLED
    Sync State: ENABLED
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
...
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

...
Local node information
Critical Interfaces: 1/1
Done

On Secondary node (Machine A)
Show ha node
Node ID: 0
IP: 10.0.4.11
Node State: UP
Master State: Secondary
..
..
INC State: DISABLED
Sync State: SUCCESS
Propagation: ENABLED
Enabled Interfaces : 1/1
Disabled Interfaces : None
HA MON ON Interfaces : 1/1
. . .
. . .
Local node information:
Critical Interfaces: 1/1
Done
On machine B (new primary node)

8. Enter the save ns config command to save the configuration.

Machine B (original secondary node) is now the primary node and machine A (original primary node) is now the secondary node.

Note: You can enter the force failover command again to make machine A (original primary node) as the primary node and machine B (original secondary node) as the secondary node.

To upgrade NetScaler units in a high availability pair running release 8.0, 8.1, 9.0, 9.1, 9.2, 9.3, 10.0, 10.1, or 10.5 by using the configuration utility

1. Log on to the secondary node and perform the upgrade as described in "To upgrade a standalone NetScaler running release 8.0, 8.1, 9.0, 9.1, 9.2, 9.3, 10.0, 10.1, or 10.5 by using the configuration utility.
   Note: Before upgrading the primary node (machine A), you have the option to test the new release by entering the force failover command at the command line interface on the secondary node (machine B). When you do so, machine B becomes the primary node. If machine B does not function as expected, enter the force failover command at the command line interface on the new primary node (machine B) forcing it to again become the secondary node, and contact Citrix Customer Service before proceeding. If machine B properly assumes the role of primary node, proceed with upgrading the former primary node (machine A).
2. Log on to the primary node and perform the upgrade as described in "To upgrade a standalone NetScaler running release 8.0, 8.1, 9.0, 9.1, 9.2, 9.3, 10.0, 10.1, or 10.5 by using the configuration utility".

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Upgrading to a Later Build within Release 11.0

May 17, 2017

To upgrade from an earlier 11.0 build to a later 11.0 build on a standalone NetScaler appliance or a high availability pair, you can use the configuration utility or the command line interface. You use the same basic procedure to upgrade either a standalone appliance or each appliance in a high availability pair, although additional considerations apply to upgrading a high availability pair.

This document includes the following information:

- Upgrading a Standalone NetScaler Appliance to a Later Build
- Upgrading a NetScaler High Availability Pair to a Later Build

Upgrading a Standalone NetScaler Appliance to a Later Build

In the following procedure, <targetbuildnumber> is the build number that you are upgrading to within the 11.0 release. The procedure includes optional steps to avoid losing any updates that are pushed to the /etc directory during the upgrade.

To upgrade a standalone NetScaler appliance running release 11.0 to a later build by using the command line interface

1. Use an SSH client, such as PuTTy, to open an SSH connection to the appliance.
2. Log on to the appliance by using the administrator credentials, and save the running configuration. At the prompt, type:
   save ns config
3. Create a copy of the ns.conf file. At the shell prompt, type:
   1. cd /nsconfig
   2. cp ns.conf ns.conf.NS<releasenumber><currentbuildnumber>

   You should backup the configuration file to another computer.
4. (Optional) If you have modified any of the following files in the /etc directory, and copied them to /nsconfig to maintain persistency, any updates that are pushed to the /etc directory during the upgrade might be lost:
   - ttys
   - resolv.conf
   - sshd_config
   - host.conf
   - newsyslog.conf
   - host.conf
   - httpd.conf
   - rc.conf
   - syslog.conf
   - crontab
   - monitrc

   To avoid losing these updates, create a /var/nsconfig_backup directory, and move the customized files to this directory. That is, move any files that you modified in /etc directory and copied to /nsconfig, by running the following command:

   cp /nsconfig/<filename> /var/nsconfig_backup

   Example:
   cp /nsconfig/syslog.conf /var/nsconfig_backup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Create a location for the installation package. At the shell prompt, type:
   1. cd/var/nsinstall
   2. mkdir <releasenumber>nsinstall
   3. cd <releasenumber>nsinstall
   4. mkdir build_<targetbuildnumber>
   5. cd build_<targetbuildnumber>
6. Download or copy the installation package (build-11.0-<targetbuildnumber>_nc.tgz) to the directory that you created for it. To download the installation package from the Citrix Web site, do the following:
   1. Go to MyCitrix.com, log on with your credentials, and click Downloads.
   2. In the Select a Product, select NetScaler ADC.
   3. Under Firmware, click the release and build number to download.
   4. Click Get Firmware.

7. Extract the contents of the installation package. Example:
   tar –xvzf build_11.0-61.10_nc.tgz
8. Run the installns script to install the new version of the system software. The script updates the /etc directory.
   Note:
   To install a FIPS appliance, run the installns script with the -F option.

   During the upgrade, you are prompted for an option to load a different configuration.

   If you do not want to load a different configuration and continue with the upgrade, type N. If want to load a different configuration file, then type Y.
   If the configuration file for the build to which you are upgrading exists in the appliance, you are prompted to load that configuration.

9. When prompted, restart the appliance.
10. (Optional) If you performed step 4, do the following:
    1. Manually compare the files in /var/nsconfig_backup and /etc and make appropriate changes in /etc.
    2. To maintain persistency, move the updated files in /etc to /nsconfig.
    3. Restart the appliance to put the changes into effect.

Example

```
login: nsroot
Password:
Last login: Thu Aug  1 12:12:54 2015 from 10.144.7.22
Done
> save ns config
> shell
Last login: Mon Aug  1 03:51:42 from 10.103.25.64
root@NSnnn# cd /var/nsinstall
root@NSnnn# cd 11.0nsinstall
root@NSnnn# mkdir build_60.10
root@NSnnn# cd build_60.10
root@NSnnn# ftp <FTP server IP address>
ftp> mget build-11.0-60.10_nc.tgz
ftp> bye
root@NSnnn# tar  build-11.0-61.10_nc.tgz
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
root@NSnnn# ./installns
installns version (11.0-61.10) kernel (ns-11.0-61.10_nc.gz)
The Netscaler version 11.0-61.10 checksum file is located on
http://www.mycitrix.com under Support > Downloads > Citrix NetScaler.
Select the Release 11.0-61.10 link to view the MD5 checksum file for build 11.0-61.10.

There may be a pause of up to 3 minutes while data is written to the flash.
Do not interrupt the installation process once it has begun....
...
...
Copying ns-11.0-61.10_nc.gz to /flash/ns-11.0-61.10_nc.gz ...
...
Installation has completed.

Reboot NOW? [Y/N] Y
```

To upgrade a standalone NetScaler running release 11.0 to a later build by using the configuration utility

1. In a web browser, type the IP address of the NetScaler, such as http://10.102.29.50.
2. In User Name and Password, type the administrator credentials.
3. In the configuration utility, in the navigation pane, click System.
4. In the System Overview page, click System Upgrade.
5. Follow the instructions to upgrade the software.
6. When prompted, select Reboot.
   Note: After the upgrade, close all browser instances and clear your computer's cache before accessing the appliance.

Upgrading a NetScaler High Availability Pair to a Later Build

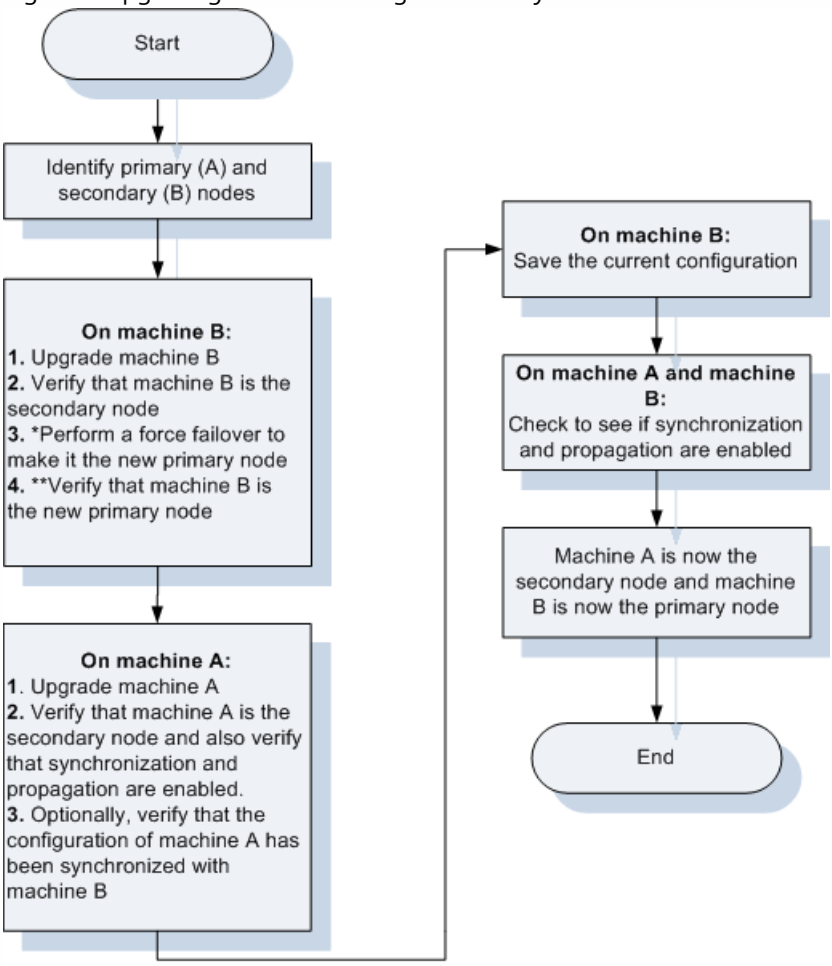
To upgrade the system software on NetScaler appliances in a high availability (HA) pair, upgrade the secondary node first, and then upgrade the primary node.
Warning: In certain cases, after you upgrade one of the nodes in an HA pair, synchronization and propagation are automatically disabled until you upgrade the other node. To determine whether synchronization and propagation are disabled, at the command line interface, type: show ha node

Points to Note

1. If the two nodes in an HA configuration are running different NetScaler software releases, the following information does not get synchronized on the primary and secondary nodes:
   -Configuration propagation and synchronization
   -States of the services
   -Connection failover sessions
   -Persistence sessions
   The above information might not get synchronized on the primary and secondary nodes if the two nodes are running different builds of the same release. Refer to the Known Issues section of the release notes to check if your NetScaler build has this issue.
2. Synchronization of the files in the All mode of the Sync HA files command works successfully if the two nodes in an HA configuration are running different NetScaler software releases, or the two nodes are running different builds of the same release. For more information, see Synchronising Configuration Files in High Availability Setup.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Figure 3. Upgrading a NetScaler High Availability Pair to a Later Build



*After upgrading machine B, it becomes the primary node. If machine B does not function as expected, enter the force failover command on the new primary node (machine B) forcing it to again become the secondary node, and contact Citrix Customer Service before proceeding. After you test that machine B properly functions as the new primary node, proceed with upgrading the former primary node (machine A).

**After machine B is upgraded successfully, both synchronization and propagation are automatically disabled until you upgrade machine A.

In the following procedure, machine A is the original primary and machine B is the original secondary node, and <targetbuildnumber> is the build number that you are upgrading to within the 10.1 release.

To upgrade a NetScaler high availability pair to a later build by using the command line interface

On machine B (original secondary node)

1. Follow the procedure for upgrading a standalone node as described in "Upgrading a Standalone NetScaler Appliance to a Later Build". The procedure includes optional steps to avoid losing any updates that are pushed to the /etc directory during the upgrade.
2. After the NetScaler restarts, log on by using the administrator credentials and enter the show ha node command to verify that the appliance is a secondary node.
3. Test the new build by entering the force failover command on the secondary node (machine B). At the command prompt type force failover.
   When you do so, machine B becomes the primary node. If machine B does not function as expected, enter the force

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

failover command on the new primary node (machine B) forcing it to again become the secondary node, and contact Citrix Customer Service before proceeding.

4. Enter the show ha node command to verify that machine B is the new primary node.
   On machine A (original primary node)

5. Follow the procedure for upgrading a standalone node as described in "Upgrading a Standalone NetScaler Appliance to a Later Build." The procedure includes optional steps to avoid losing any updates that are pushed to the /etc directory during the upgrade.
6. After the appliance restarts, log on by using the administrator credentials and enter the show ha node command to verify that the appliance is a secondary node and that synchronization and propagation are enabled. Optionally, enter the show ns runningconfig command on both the nodes and compare the result to verify that the configuration of machine A has been synchronized with that of machine B.
   On machine B (new primary node)

7. Enter the save ns config command to save the current configuration.
   On machine A and machine B

8. After successfully upgrading both the nodes, run the show ha node command to verify that synchronization and propagation are enabled.
   Example

```
show ha node
    Node ID:    0
    IP:  10.0.4.2
    Node State: UP
    Master State: Primary
...
...
    INC State: DISABLED
    Sync State: ENABLED
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
...
...
    Local node information
    Critical Interfaces: 1/1
Done


Show ha node
    Node ID:    0
    IP:  10.0.4.11
    Node State: UP
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        Master State: Secondary
        ..
       ..
        INC State: DISABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 1/1
        Disabled Interfaces : None
        HA MON ON Interfaces : 1/1
        . . .
        . . .
        Local node information:
        Critical Interfaces: 1/1
    Done
```

Machine B (original secondary node) is now the primary node and machine A (original primary node) is now the secondary node.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Downgrading from Release 11.0

May 17, 2017

You can downgrade to any release on a standalone NetScaler or a high availability pair by using the command line interface.

Caution: Loss in configuration may occur when downgrading. You should compare the configurations before and after the downgrade, and then manually reenter any missing entries.

This procedure provides steps to downgrade from release 11.0 to an earlier release.

Note: Downgrading using the configuration utility is not supported.

This document includes the following information:

- Downgrading a Standalone NetScaler
- Downgrading a High Availability Pair

Downgrading a Standalone NetScaler

In the following procedure, <release> and <releasenumber> represent the release version you are downgrading to, and <targetbuildnumber> represents the build number that you are downgrading to. Refer to the table below for specific values.

To downgrade a standalone NetScaler

1. Open an SSH connection to the NetScaler appliance by using an SSH client, such as PuTTY.
2. Log on to the NetScaler appliance by using the administrator credentials. Save the running configuration. At the prompt, type:
   save config
3. Create a copy of the ns.conf file. At the shell prompt, type:
   1. cd /nsconfig
   2. cp ns.conf ns.conf.NS<currentbuildnumber>

   You should backup a copy of the configuration file on another computer.
4. Copy the <releasenumber> configuration file (ns.conf.NS<releasenumber>) to ns.conf. At the shell prompt, type:
   cp **ns.conf.NS**<releasenumber> **ns.conf**

   Note: ns.conf.NS<releasenumber> is the backup configuration file that is automatically created when the system software is upgraded from release version <releasenumber> to the current release version. There may be some loss in configuration when downgrading. After the appliance restarts, compare the configuration saved in step 3 with the running configuration, and make any adjustments for features and entities configured before the downgrade. Save the running configuration after making the changes.

   Important: If routing is enabled, perform step 5. Otherwise, skip to step 6.
5. If routing is enabled, the ZebOS.conf file will contain the configuration. At the shell prompt, type:
   1. cd **/nsconfig**
   2. cp **ZebOS.conf ZebOS.conf.NS**
   3. cp ZebOS.conf.NS<targetreleasenumber> ZebOS.conf
6. Change directory to /var/nsinstall/<releasenumber>nsinstall, or create one if it does not exist.
7. Change directory to build_<targetbuildnumber>, or create one if it does not exist.
8. Download or copy the installation package (build-<release>-<targetbuildnumber>.tgz) to this directory and extract the contents of the installation package.
9. Run the installns script to install the new version of the system software. The script updates the /etc directory.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If the configuration file for the build that you are downgrading to exists on the appliance, you are prompted to load that configuration, as shown in the following figure.

Figure 1. Downgrade menu if configuration file exists

```
 version build        size  last modified  file name
Copied to ns.conf     72545  Jun 18 04:42   ns.conf.NS10.1-112.13
  NS10.1  112.13      72545  Jun 18 04:42   ns.conf.NS10.1
  NS10.1  112.13      72545  Jun 18 04:42   ns.conf.4
  NS10.1  109.1       87219  Jun 18 04:42   ns.conf.NS10.1-109.1
  NS10.1  93.051     74443  Jun 18 04:42   ns.conf.NS10.1-93.051
  NS10.0  29.1.      62849  Jun 18 04:42   ns.conf.NS10.0-29.1.

Listed above are 5 configuration files, found in /nsconfig, that are
appropriate for use with build 112.13.

Use the arrow keys to select an item in the menu above, then type:
  'c' - copy file over ns.conf
  'v' - view file (with vi; type ':q!' to exit vi)
  '>' - more files
  '<' - fewer files
  'd' - done
```

If the free space available on the flash drive is insufficient to install the new build, the NetScaler prompts you to clean up the flash drive.

10. When prompted, restart the NetScaler.

Example

login: nsroot
Password: nsroot
Last login: Thu Jun 18 02:06:52 2015 from 10.102.29.9

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done
> save config
> shell
root@NSnnn# cp ns.conf.NS10.5 ns.conf
root@NSnnn# cd /var/nsinstall
root@NSnnn# mkdir 10.5nsinstall
root@NSnnn# cd 10.5nsinstall
root@NSnnn# mkdir build_57
root@NSnnn# cd build_57
root@NSnnn# ftp 10.102.1.1
ftp> mget build-10.5-57_nc.tgz
ftp> bye
root@NSnnn# tar -xzvf build-10.1-125_nc.tgz
root@NSnnn# ./installns
installns version (10.5-57) kernel (ns-10.5-57.gz)
...
...
...
Copying ns-10.5-57.gz to /flash/ns-10.5-57_nc.gz ...
Changing /flash/boot/loader.conf for ns-10.5-57 ...

Installation has completed.

Reboot NOW? [Y/N] Y
```

Downgrading a High Availability Pair

To downgrade the system software on NetScaler units in a high availability pair, you need to downgrade the software first on the secondary node and then on the primary node. For instructions on downgrading each node separately, see "Downgrading a Standalone NetScaler".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Downgrading to an Earlier Build within Release 11.0

Jan 29, 2014

You can downgrade from a later 11.0 build to an earlier 11.0 build on a standalone NetScaler or a high availability pair. This procedure must be performed by using the command line interface.

Warning: Loss in configuration may occur when downgrading. You should compare the configurations before and after the downgrade, and then manually readd any missing entries.

This document includes the following information:

- Downgrading a Standalone NetScaler to an Earlier Build
- Downgrading a NetScaler High Availability Pair to an Earlier Build

Downgrading a Standalone NetScaler to an Earlier Build

In the procedure below, <targetbuildnumber> is the build number that you are downgrading to within the same release.

To downgrade a standalone NetScaler to an earlier build

1. Use an SSH client, such as PuTTy, to open an SSH connection to the appliance.
2. Log on to the NetScaler by using the administrator credentials. Save the running configuration. At the prompt, type:
   save ns config

   Caution: If ns.conf.NS11.0-<targetbuildnumber> does not exist, loss in configuration may occur when downgrading to an earlier build. The errors and warnings appear only on the console. Please watch the console closely for these errors and warnings. After the appliance restarts, compare the saved configuration with the running configuration, and make any adjustments for features and entities configured before the downgrade. Save the running configuration after making the changes.
3. Change directory to /var/nsinstall/11.0nsinstall.
4. Change directory to build_<targetbuildnumber>, or create one if it does not exist.
5. Download or copy the installation package (build-11.0-<targetbuildnumber>_nc.tgz) to this directory and extract the contents of the installation package.
6. Run the installns script to install the old version of the system software. The script updates the /etc directory.
   If the configuration file for the build that you are downgrading to exists on the appliance, you are prompted to load that configuration.

7. When prompted, restart the NetScaler.

Example

```
login: nsroot
Password: nsroot
Last login: Sun Aug  3 08:38:25 2015 from 10.102.29.4
Done
> save ns config
> shell
Last login: Sun Aug 3  09:07:06 from 10.103.25.64
root@NSnnn# cp ns.conf.NS11.0-55.23 ns.conf
root@NSnnn# cd /var/nsinstall
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
root@NSnnn# cd 11.0nsinstall
root@NSnnn# cd build_55_23
root@NSnnn# ftp <FTP server IP address>
ftp> mget build-11.0-55.23_nc.tgz
ftp> bye
root@NSnnn# tar xzvf build-11.0-55.23_nc.tgz
root@NSnnn# ./installns
installns version (11.0-55.23) kernel (ns-11.0-55.23.gz)
...
...
...
Copying ns-11.0-55.23_nc.gz to /flash/ns-11.0-55.23_nc.gz ...
Changing /flash/boot/loader.conf for ns-11.0-55.23 ...

Installation has completed.

Reboot NOW? [Y/N] Y
```

Downgrading a NetScaler High Availability Pair to an Earlier Build

To downgrade the system software on NetScaler units in a high availability pair, you need to downgrade the software first on the secondary node and then on the primary node. For instructions on downgrading each node separately, see "Downgrading a Standalone NetScaler to an Earlier Build".

Note: In an HA setup, both nodes must run NetScaler nCore or NetScaler classic. If the nodes are running NetScaler classic and you want to migrate to NetScaler nCore of the same NetScaler release, propagation and synchronization are not supported during the migration process. Once migration is complete, you have to manually enable propagation and synchronization. The same applies if you migrate from NetScaler nCore to NetScaler classic.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Aug 13, 2013

If the appliance does not work as expected after you complete the installation, upgrade, or downgrade process, the first thing to do is to check for the most common causes of the problem.

This document includes the following information:

- Resources for Troubleshooting
- Troubleshooting Issues Related to the Installation, Upgrade, and Downgrade processes

Resources for Troubleshooting

Updated: 2013-08-13

For best results, use the following resources to troubleshoot an issue related to installing, upgrading, or downgrading a NetScaler appliance:

- The configuration files from the appliance. In case of a High Availability pair, the configuration files from both appliances.
- The following files from the appliance(s):
  - The relevant newnslog files.
  - The ns.log file.
  - The messages file.
- A network topology diagram.

Troubleshooting Issues Related to the Installation, Upgrade, and Downgrade processes

Updated: 2013-11-22

Following are the most common installation, upgrade, and downgrade issues, and tips for resolving them:

- **Issue**
  The NetScaler appliance is not accessible after the software downgrade

  Cause

  During the software downgrade process, if the configuration file of the existing release and build does not match the configuration file of the earlier release and build, the appliance cannot load the configuration, and the default IP address is assigned to the appliance.

  Resolution

  - Verify that the appliance is accessible from the console.
  - Verify the NSIP address and the routes on the appliance.
    - If the IP address has changed to the default 192.168.100.1 IP address, change the IP address as required.
    - Verify that the appliance is accessible.
- **Issue**
  Configuration of the appliance is lost after you upgrade the software across multiple releases.

  Cause

  Some migration commands are built in for upgrading to the next release. Such commands might not be available across releases.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Resolution

- Verify the path of the upgrade process. Citrix recommends upgrading by one release at a time. For example, if the softer needs to be upgraded from NetScaler release 9.2 to NetScaler release 10.1, the following is the recommended path for the upgrade:
  - NetScaler release 9.2 to NetScaler release 9.3
  - NetScaler release 9.3 to NetScaler release 10
  - NetScaler release 10 to NetScaler release 10.1
- Verify that the appliance has appropriate license files.
- Verifying the configuration at each step of the upgrade process can give you pointers to the issue.

- **Issue**

  During an upgrade, if I run the command for synchronizing, the following message appears:

  Command failed on the secondary node but succeeded on the primary node.

  Resolution

  Do not run any dependent commands (set /unset /bind /unbind) when High Availability (HA) synchronization is in progress.

- **Issue**

  During an upgrade process, traffic does not pass through the new primary node when you run the force failover command.

  Resolution

  - Check for problems with the network topology and the switch configurations.
  - Run the set L2param -garpreply ENABLED command to enable the GARP reply.
  - Try using VMAC if not already used.
  - Run the sendarp –a command from the primary node.

- **Issue**

  In an HA pair, after you run the force HA failover command the devices keep rebooting. The secondary device does not come up after an upgrade.

  Resolution

  Check to see if the /var directory is full. If so, remove the old installation files. Run the df –h command to show the available disk space.

- **Issue**

  After upgrading an HA pair, one of the nodes is listed as state UNKNOWN.

  Resolution

  - Check to see if both nodes are running the same build. If the builds are not same and HA nodes have a version mismatch, some of the fields are shown as UNKNOWN when you run the show ha node command.
  - Check to see if the secondary appliance is reachable.

- **Issue**

  After you upgrade the NetScaler appliance, the interface shows most of the load balancing virtual servers and services are DOWN.

  Resolution

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Verify that the SNIP address is active on the secondary appliance. Also, type the show service command to see if the service is running.

- **Issue**
  After performing an upgrade, all virtual servers are down on the secondary appliance.

  Resolution

  Enable the HA state and HA synchronization by running the following commands:
  - set node hastate enable
  - set node hasync enable

  Disabling HA is not recommended.
- **Issue**
  After performing a downgrade, the NetScaler appliance does not boot up properly.

  Resolution

  Check to see if the correct license has been installed.

- **Issue**
  In an HA pair, some features do not get synchronized after an upgrade is performed.

  Resolution

  Run the sync ha file misc command to synchronize the configurations files from the primary node to secondary node.

- **Issue**
  During reboot, the following error message appears:

  One or more commands in ns.conf failedWhat should I do?

  Resolution

  Make sure that no command in the ns.conf file exceeds the 255 byte limit. In commands that create policies that are too long for the 255-byte limit, you can use pattern sets to shorten the policies.

  Example:
  add cs policy p11 -rule 'HTTP.REQ.URL.ENDSWITH_ANY("ctx_file_extensions")'
  Done
  ctx_file_extensions is a default patset that covers a large number of extensions. In addition to the default pattern sets, you can create user-defined pattern sets. Add a patset by running the following command:
  add patset <name>
  Note: Patsets are supported only in release 9.3 or later.
- **Issue**
  When upgrading a NetScaler VPX appliance, I am told to free up space in /var. What files do I remove?

  Resolution

  Remove the old installation files from /var/tmp/ directory. Also remove unwanted files from /flash.

- **Issue**
  There is no connectivity to the graphical user interface (GUI) when you run the force HA failover command on the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

secondary appliance.

Resolution

Log on to the secondary appliance using the command line interface and enable the access to GUI by running the set ns ip <IP> -gui enabled command.

- **Issue**
  After performing an upgrade, and when I click on any link on the GUI that has to load a java applet (Upgrade Wizard or license Wizard), the following error message appears: **GUI version does not match with the kernel version. Please close this instance, clear java plug-in cache and reopen**.

  Resolution

  - Log on to the NetScaler appliance using the GUI.
  - Navigate to NetScaler Gateway > Global Settings.
  - Click Change Global Settings under Settings.
  - In the details pane, under Client Experience, select Default from the UI theme list.
  - Click OK.

Note: These troubleshooting steps also apply to issues with configuration loss when downgrading the software across multiple releases.
For any other issue, see the release notes, Knowledge Center articles, and FAQs.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

System

May 13, 2016

This section provides system-level information of the NetScaler ADC. This includes a detailed explanation of system-level features, the scenarios in which the features can be used, the configuration steps, and examples to help you better understand the features.

- Basic Operations
- Admin Partitioning
- Authentication and Authorization
- TCP Configurations
- HTTP Configurations
- SNMP
- Audit Logging
- Web Server Logging
- Call Home
- Reporting Tool
- AppFlow
- AutoScale
- Clustering
- CloudBridge Connector
- High Availability

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Basic Operations

Jun 02, 2015

Any changes you make to the configuration of a NetScaler appliance are not saved automatically. You have to save the settings manually. When an appliance is restarted, it loads the latest saved configuration.

This document includes the following details:

- Viewing and Saving Configurations
- Clearing the NetScaler Configuration

Viewing and Saving Configurations

Configurations are stored in the /nsconfig/ns.conf directory. For configurations to be available across sessions, you must save the configuration after every configuration change.

To view the running configuration by using the command line interface

At the command prompt, type:

show ns runningConfig

To view the running configuration by using the configuration utility

- Navigate to System > Diagnostics and, in the View Configuration group, click Running Configuration.

To find the difference between two configuration files by using the command line interface

At the command prompt, type:

diff ns config <configfile1> <configfile2>

To find the difference between two configuration files by using the configuration utility

- Navigate to System > Diagnostics and, in the View Configuration group, click Configuration difference.

To save configurations by using the command line interface

At the command prompt, type:

save ns config

To save configurations by using the configuration utility

On the Configuration tab, in the top-right corner, click the Save icon.

To view the saved configurations by using the command line interface

At the command prompt, type:

show ns ns.conf

To view the saved configurations by using the configuration utility

Navigate to System > Diagnostics and, in the View Configuration group, click Saved Configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Clearing the NetScaler Configuration

You have the following three options for clearing the NetScaler configuration.

Basic level. Clearing your configuration at the basic level clears all settings except the following:

- NSIP, MIP(s), and SNIP(s)
- Network settings (Default Gateway, VLAN, RHI, NTP, and DNS settings)
- HA node definitions
- Feature and mode settings
- Default administrator password (nsroot)

Extended level. Clearing your configuration at the extended level clears all settings except the following:

- NSIP, MIP(s), and SNIP(s)
- Network settings (Default Gateway, VLAN, RHI, NTP, and DNS settings)
- HA node definitions

Feature and mode settings revert to their default values.

Full level. Clearing your configuration at the full level returns all settings to their factory default values. However, the NSIP and default gateway are not changed, because changing them could cause the appliance to lose network connectivity.

To clear the configuration by using the command line interface

At the command prompt, type:

clear ns config -force <level>

Example: To forcefully clear the basic configurations on an appliance.

```
clear ns config -force basic
```

To clear the configuration by using the configuration utility

- Navigate to System > Diagnostics and, in the Maintenance group, click Clear Configuration and select the configuration level to be cleared from the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Clock Synchronization

Jun 02, 2015

You can configure your NetScaler appliance to synchronize its local clock with a Network Time Protocol (NTP) server. This ensures that its clock has the same date and time settings as the other servers on your network.

You can configure clock synchronization on your appliance by adding NTP server entries to the ntp.conf file from either the configuration utility or the command line interface, or by manually modifying the ntp.conf file and then starting the NTP daemon (NTPD). The clock synchronization configuration does not change if the appliance is restarted, upgraded, or downgraded. However, the configuration does not get propagated to the secondary NetScaler in a high availability setup.

Note: If you do not have a local NTP server, you can find a list of public, open access, NTP servers at the official NTP site, http://www.ntp.org, under Public Time Servers List. Before configuring your NetScaler to use a public NTP server, be sure to read the Rules of Engagement page (link included on all Public Time Servers pages).
In NetScaler release 11, the NTP version has been updated from 4.2.6p3 to 4.2.8p2.
This document includes the following details:

- Setting Up Clock Synchronization
- Starting the NTP Daemon
- Configuring Clock Synchronization Manually

Setting Up Clock Synchronization

To configure clock synchronization, you must add NTP servers and then enable NTP synchronization.

To add an NTP server by using the command line interface

At the command prompt, type the following commands to add an NTP server and verify the configuration:

- add ntp server (<serverIP> | <serverName>) [-minpoll <positive_integer>] [-maxpoll <positive_integer>]
- show ntp server

Example

```
> add ntp server 10.102.29.30 -minpoll 6 -maxpoll 11
```

To configure an NTP server by using the configuration utility

Navigate to System > NTP Servers, and create the NTP server.

Starting the NTP Daemon

When you enable NTP synchronization, the NetScaler starts the NTP daemon and uses the NTP server entries in the ntp.conf file to synchronize its local time setting. If you do not want to synchronize the appliance time with the other servers in the network, you can disable NTP synchronization, which stops the NTP daemon (NTPD).

To enable NTP synchronization by using the command line interface

At the command prompt, type one of the following commands:

enable ntp sync

To enable NTP synchronization by using the configuration utility

Navigate to System > NTP Servers, click Action and select NTP Synchronization.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Clock Synchronization Manually

You can configure clock synchronization manually by logging on to the NetScaler appliance and editing the ntp.conf file.

To enable clock synchronization on your NetScaler appliance by modifying the ntp.conf file

1. Log on to the command line interface.
2. Switch to the shell prompt.
3. Copy the /etc/ntp.conf file to /nsconfig/ntp.conf, unless the /nsconfig directory already contains an ntp.conf file.
4. For each NTP server you want to add, you must add the following two lines to the /nsconfig/ntp.conf file:

   server <IP address for NTP server> **iburst**

   restrict <IP address for NTP server> **mask** <netmask> **nomodify notrap nopeer noquery**

   Note: For security reasons, there should be a corresponding restrict entry for each server entry.

   Example
   In the following example, an administrator has inserted # characters to “comment out” an existing NTP entry, and then added a new entry:

   #server 1.2.3.4 iburst

   #restrict 1.2.3.4 mask 255.255.255.255 nomodify notrap nopeer noquery

   server 10.102.29.160 iburst

   restrict 10.102.29.160 mask 255.255.255.255 nomodify notrap nopeer noquery

5. If the /nsconfig directory does not contain a file named rc.netscaler, create the file.

6. Add the following entry to /nsconfig/rc.netscaler: **/bin/sh /etc/ntpd_ctl full_start**

   This entry starts the ntpd service, checks the ntp.conf file, and logs messages in the /var/log directory.

   This process runs every time the NetScaler is restarted.

7. Restart the NetScaler appliance to enable clock synchronization. Or, to start the time synchronization process without restarting the appliance, enter the following commands at the shell prompt:

- rm /etc/ntp.conf
- ln -s /nsconfig/ntp.conf /etc/ntp.conf
- /bin/sh /etc/ntpd_ctl full_start

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring System Session Timeout

Sep 12, 2014

A session timeout interval is provided to restrict the time duration for which a session (GUI, CLI, or API) remains active when not in use. For the NetScaler, the system session timeout can be configured at the following levels:

- **User level timeout.** Applicable to the specific user.

GUI	Navigate to System > User Administration > Users, select a user, and edit the user's timeout setting.
CLI	At the command prompt, enter the following command: set system user <name> -timeout <secs>

- **User group level timeout.** Applicable to all users in the group.

GUI	Navigate to System > User Administration > Groups, select a group, and edit the group's timeout setting.
CLI	At the command prompt, enter the following command: set system group <groupName> -timeout <secs>

- **Global system timeout.** Applicable to all users and users from groups who do not have a timeout configured.

GUI	Navigate to System > Settings, click Change global system settings, and update the timeout value as required.
CLI	At the command prompt, enter the following command: set system parameter -timeout <secs>

The timeout value specified for a user has the highest priority. If timeout is not configured for the user, the timeout configured for a member group is considered. If timeout is not specified for a group (or the user does not belong to a group), the globally configured timeout value is considered. If timeout is not configured at any level, the default value of 900 seconds is set as the system session timeout.

Additionally, you can specify timeout durations for each of the interfaces you are accessing. However, the timeout value specified for a specific interface is restricted to the timeout value configured for the user that is accessing the interface. For example, let us consider an user "publicadmin" who has a timeout value of 20 minutes. Now, when accessing an interface, the user must specify a timeout value that is within 20 minutes.

Note: You can choose to keep a check on the minimum and maximum timeout values by specifying the timeout as restricted (in CLI by specifying the restrictedTimeout parameter). This parameter is provided to account for previous NetScaler versions where the timeout value was not restricted.

- When enabled, the minimum configurable timeout value is 5 minutes (300 secs) and the maximum value is 1 day (86400 secs). If the timeout value is already configured to a value larger than 1 day, when this parameter is enabled, you are prompted to change it. If you do not change the value, the timeout value will automatically be reconfigured to the default timeout duration of 15 minutes (900 secs) on next reboot. The same will happen is the configured timeout value is less than 5 minutes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- When disabled, the configured timeout durations are considered.

To configure the timeout duration at each interface:

NetScaler user interface	Timeout configuration
CLI	Specify the timeout value on the command prompt by using the following command: set cli mode -timeout <secs>
API	Specify the timeout value in the login payload.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing the System Date and Time

Aug 08, 2014

To change the system date and time, you must use the shell interface to the underlying FreeBSD OS. However, to view the system date and time, you can use the command line interface or the configuration utility.

To view the system date and time by using the command line interface

At the command prompt, type:

```
show ns config
```

To view the system date and time by using the configuration utility

Navigate to System and select the System Information tab to view the system date.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Backing up and Restoring the NetScaler Appliance

Sep 15, 2015

You can back up the current state of a NetScaler appliance, and later use the backed up files to restore the appliance to the same state. You must use this feature before performing an upgrade or for precautionary reasons. A backup of a stable system enables you to restore the system to a stable point in the event that it becomes unstable.

Points to remember

- Sysid and Interfaces should be same.
- Network configuration should be supported on New Platform.
- New Platform build should is either same of backup file or higher version than it.

This document includes the following details:

- Backing up a NetScaler Appliance
- Restoring the NetScaler Appliance

Backing up a NetScaler Appliance

Depending on the type of data to be backed up and the frequency at which you will create a backup, you can take a basic backup or a full backup.

- **Basic backup.** Backs up only configuration files. You might want to perform this type of backup frequently, because the files it backs up change constantly. The files that are backed up are:

Directory	Sub-Directory or Files
/nsconfig/	• ns.conf • ZebOS.conf • rc.netscaler • snmpd.conf • nsbefore.sh • nsafter.sh • inetd.conf • ntp.conf • syslog.conf • newsyslog.conf • crontab • host.conf • hosts • ttys • sshd_config • httpd.conf • monitrc • rc.conf • ssh_config • localtime • issue

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Directory	Sub-Directory or Files
	• issue.net
/var/	• download/* • log/wicmd.log • wi/tomcat/webapps/* • wi/tomcat/logs/* • wi/tomcat/conf/catalina/localhost/* • nslw.bin/etc/krb.conf • nslw.bin/etc/krb.keytab • netscaler/locdb/* • lib/likewise/db/* • vpn/bookmark/* • netscaler/crl • nstemplates/* • learnt_data/*
/netscaler/	• custom.html • vsr.htm

- **Full backup.** In addition to the files that are backed up by a basic backup, a full backup backs up some less frequently updated files. The files that are backed up when using the full backup option are:

Directory	Sub-Directory or Files
/nsconfig/	• ssl/* • license/* • fips/*
/var/	• netscaler/ssl/* • wi/java_home/jre/lib/security/cacerts/* • wi/java_home/lib/security/cacerts/*

The backup is stored as a compressed TAR file in the /var/ns_sys_backup/ directory. To avoid issues due to non-availability of disk space, you can store a maximum of 50 backup files in this directory. You can use the rm system backup command to delete existing backup files so that you can create more backups.

Note:

- While the backup operation is in progress, do not execute commands that affect the configuration.
- If a file that is required to be backed up is not available, the operation skips that file.

To backup the NetScaler by using the NetScaler command line interface

At the command prompt, do the following:

1. Save the NetScaler configurations.
   save ns config
2. Create the backup file.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

create system backup [<fileName>] -level <basic ¦ full> -comment <string>

Note: If the file name is not specified, the appliance creates a TAR file with the following naming convention: backup_<level>_<nsip_address>_<date-timestamp>.tgz.
Example: To backup the full appliance using the default naming convention for the backup file.
> create system backup -level full

3. Verify that the backup file was created.
   show system backup

   You can view properties of a specific backup file by using the fileName parameter.

To backup the NetScaler by using the configuration utility

Navigate to System > Backup and Restore, click Backup and then specify the details of the backup.

Restoring the NetScaler Appliance

When you restore the appliance from a backup file, the restore operation untars the backup file into the /var/ns_sys_backup/ directory. Once the untar operation is complete, the files are copied to their respective directories.

Attention: The restore operation does not succeed if the backup file is renamed or if the contents of the file are modified.

To restore the NetScaler from a local backup file by using the command line interface

Note: Citrix recommends backing up the current configuration before restoring a previous configuration. However, if you do not want the restore command to automatically create a backup of the current configuration, use the **–skipBackup** parameter.

At the command prompt, do the following:

1. Obtain a list of the backup files available on the appliance.
   show system backup

2. Restore the appliance by specifying one of the backup files.
   restore system backup <filename> [**-skipBackup**]

   Example: To restore by using a full backup of an appliance.
   > restore system backup backup_full_<nsip_address>_<date-timestamp>.tgz
3. Reboot the appliance.
   reboot

To restore the NetScaler from a local backup file by using the configuration utility

Navigate to System > Backup and Restore, right-click the backup file to be restored and click Restore.

To restore the NetScaler from a remote backup file by using the command line interface

Note: A remote backup file is any backup file that has been deleted from the NetScaler appliance.
Citrix recommends backing up the current configuration before restoring a previous configuration. However, if you do not want the restore command to automatically create a backup of the current configuration, use the –skipBackup parameter.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Copy the backup tar file from remote server to the local NetScaler appliance's /var/ns_sys_backup directory.
2. Add the backup metadata to the appliance's memory:

add system backup <backupFileName>

3. Restore the backed up configuration:

restore system backup <fileName> [–**skipBackup**]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Restarting or Shutting down the Appliance

Sep 16, 2014

The NetScaler appliance can be remotely restarted or shut down from the available user interfaces. When a standalone NetScaler appliance is restarted or shut down, the unsaved configurations (configurations performed since the last save ns config command was issued) are lost.

In a high availability setup, when the primary appliance is rebooted/shut down, the secondary appliance takes over and becomes the primary. The unsaved configurations from the old primary are available on the new primary appliance.

You can also restart the appliance by only rebooting the NetScaler software and not rebooting the underlying operating system. This is called a warm reboot. For example, when you add a new license or change the NetScaler IP address, you can warm reboot the NetScaler appliance for these changes to take place.

Note: Warm reboot can be performed only on nCore appliances.

To restart the NetScaler by using the command line interface

At the command prompt, type:

reboot [-warm]

To restart the NetScaler by using the configuration utility

1. In the configuration utility, click Reboot on the home page of the Configuration tab.
2. When prompted to reboot, select Save configuration to make sure that you do not lose any configurations.
   Note: You can perform a warm reboot by selecting Warm reboot.

To shut down the NetScaler by using the command line interface

At the command prompt, type:

- shutdown –p now: Shuts down the software and switches off the NetScaler. To restart NetScaler MPX, press the AC power switch. To Restart NetScaler VPX , restart the VPX instance.
- shutdown –h now: Shuts down the software and leaves the NetScaler switched on. Press any key to restart the NetScaler. This command does not switch off the NetScaler. Therefore, do not switch off the AC power or remove the AC power cables.

Note: The appliance cannot be shut down from the configuration utility.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Generating the NetScaler Technical Support Bundle

Oct 12, 2015

For help with analyzing and resolving any issues with a NetScaler appliance, you can generate a technical support bundle on the appliance and send the bundle to Citrix Technical Support. The NetScaler technical support bundle is a gzipped tar archive of system configuration data and statistics. It collects the following data from the NetScaler appliance on which you generate the bundle:

- **Configuration files**. All files in the /flash/nsconfig directory.
- **Newnslog files**. The currently running newnslog and some previous files. To minimize the archive file size, newnslog collection is restricted to 500 MB, 6 files, or 7 days, whichever occurs first. If older data is needed, it might require manual collection.
- **Log files**. Files in /var/log/messages*, /var/log/ns.log*, and other files under /var/log and /var/nslog.
- **Application core files**. Files created in the /var/core directory within the last week, if any.
- **Output of some CLI show commands**.
- **Output of some CLI stat commands**.
- **Output of BSD shell commands**.

You can use a single command to generate the technical support bundle and securely upload it to the Citrix Technical Support server. To upload, you must specify your My Citrix credentials. When you generate the bundle, you can specify the case or service request number that was allotted to you by Citrix Technical Support. If you have already generated a technical support bundle, you can upload the existing archive file to the Citrix Technical Support server by specifying the file name with the full path.

The technical support bundle is saved on the NetScaler appliance in an archive at the following location:

/var/tmp/support/support.tgz

The above path is a symlink to the most recent collector for easy access. The full filename varies, depending on the deployment topology, but generally follows a format similar to:

collector_<P/S>_<NS IP>_<DateTime>.tgz.

If your NetScaler appliance does not have direct Internet connectivity, you can use a proxy server to directly upload the technical support bundle to the Citrix Technical Support server. The basic format of the proxy string is:

proxy_IP:<proxy_port>

If the proxy server requires authentication, the format is:

username:password@proxsy_IP:<proxy_port>

> ## Note
>
> For NetScaler appliances in a high availability pair, you must generate the technical support bundle on each of the two nodes.
>
> For NetScaler appliances in a Cluster setup, you can generate the technical support bundle on each node individually, or you can generate smaller abbreviated archives for all nodes by using the cluster IP address.
>
> For NetScaler Admin Partitions, you must generate the technical support bundle from the default admin partition. To get the technical

© 1999-2017 Citrix Systems, Inc. All rights reserved.

support bundle for a specific partition, you must specify the name of the partition for which you want to generate the technical support bundle. If you do not specify the name of the partition, data is collected from all admin partitions.

To generate the NetScaler technical support bundle by using the command line interface

At the command prompt, type:

show techsupport [-**scope** <scope> <partitionName>] [-**upload** [-**proxy** <string>] [-**casenumber** <string>] [-**file** <string>] [-**description** <string>] [-**userName** <string> -**password**]]

Examples

Sr. No	Task	Command
1.	Generate and upload the technical support bundle to the Citrix Technical Support server.	show techsupport –upload –userName account1 –password xxxxxxx
2.	Generate and upload the technical support bundle to the Citrix Technical Support server through a proxy server.	show techsupport –upload –proxy 1.1.1.1:80 –userName account1 –password xxxxxxx
3.	Upload an existing technical support bundle to the Citrix Technical Support server.	show techsupport –upload -file /var/tmp/support/collector_P_10.102.29.160_9Sep2015_15_22.tar.gz –userName account1 –password xxxxxxx
4.	Generate small, abbreviated archives for all nodes in a Cluster setup. Execute this command by using the cluster IP address.	show techsupport –scope CLUSTER
5.	Generate a technical support bundle specific to an admin partition. Execute this command on the default admin partition.	show techsupport –scope PARTITION partition1

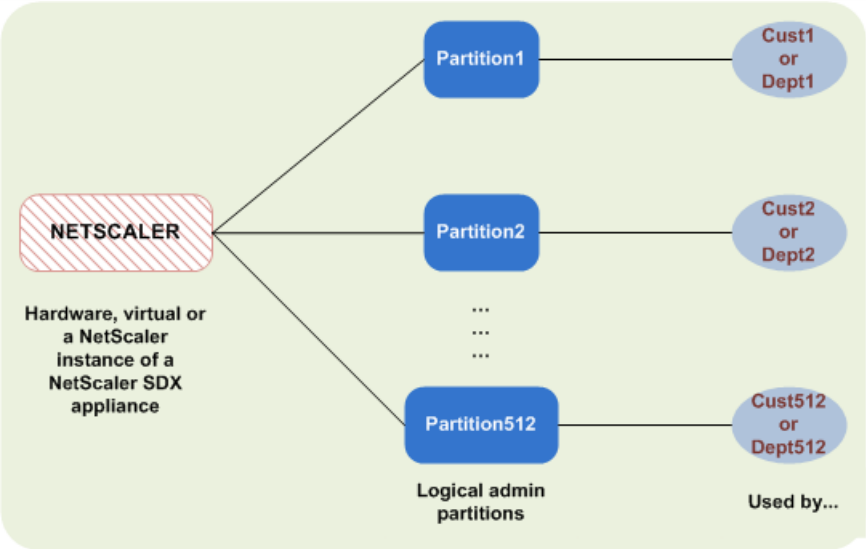
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Admin Partitioning

Nov 16, 2015

> **Important**
>
> - The admin partitioning feature is provided from NetScaler 11.0 onwards. It is also provided as a separate enhancement release that is based on NetScaler 10.5 Build 52.11.
>
> - Until specified otherwise, this documentation applies to NetScaler versions 11.0 and 10.5 Build 52.11.

A NetScaler appliance can be partitioned into logical entities called admin partitions, where each partition can be configured and used as a separate NetScaler appliance. The following figure shows the partitions of a NetScaler being used by different customers and departments:



For information on how admin partitions can benefit your business, see Benefits and Uses of Admin Partitions.

A partitioned NetScaler appliance has a single default partition and one or more admin partitions. The following table provides further details on the two partition types:

	Default Partition	Admin Partitions
Availability	The NetScaler ships with a single partition, which is called a default partition. The default partition is retained even after the NetScaler is partitioned.	Must be explicitly created as described in Partitioning a NetScaler.
Number of Partitions	One	A NetScaler appliance can have one or more (maximum of 512) admin partitions.
User Access and	The default partition can be accessed and configured by all NetScaler users who are not associated with a *partition-specific* command policy. As always, the operations that a user	Can be created only by NetScaler superusers who also specify the users for that partition. Only superusers and associated users of the partition can access and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

		Note: Partition users do not have shell access.
	Default Partition	**Admin Partitions**
File Structure	All files in a default partition are stored in the default NetScaler file structure. For example, the NetScaler configuration file is stored in the */nsconfig* directory and NetScaler logs are stored in the */var/log/* directory.	All files in an admin partition are stored in directory paths that have the name of the admin partition. For example, the NetScaler configuration file (ns.conf) is stored in the */nsconfig/partitions/<partitionName>* directory. Other partition-specific files are stored in the */var/partitions/<partitionName>* directories. Some other paths in an admin partition: • Downloaded files: */var/partitions/<partitionName>/download/* • Log files: */var/partitions/<partitionName>/log/* **Note:** Currently, logging is not supported at partition-level. Therefore, this directory is empty and all logs are stored in the */var/log/* directory. • SSL CRL certificate related files: */var/partitions/<partitionName>/netscaler/ssl*
Resources Available	All NetScaler resources.	NetScaler resources that are explicitly assigned to the admin partition.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Benefits and Uses of Admin Partitions

Dec 09, 2015

You can avail the following benefits by using admin partitions for your deployment:

- Allows delegation of administrative ownership of an application to the customer.
- Reduces the cost of ADC ownership without compromising on performance and ease-of-use.
- Safeguards from unwarranted configuration changes. In a non-partitioned NetScaler, authorized users of other application could intentionally or unintentionally change configurations that are required for your application. This could lead to undesirable behavior. This possibility is reduced in a partitioned NetScaler.
- Isolates traffic between different applications by the use of dedicated VLANs for each partition.
- Accelerates and allows to scale application deployments.
- Allows application-level or localized management and reporting.

Let us analyze a couple of cases to understand the scenarios in which you can use admin partitions.

Case 1 : Admin partitions used by an enterprise

Case 2 : Admin partitions used by a service provider

© 1999-2017 Citrix Systems, Inc. All rights reserved.

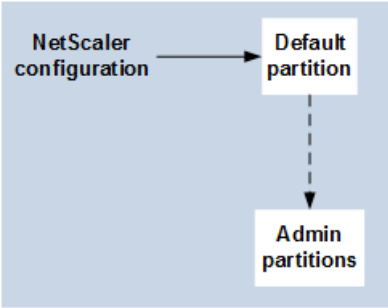
NetScaler Configurations Supported in Partitions

Feb 04, 2018

Depending on the NetScaler configuration and the partition in which the configuration is performed, NetScaler configurations can be categorized into the cases mentioned below.

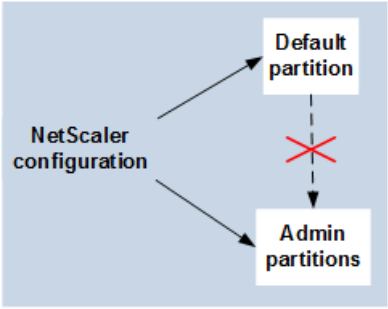
> ## Note
>
> - Admin partitions cannot be set up on a NetScaler cluster. This means that a NetScaler cluster cannot be partitioned.
> - Admin partitions cannot be set up on a NetScaler MPX-FIPS appliance.
> - Case 3 lists the NetScaler features that are not supported in admin partitions.

Case 1 (global configurations). Configurations that can be performed ONLY in the default partition and which are available or impact all the admin partitions.



- Updates to built-in entities for monitors, TCP profiles, HTTP profiles, and so on.
- Updates to global parameters for syslog, nslog, weblog, content switching, IPSEC, SIP, DHCP, Surge protection, TCP buffering, and system collection.
- High availability (HA) configurations
- Interface and VLAN changes
- User configurations

Case 2 (partition-specific configurations). Configurations that can be performed independently in default and admin partitions. These configurations are applicable only to the partition in which they are performed.

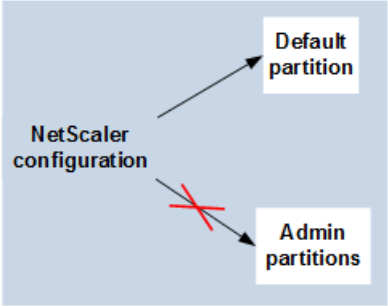


- Getting traffic level statistics for a partition.
- Partition admin can update IP bindings for VLAN which is bound to that partition. But cannot update the interface bindings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Clearing NetScaler configurations.
- Feature-specific parameters for the following features: AppFlow, AppQoE, HTTP compression, DNS, TCP, HTTP, encryption, responder, rewrite, and SSL.
- Feature-specific configurations such as virtual servers, services, monitors, and so on.

Case 3: Configurations that cannot be performed on admin partitions. These features can be configured in the default partition, but there is no impact on admin partitions.



Note: Configurations that are supported on admin partitions for a particular release are marked as **Yes**.

Group	Feature	NetSCALER 10.5	NetScaler 11.0
Policy	Extensibility	No	Yes
Load Balancing	DBS AutoScale	No	Yes
Load Balancing	DNSSEC	No	No
Load Balancing	Diameter	No	No
Load Balancing	RTSP	No	No
Load Balancing	Sure Connect	No	Yes
Load Balancing	Autoscale Service Group	No	No
Manageability	RBA External Authentication	No	No
Manageability	RISE Cisco	No	No
Manageability	ACI-Cisco	No	No
Manageability	AppExpert	Yes	Yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Group	Feature	NetSCALER 10.5	NetScaler 11.0
Manageability	HDX Insight	No	No
ManageabilityNo	Insight	No	No
VPN	Cloudbridge Connector	No	No
VPN	NetScaler Gateway or SSL VPN	No	No
VPN	SSL VPN ICA Proxy	No	No
VPN	Web Interface on NetScaler	No	No
SSL	SSL Profile	No	No
SSL	SSL-FIPS	No	No
SSL	External-HSM	No	No
Infra	Cache Re-direction	No	No
Infra	Integrated Caching (Restricted Feature)	No	Yes
Network	VXLAN	No	No
Network	Graceful Shutdown	No	Yes
Network	LSN	No	No
Network	IP v6 Ready Logo	No	No
Network	Vpath	No	No
Load Balancing	Datastream	No	Yes
Logging	Web logging	No	Yes
Network	L2 Param/L3 Param	No	Yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Group	Feature	NetSCALER 10.5	NetScaler 11.0
Network	GRE Tunnel	No	No
Load Balancing	Scriptable Monitoring	No	Yes
Load Balancing	GSLB	No	No
Infra	Connection Mirroring	No	No
Infra	FEO	No	No
Infra	Nstrace	No	No
Load Balancing	SureConnect	No	Yes
Load Balancing	Priority Queuing	No	No
Network	HDOSP	No	No
Network	Netprofile (Restricted Feature)	No	No
Network	Networking (Restricted Feature)	No	No
Network	VRRP (Restricted Feature)	No	No
Logging	Audit Logging (Restricted Feature)	No	No
VPN	NetScaler Gateway	No	No
VPN	AAA-TM	No	Yes except RBA authentication feature.
APPFW	Application Firewall	No	No
Load Balancing	TCP Buffering (Restricted Feature)	No	No
Policies	OCSP Responder	No	Yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Partitioning a NetScaler

Mar 07, 2017

> **Important**
>
> - Only superusers are authorized to create and configure admin partitions.
> - Unless specified otherwise, configurations to set up an admin partition must be done from the default partition.

By partitioning a NetScaler appliance, you are in-effect creating multiple instances of a single NetScaler appliance. Each instance has its own configurations and the traffic of each of these partitions is isolated from the other by assigning each partition a dedicated VLAN.

A partitioned NetScaler has one default partition and the admin partitions that are created. To set up an admin partition, you must first create a partition with the relevant resources (memory, maximum bandwidth, and connections). Then, specify the users that can access the partition and the level of authorization for each of the users on the partition.

Finally, assign a VLAN to the partition so that the traffic for that partition is isolated and segregated.

> **Tip**
>
> Citrix recommends that you bind only tagged VLANs to partitions. This ensures that control traffic packets (for example, LACP, LLDP, and xSTP packets) are handled in the default partition.

Rate limits for an admin partition are as follows:

- **Maximum memory limit.** Must be configured as the memory that will be required for each admin partition. You must make sure that you set the appropriate value when creating the partition.

  Once an admin partition is created, the memory limit cannot be decreased. The memory limit can however be increased when required or more specifically, when there is execution failure due to insufficient memory in a partition; provided sufficient memory is available in the default partition.

  Note: From NetScaler 11.0 Build 64.x onwards, you can set the memory limit to a minimal value of 5 MB, when creating the admin partition. This setting can be useful for lighter deployments of the NetScaler appliance.

- **Maximum bandwidth.** The maximum bandwidth that can be used by an admin partition. This value must be limited to the appliance's licensed throughput. Otherwise, in effect, you are NOT limiting the bandwidth that can be used by the admin partition.

  It must be configured such that it accounts for the bandwidth that the application requires. If the application bandwidth exceeds the configured value, packets will be dropped. It accounts for incoming and outgoing packets.

  The maximum bandwidth can be increased or decreased when required.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note:

- The default value is 10240 kbps, minimum value is 0, and maximum value is 4294967295 kbps.
- Setting this parameter to its minimum value (0) means that you are not assigning any bandwidth to the partition. Traffic received for this partition will be dropped.
- This is not the guaranteed bandwidth available for the admin partition. After a partition is configured with a maximum bandwidth value, the actual bandwidth assigned depends on the appliance's licensed throughput.

- **Maximum number of connections.** Must be configured such that it accounts for the maximum simultaneous flows expected within a partition. It accounts for client-side and server-side TCP connections. New connections cannot be established beyond this configured value.

  The maximum number of connections can be increased or decreased when required.

Note: When the bandwidth and number of connections crosses the threshold value, if SNMP is configured, traps will be sent with the relevant information.

> ### Note
>
> - After creating a partition, inform the users that the NetScaler configurations they perform will be isolated from users who are not members of the partition.
> - Make sure the relevant users, command policies, VLANs, and bridgegroups are available on the NetScaler appliance.
> - For deployments that have large size of NetScaler configuration and large quantum of traffic, Citrix advises that you increase the default values for the maximum memory limit, maximum bandwidth, and maximum number of connections.

Configuring Rate Limit Alarm

In an administrative partition, you can configure a SNMP-RATE-LIMIT alarm to generate the following SNMP traps for rate limiting a partition memory or bandwidth when its limit exceeds configured threshold or returns to normal.

- PartitionCONNLimitExceeded
- PartitionCONNLimitNormal
- PartitionBWLimitExceeded

Note: To enable generation of the SNMP trap, you must enable the SNMP-RATE-LIMIT alarm in the partition and then configure the destination device to which the appliance can send the trap messages.

For more information about SNMP alarms and traps, see https://docs.citrix.com/en-us/netscaler/11/system/snmp/generating-snmp-traps-on-netscaler.html

To partition a NetScaler by using the command line interface

On the command prompt, do the following:

1. Create a partition and configure the NetScaler resources for that partition.
   add ns partition <partitionName> [-maxBandwidth <positive_integer>] [-maxConn <positive_integer>] [-maxMemLimit <positive_integer>]

   Note: Check the rate limiting content provided above for tips to update the maximum memory limit, maximum bandwidth, and maximum number of connections.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Associate the appropriate users with the partition.
   bind system user <name> -partitionName <string>

3. Specify the level of authorization for each user by associating one of the following command policies: *partition-operator, partition-read-only, partition-network, and partition-admin*.
   bind system user <name> <policyName> <priority>

4. Configure the VLAN through which traffic for this partition must be routed. You can use bridgegroups instead of VLANs to route the traffic.

   - Add the VLAN and bind the required interfaces to it.
     add vlan <id>

     bind vlan <id> -ifnum <interface>

     Note: When a VLAN is bound to an admin partition, its IP address bindings are lost. To make sure that the VLAN continues to have the IP address, create the IP address on the admin partition and then bind it to that VLAN.

   OR

   - Add the bridgegroup and bind the required VLANs to it.
     add bridgegroup <id>

     bind bridgegroup <id> -vlan <id>

5. Bind the VLAN or bridgegroup to the partition.
   bind ns partition <partitionName> -vlan <positive_ integer>

   OR

   bind ns partition <partitionName> -bridgegroup <positive_ integer>

   Note: Use the show vlan or the show bridgegroup command to view the partitions associated with that VLAN or bridgegroup.
6. Verify the configurations of the partition.
   show ns partition <partitionName>

   Note: You can also use the stat ns partition command to view partition configurations.
7. Save the configuration.
   save ns config

To partition a NetScaler by using the configuration utility

On the Configuration tab of the graphical user interface:

1. Navigate to System > Partition Administration, click Add and do the following:
   1. Create and configure the resources for the admin partition.
   2. Specify the VLANs or bridgegroups to be associated with the partition.
   3. Associate user(s) with the partition.
      Note: Make sure you bind users who are not yet associated with partition type command policies.
2. Navigate to System > User Administration, and to the partition user, bind the appropriate command policy. The command

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

policy must be one of the partition- entries. The choice depends on the level of authorization you intend the user to have.

3. Save the configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring in a NetScaler Partition

Mar 30, 2016

Accessing a partitioned NetScaler is the same as accessing a non-partitioned NetScaler: through the NetScaler IP (NSIP) address or any other management IP address. As a user, after you provide your valid logon credentials, you are taken to the partition to which you are bound. Any configurations that you create are saved to that partition. If you are associated with more than one partition, you are taken to the first partition with which you were associated. If you want to configure entities on one of your other partitions, you must explicitly switch to that partition.

After accessing the appropriate partition, configurations that you perform are saved to that partition and are specific to that partition.

Note

- NetScaler superusers and other non-partition users are taken to the default partition.
- Users of all the 512 partitions can log in simultaneously.

Tip

To access a partitioned NetScaler appliance over HTTPS by using the SNIP (with management access enabled), make sure that each partition has the certificate of its partition administrator. Within the partition, the partition admin must do the following:

1. Add the certificate to the NetScaler.
   > **add ssl certKey** ns-server-certificate **-cert** ns-server.cert **-key** ns-server.key

2. Bind it to a service named "nskrpcs-<SNIP>-3009", where <SNIP> must be replaced with the SNIP address, in this case 100.10.10.1.
   > **bind ssl service** nskrpcs-100.10.10.1-3009 **-certkeyName** ns-server-certificate

To configure in a NetScaler partition by using the command line interface

1. Log on to the NetScaler appliance.

2. Check if you are in the correct partition. The command prompt displays the name of the currently selected partition.

   - If yes, skip to the next step.

   - If no, get a list of the partitions with which you are associated and switch over to the appropriate partition.

     - **show system user** <username>
     - **switch ns partition** <partitionName>

3. Now, you can perform the required configurations just as a non-partitioned NetScaler.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure in a NetScaler partition by using the configuration utility

1. Log on to the NetScaler appliance.

2. Check if you are in the correct partition. The top bar of the graphical user interface displays the name of the currently selected partition.

   - If yes, skip to the next step.

   - If no, navigate to **Configuration > System > Administrative Partitions > Partitions**, right-click the partition to which you want to switch, and select **Switch**.

3. Now, you can perform the required configurations just as a non-partitioned NetScaler.

Configuring an LACP Ethernet Channel on the Default Admin Partition

With Link Aggregation Control Protocol (LACP), you can combine multiple ports into a single, high-speed link (also called a channel). An LACP-enabled appliance exchanges LACP Data Units (LACPDU) over the channel.

There are three LACP configuration modes that you can enable in the default partition of a NetScaler appliance:

1. Active. A port in active mode sends LACPDUs. Link aggregation is formed if the other end of the Ethernet link is in the LACP active or passive mode.
2. Passive. A port in passive mode sends LACPDUs only when it receives LACPDUs. The link aggregation is formed if the other end of the Ethernet link is in the LACP active mode.
3. Disable. Link aggregation is not formed.

Note: By default, the link aggregation is disabled in the default partition of the appliance.

LACP exchanges LACPDU between devices connected by an Ethernet link. These devices are typically referred as an actor or partner.

A LACPDU data unit contains the following parameters:

- LACP Mode. Active, passive or disable.
- LACP timeout. The waiting period before timing out the partner or actor. Possible values: Long and Short. Default: Long.
- Port Key. To distinguish between the different channel. When key is 1, LA/1 is created. When key is 2, LA/2 is created. Possible values: Integer from 1 through 8. 4 through 8 is for cluster CLAG.
- Port Priority. Minimum value: 1. Maximum value: 65535. Default: 32768.
- System Priority. Uses this priority along with system MAC to form the system ID to uniquely identify the system during LACP negotiation with the partner. Sets system priority from 1 and 65535. The default value is set to 32768.
- Interface. Supports 8 interfaces per channel on NetScaler 10.1 appliance and supports 16 interfaces per channel on NetScaler 10.5 and 11.0 appliances.

After exchanging LACPDUs, the actor and partner negotiate the settings and decide whether to add the ports to the aggregation.

Configuring and Verifying LACP on a NetScaler appliance by using the command line interface

To configure and verify LACP on a NetScaler appliance by using the command line

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Enable LACP on each interface.

At the command prompt, type:

set interface <Interface_ID> -lacpMode PASSIVE -lacpKey 1

When you enable LACP on an interface, the channels are dynamically created. Additionally, when you enable LACP on an interface and set lacpKey to 1, the interface is automatically bound to channel LA/1.

Note: When you bind an interface to a channel, the channel parameters take precedence over the interface parameters, so the interface parameters are ignored. If a channel was created dynamically by LACP, you cannot perform add, bind, unbind, or remove operations on the channel. A channel dynamically created by LACP is automatically deleted when you disable LACP on all interfaces of the channel.

2. Set the system priority.

At the command prompt, type:

set lacp -sysPriority <Positive_Integer>

3. Verify that LACP is working as expected.

show interface <Interface_ID>

show channel

show LACP

Note: In some versions of Cisco IOS, running the switchport trunk native vlan <VLAN_ID> command causes the Cisco switch to tag LACP PDUs. This causes the LACP channel between the Cisco switch and the NetScaler appliance to fail. However, this issue does not affect the static link aggregation channels configured in the above procedure.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SNMP Support for Admin Partitions

Jan 31, 2018

A partitioned NetScaler appliance uses SNMP infrastructure for partition rate limiting and for monitoring partition resource utilization details.

SNMP Traps for Admin Partition Rate Limiting

On a partitioned NetScaler appliance, a PARTITION-RATE-LIMIT alarm can generate three SNMP traps for notifying a partition resource (such as bandwidth, connection, or memory) has reached its limit or returned to normal.

Following are the three SNMP traps generated.

- partitionCONNLimitExceeded
- partitionCONNLimitNormal
- partitionBWLimitExceeded

Configuring PARTITION-RATE-LIMIT Alarm

To configure PARTITION-RATE-LIMIT alarm in a specific partition and enable generation of the SNMP trap messages.

1. Enable PARTITION-RATE-LIMIT Alarm
2. Configure PARTITION-RATE-Limit Alarm
3. Configure SNMP Trap Destination

To enable PARTITION-RATE-LIMIT alarm by using the NetScaler command line

At the command prompt, type the following commands:

enable snmp alarm PARTITION-RATE-LIMIT

show snmp alarm PARTITION-RATE-LIMIT

To configure PARTITION-RATE-LIMIT Alarm by using the NetScaler command line

At the command prompt, type the following command:

set snmp alarm PARTITION-RATE-LIMIT [**-state** (ENABLED | DISABLED)] [**-severity** <severity>] [**-logging** (ENABLED | DISABLED)]

To configure SNMP Trap Destination by using the NetScaler command line

At the command prompt, type the following command:

add snmp trap <trapClass> <trapDestination> [-**version** <version>] [-**td** <positive_integer>] [-**destPort** <port>] [-**communityName** <string>] [-**srcIP** <ip_addr|ipv6_addr>] [-**severity** <severity>] [-**allPartitions** (ENABLED | DISABLED)]

To configure PARTITION-Rate-Limit Alarm by using the NetScaler GUI

Navigate to **System** > **SNMP** > **Alarms**, select **PARTITION-RATE-LIMIT** alarm and configure the alarm parameters.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure SNMP trap destination by using the NetScaler GUI

Navigate to **System** > **SNMP** > **Trap**, specify the IP address of the destination device.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case: Reusing the Same Identifier in Different Partitions

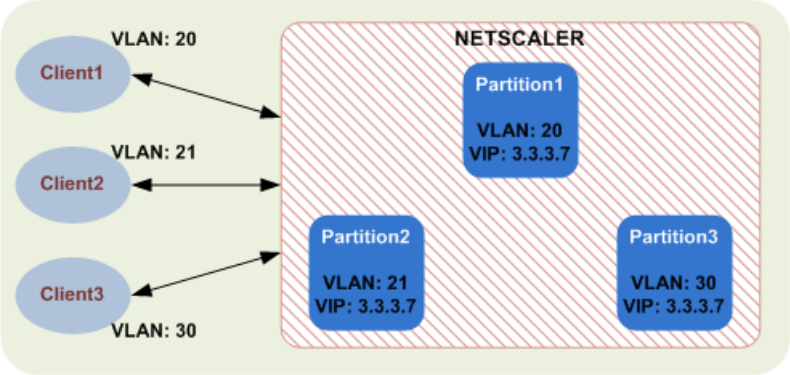
Jul 14, 2015

By using admin partitions, resource identifiers such as IP addresses and entity names can be reused in different partitions. This means that:

- You can use an IP address (for example, 3.3.3.7) as a virtual server IP address in different partitions.
- You can use the same name (for example, lbvserver1) for a virtual server in different partitions.

This is possible as each partition is associated with a different VLAN (or bridgegroup) and therefore traffic destined for different applications is segregated.

As shown in the following image, the virtual server IP address 3.3.3.7 is used in Partition1, Partition2, and Partition3.



Let us understand how the configurations must be performed.

Creating and configuring partitions that share the same IP address among virtual servers

1. **On default partition:** Log on to the NetScaler appliance as a super user and configure the three partitions as follows:

```
shell> ssh 10.102.29.60 -l nsroot
password: ******

> add ns partition Partition1
Done

> add ns partition Partition2
Done

> add ns partition Partition3
Done

> bind system user user1 -partitionName Partition1
Done

> bind system user user2 -partitionName Partition2
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> bind system user user3 -partitionName Partition3
Done

> bind system user user1 partition-admin 10
Done

> bind system user user2 partition-admin 20
Done

> bind system user user3 partition-admin 20
Done

> add vlan 20
Done

> bind vlan 20 -ifnum 2/1
Done

> add vlan 21
Done

> bind vlan 21 -ifnum 3/1
Done

> add vlan 30
Done

> bind vlan 30 -ifnum 4/1
Done

> bind ns partition Partition1 -vlan 20
Done

> bind ns partition Partition2 -vlan 21
Done

> bind ns partition Partition3 -vlan 30
Done
```

2. **On Partition1:** Log on to the NetScaler appliance as user1 and configure on Partition1.

```
shell> ssh 10.102.29.60 -l user1
password: *****

Partition1> add ns ip 3.3.3.2 255.255.255.0 -vServer DISABLED -type SNIP
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Partition1> add service s1 3.3.3.5 HTTP 80
Done

Partition1> add lb vserver lbvserver1 HTTP 3.3.3.7 80 -persistenceType NONE
Done

Partition1> bind lb vserver lbvserver1 s1
Done

Partition1> bind vlan 20 -IPAddress 3.3.3.2 255.255.255.0
Done
```

3. **On Partition2:** Log on to the NetScaler appliance as user2 and configure on Partition2.

```
shell> ssh 10.102.29.60 -l user2
password: *****

Partition2> add ns ip 5.5.5.3 255.255.255.0 -vServer DISABLED -type SNIP
Done

Partition2> add service s1 5.5.5.5 HTTP 80
Done

Partition2> add lb vserver lbvserver1 HTTP 3.3.3.7 80 -persistenceType NONE
Done

Partition2> bind lb vserver lbvserver1 s1
Done

Partition2> bind vlan 21 -IPAddress 5.5.5.3 255.255.255.0
Done
```

4. **On Partition3:** Log on to the NetScaler appliance as user3 and configure on Partition3.

```
shell> ssh 10.102.29.60 -l user3
password: *****

Partition3> add ns ip 6.6.6.3 255.255.255.0 -vServer DISABLED -type SNIP
Done

Partition3> add service s1 6.6.6.6 HTTP 80
Done

Partition3> add lb vserver lbvserver1 HTTP 3.3.3.7 80 -persistenceType NONE
Done

Partition3> bind lb vserver lbvserver1 s1
Done

Partition3> bind vlan 30 -IPAddress 6.6.6.3 255.255.255.0
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

FAQs

Feb 02, 2016

- Where can I get the NetScaler configuration file for a partition?
- How can I configure integrated caching in a partitioned NetScaler appliance?
- What is the scope for L2 and L3 parameters in admin partitions?
- How to enable dynamic routing in an admin partition?
- Where can I find the logs for a partition?
- How can I get auditlogs for an admin partition?
- How can I get web logs for an admin partition?
- How can I get the trace for an admin partition?
- How can I get the technical support bundle specific to an admin partition?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Authentication and Authorization

Mar 17, 2014

To configure NetScaler authentication and authorization, you must first define the users who have access to the NetScaler appliance, and then you can organize these users into groups. After configuring users and groups, you need to configure command policies to define types of access, and assign the policies to users and/or groups.

You must log on as an administrator to configure users, groups, and command policies. The default NetScaler administrator user name is *nsroot*. After logging on as the default administrator, you should change the password for the nsroot account. Once you have changed the password, no user can access the NetScaler appliance until you create an account for that user. If you forget the administrator password after changing it from the default, you can reset it to nsroot.

Note: Local users can authenticate to the NetScaler even if external authentication servers are configured. You can restrict this by disabling the localAuth parameter of the set system parameter command.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Users, User Groups, and Command Policies

Nov 02, 2017

You must define your users by configuring accounts for them. To simplify the management of user accounts, you can organize them into groups. You can create command policies, or use built-in command policies, to regulate user access to commands.

You can also customize the command-line prompt for a user. Prompts can be defined in a user's configuration, in a user-group configuration, and in the global system configuration settings. The prompt displayed for a given user is determined by the following order of precedence:

1. Display the prompt as defined in the user's configuration.
2. Display the prompt as defined in the group configuration for the user's group.
3. Display the prompt as defined in the system global configuration settings.

You can now specify a timeout value for inactive CLI sessions for a system user. If a user's CLI session is idle for a time that exceeds the timeout value, the NetScaler appliance terminates the connection. The timeout can be defined in a user's configuration, in a user-group configuration, and in the global system configuration settings. The timeout for inactive CLI sessions for a user is determined by the following order of precedence:

1. Timeout value as defined in the user's configuration.
2. Timeout value as defined in the group configuration for the user's group.
3. Timeout value as defined in the global system configuration settings.

A NetScaler root administrator can configure the maximum concurrent session limit for system users. By restricting the limit, you can reduce the number of open connections and improve server performance. As long as the CLI count is within the configured limit, concurrent users can log on the configuration utility any number of times. However, if the number of CLI sessions reaches the configured limit, users can no longer log on to the configuration utility. For example, if the number of concurrent session is configured to 20, concurrent users can log on to 19 CLI sessions. But if the user is logged on to the 20th CLI session, any attempt to log on to the configuration utility, CLI or NITRO results in an error message ((ERROR: Connection limit to CFE exceeded).

Note: The default the number of concurrent sessions is configured to 20 and the maximum number of concurrent sessions is configured to 40.

This document includes the following details:

- Configuring User Accounts
- Configuring User Groups
- Configuring Command Policies

Configuring User Accounts

Updated: 2014-08-07

To configure user accounts, you simply specify usernames and passwords. You can change passwords and remove user accounts at any time.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To create a user account by using the command line interface

At the command prompt, type the following commands to create a user account and verify the configuration:

- add system user <username> [-externalAuth (ENABLED | DISABLED)] [-promptString <string>] [-timeout <secs>] [-logging (ENABLED | DISABLED)] [-maxsession <positive_integer>]
- show system user <userName>

Where Logging option is for external users to collect log data externally using weblogging or audit logging mechanism. If enabled, the auditing client authenticates itself with NetScaler to collect logs through this user account.

Example

> add system user johnd -promptString user-%u-at-%T

Enter password:

Confirm password:

> show system user johnd

user name: john

Timeout:900 Timeout Inherited From: Global

External Authentication: ENABLED

Logging: DISABLED

Maximum Client Sessions: 20

To configure a user account by using the configuration utility

Navigate to **System** > **User Administration** > **Users**, and create the user.

Configuring User Groups

Updated: 2014-08-07

After configuring a user group, you can easily grant the same access rights to everyone in the group. To configure a group, you create the group and bind users to the group. You can bind each user account to more than one group. Binding user accounts to multiple groups might allow more flexibility when applying command policies.

To create a user group by using the command line interface

At the command prompt, type the following commands to create a user group and verify the configuration:

- add system group <groupName> [-promptString <string>] [-timeout <secs>]
- show system group <groupName>

Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

> add system group Managers -promptString Group-Managers-at-%h

To bind a user to a group by using the command line interface

At the command prompt, type the following commands to bind a user account to a group and verify the configuration:

- bind system group <groupName> -userName <userName>
- show system group <groupName>

Example

> bind system group Managers -userName user1

To configure a user group by using the configuration utility

Navigate to System > User Administration > Groups, and create the user group.

Note: To add members to the group, in the Members section, click Add. Select users from the Available list and add them to the Configured list.

Configuring Command Policies

Command policies regulate which commands, command groups, virtual servers, and other entities that users and user groups are permitted to use.

The appliance provides a set of built-in command policies, and you can configure custom policies. To apply the policies, you bind them to users and/or groups.

Here are the key points to keep in mind when defining and applying command policies.

- You cannot create global command policies. Command policies must be bound directly to the users and groups on the appliance.
- Users or groups with no associated command policies are subject to the default (DENY-ALL) command policy, and are therefore unable to execute any configuration commands until the proper command policies are bound to their accounts.
- All users inherit the policies of the groups to which they belong.
- You must assign a priority to a command policy when you bind it to a user account or group account. This enables the appliance to determine which policy has priority when two or more conflicting policies apply to the same user or group.
- The following commands are available by default to any user and are unaffected by any command you specify:
  help, show cli attribute, set cli prompt, clear cli prompt, show cli prompt, alias, unalias, history, quit, exit,whoami, config, set cli mode, unset cli mode, and show cli mode.

Built-in Command Policies

Updated: 2015-06-15

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following table describes the built-in policies.

Table 1. Built-in Command Policies

Policy name	Allows
read-only	Read-only access to all show commands except show ns runningConfig, show ns ns.conf, and the show commands for the NetScaler command group.
operator	Read-only access and access to commands to enable and disable services and servers.
network	Full access, except to the set and unset SSL commands, show ns ns.conf, show ns runningConfig, and show gslb runningConfig commands.
sysadmin	[Included in NetScaler 11.0 and later] A sysadmin is lower than a superuser is terms of access allowed on the appliance. A sysadmin user can perform all NetScaler operations with the following exceptions: no access to the NetScaler shell, cannot perform user configurations, cannot perform partition configurations, and some other configurations as stated in the sysadmin command policy.
superuser	Full access. Same privileges as the nsroot user.

Creating Custom Command Policies

Updated: 2015-06-15

Regular expression support is offered for users with the resources to maintain more customized expressions, and for those deployments that require the flexibility that regular expressions offer. For most users, the built-in command policies are sufficient. Users who need additional levels of control but are unfamiliar with regular expressions might want to use only simple expressions, such as those in the examples provided in this section, to maintain policy readability.

When you use a regular expression to create a command policy, keep the following in mind.

- When you use regular expressions to define commands that will be affected by a command policy, you must enclose the commands in double quotation marks. For example, to create a command policy that includes all commands that begin with show, type the following:
  "^show .*$"

  To create a command policy that includes all commands that begin with rm, type the following:

  "^rm .*$"

- Regular expressions used in command policies are not case sensitive.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The following table lists examples of regular expressions:

Table 2. Examples of Regular Expressions for Command Policies

Command specification	Matches these commands
"^rm\s+.*$"	All remove actions, because all remove actions begin with the rm string, followed by a space and additional parameters such as command groups, command object types and arguments.
"^show\s+.*$"	All show commands, because all show actions begin with the show string, followed by a space and additional parameters such as command groups, command object types and arguments.
"^shell$"	The shell command alone, but not combined with any other additional parameters such as command groups, command object types and arguments.
"^add\s+vserver\s+.*$"	All create virtual server actions, which consist of the add virtual server command followed by a space and additional parameters such as command groups, command object types and arguments.
"^add\s+ (lb\s+vserver)\s+.*"	All create lb virtual server actions, which consist of the add lb virtual server command followed by a space and additional parameters such as command groups, command object types and arguments.

The following table shows the command specifications for each of the built-in command policies.

Table 3. Expressions Used in the Built-in Command Policies

Policy name	Command specification regular expression
read-only	(^man.*)|(^show\s+(?!system)(?!configstatus)(?!ns ns\.conf)(?!ns savedconfig)(?!ns runningConfig)(?!gslb runningConfig)(?!audit messages)(?!techsupport).*)|(^stat.*)
operator	(^man.*)|(^show\s+(?!system)(?!configstatus)(?!ns ns\.conf)(?!ns savedconfig)(?!ns runningConfig)(?!gslb runningConfig)(?!audit messages)(?!techsupport).*)|(^stat.*)|(^(enable|disable) (server|service).*)
network	^(?!clear ns config.*)(?!scp.*)(?!set ssl fips)(?!reset ssl fips)(?!diff ns config)(?!shell)(?!reboot)(?!batch)\S+\s+(?!system)(?!configstatus)(?!ns ns\.conf)(?!ns savedconfig)(?!ns runningConfig)(?!gslb runningConfig)(?!techsupport).*
sysadmin	[From NetScaler 11 onwards] ^(?!shell)(?!sftp)(?!scp)(?!batch)(?!source)(?!.*superuser)(?!.*nsroot)(?!show\s+system\s+(user|cmdPolicy))(?!(set|add|rm|create|export|kill)\s+system)(?!(unbind|bind)\s+system\s+(user|group))(?!diff\s+ns\s+config)(?!\S+\s+ns\s+partition).*
superuser	.*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To create a command policy by using the command line interface

At the command prompt, type the following commands to create a command policy and verify the configuration:

- add system cmdPolicy <policyname> <action> <cmdspec>
- show system cmdPolicy <policyName>

Example

```
> add system cmdPolicy read_all ALLOW (^show\s+(!system)(!ns ns.conf)(!ns runningConfig).*)|(^stat.*)
```

To configure a command policy by using the configuration utility

Navigate to System > User Administration > Command Policies, and create the command policy.

Binding Command Policies to Users and Groups

Updated: 2014-08-07

Once you have defined your command policies, you must bind them to the appropriate user accounts and groups. When you bind a policy, you must assign it a priority so that the appliance can determine which command policy to follow when two or more applicable command policies are in conflict.

Command policies are evaluated in the following order:

- Command policies bound directly to users and the corresponding groups are evaluated according to priority number. A command policy with a lower priority number is evaluated before one with a higher priority number. Therefore, any privileges the lower-numbered command policy explicitly grants or denies are not overridden by a higher-numbered command policy.
- When two command policies, one bound to a user account and other to a group, have the same priority number, the command policy bound directly to the user account is evaluated first.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To bind command policies to a user by using the command line interface

At the command prompt, type the following commands to bind a command policy to a user and verify the configuration:

- bind system user <userName> -policyName <policyName> <priority>
- show system user <userName>

Example

> bind system user user1 -policyName read_all 1

To bind command policies to a user by using the configuration utility

Navigate to System > User Administration > Users, select the user and bind command policies.

Optionally, you can modify the default priority to ensure that the policy is evaluated in the proper order.

To bind command policies to a group by using the command line interface

At the command prompt, type the following commands to bind a command policy to a user group and verify the configuration:

- bind system group <groupName> -policyName <policyName> <priority>
- show system group <groupName>

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> bind system group Managers -policyName read_all 1
```

To bind command policies to a group by using the configuration utility

Navigate to System > User Administration > Groups, select the group and bind command policies.

Optionally, you can modify the default priority to ensure that the policy is evaluated in the proper order.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Resetting the Default Administrator (nsroot) Password

Nov 24, 2014

The nsroot account provides complete access to all features of the appliance. Therefore, to preserve security, the nsroot account should be used only when necessary, and only individuals whose duties require full access should know the password for the nsroot account. Frequently changing the nsroot password is advisable. If you lose the password, you can reset it to the default and then change it.

To reset the nsroot password, you must boot the appliance into single user mode, mount the file systems in read/write mode, and remove the set NetScaler user nsroot entry from the ns.conf file. You can then reboot, log on with the default password, and choose a new password.

To reset the nsroot password

1. Connect a computer to the console port of the NetScaler ADC and log on.
   Note: You cannot log on by using SSH to perform this procedure; you must connect directly to the appliance.
2. Reboot the NetScaler ADC.
3. Press CTRL+C when the following message appears:
   Press [Ctrl-C] for command prompt, or any other key to boot immediately.

   Booting [kernel] in # seconds.

4. Run the following command to start the NetScaler in a single user mode:
   boot -s

   Note: If boot -s does not work, then try reboot -- -s and appliance will reboot in single user mode.
   After the appliance boots, it displays the following message:

   Enter full path name of shell or RETURN for /bin/sh:

5. Press ENTER key to display the # prompt, and type the following commands to mount the file systems:
   1. Run the following command to check the disk consistency:
      fsck /dev/ad0s1a

      Note: Your flash drive will have a specific device name depending on your NetScaler; hence, you have to replace ad0s1a in the preceding command with the appropriate device name.
   2. Run the following command to display the mounted partitions:
      df

      If the flash partition is not listed, you need to mount it manually.

   3. Run the following command to mount the flash drive:
      mount /dev/ad0s1a /flash

6. Run the following command to change to the nsconfig directory:
   cd /flash/nsconfig

7. Run the following commands to rewrite the ns.conf file and remove the set of system commands defaulting to the nsroot user:
   1. Run the following command to create a new configuration file that does not have commands defaulting to the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsroot user:
grep –v “set system user nsroot” ns.conf > new.conf

2. Run the following command to make a backup of the existing configuration file:
   mv ns.conf old.ns.conf

3. Run the following command to rename the new.conf file to ns.conf:
   mv new.conf ns.conf

8. Run the following command to reboot the NetScaler:
   reboot

9. Log on using the default nsroot user credentials.
10. Run the following command to reset the nsroot user password:
    set system user nsroot <New_Password>

    Note: To use the "?” character in a password string, precede this character with the “\” character.

    For example, yourexamplepasswd\? is set for the nsroot account after you perform the following operation:

    > set system user nsroot yourexamplepasswd\?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example of a User Scenario

Sep 06, 2013

The following example shows how to create a complete set of user accounts, groups, and command policies and bind each policy to the appropriate groups and users. The company, Example Manufacturing, Inc., has three users who can access the NetScaler appliance:

- **John Doe.** The IT manager. John needs to be able to see all parts of the NetScaler configuration but does not need to modify anything.
- **Maria Ramiez.** The lead IT administrator. Maria needs to be able to see and modify all parts of the NetScaler configuration except for NetScaler commands (which local policy dictates must be performed while logged on as nsroot).
- **Michael Baldrock.** The IT administrator in charge of load balancing. Michael needs to be able to see all parts of the NetScaler configuration, but needs to modify only the load balancing functions.

The following table shows the breakdown of network information, user account names, group names, and command policies for the sample company.

Table 1. Sample Values for Creating Entities

Field	Value	Note
NetScaler host name	ns01.example.net	N/A
User accounts	johnd, mariar, and michaelb	John Doe, IT manager, Maria Ramirez, IT administrator and Michael Baldrock, IT administrator.
Groups	Managers and SysOps	All managers and all IT administrators.
Command Policies	read_all, modify_lb, and modify_all	Allow complete read-only access, Allow modify access to load balancing, and Allow complete modify access.

The following description walks you through the process of creating a complete set of user accounts, groups, and command policies on the NetScaler appliance named ns01.example.net.

The description includes procedures for binding the appropriate user accounts and groups to one another, and binding appropriate command policies to the user accounts and groups.

This example illustrates how you can use prioritization to grant precise access and privileges to each user in the IT department.

The example assumes that initial installation and configuration have already been performed on the NetScaler.

Configuration steps

1. Use the procedure described in "Configuring User Accounts" to create user accounts **johnd**, **mariar**, and **michaelb**.
2. Use the procedure described in "Configuring User Groups" to create user groups **Managers** and **SysOps**, and then bind the users **mariar** and **michaelb** to the **SysOps** group and the user **johnd** to the **Managers** group.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. Use the procedure described in "Creating Custom Command Policies" to create the following command policies:
   - **read_all** with action **Allow** and command spec "(^show\s+(?!system)(?!ns ns.conf)(?!ns runningConfig).*)|(^stat.*)"
   - **modify_lb** with action as **Allow** and the command spec "^set\s+lb\s+.*$"
   - **modify_all** with action as **Allow** and the command spec **"^\S+\s+(?!system).*"**
4. Use the procedure described in "Binding Command Policies to Users and Groups" to bind the **read_all** command policy to the **SysOps** group, with priority value **1**.
5. Use the procedure described in "Binding Command Policies to Users and Groups" to bind the **modify_lb** command policy to user **michaelb**, with priority value **5**.

The configuration you just created results in the following:

- John Doe, the IT manager, has read-only access to the entire NetScaler configuration, but he cannot make modifications.
- Maria Ramirez, the IT lead, has near-complete access to all areas of the NetScaler configuration, having to log on only to perform NetScaler-level commands.
- Michael Baldrock, the IT administrator responsible for load balancing, has read-only access to the NetScaler configuration, and can modify the configuration options for load balancing.

The set of command policies that applies to a specific user is a combination of command policies applied directly to the user's account and command policies applied to the group(s) of which the user is a member.

Each time a user enters a command, the operating system searches the command policies for that user until it finds a policy with an ALLOW or DENY action that matches the command. When it finds a match, the operating system stops its command policy search and allows or denies access to the command.

If the operating system finds no matching command policy, it denies the user access to the command, in accordance with the NetScaler appliance's default deny policy.

Note: When placing a user into multiple groups, take care not to cause unintended user command restrictions or privileges. To avoid these conflicts, when organizing your users in groups, bear in mind the NetScaler command policy search procedure and policy ordering rules.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring External User Authentication

Jun 01, 2015

External user authentication is the process of authenticating the users of the Citrix NetScaler appliance by using an external authentication server. The NetScaler supports LDAP, RADIUS, and TACACS+ authentication servers. To configure external user authentication, you must create authentication policies. You can configure one or many authentication policies, depending on your authentication needs. An authentication policy consists of an expression and an action. Authentication policies use NetScaler classic expressions, which are described in detail in "Policy Configuration and Reference."

After creating an authentication policy, you bind it to the system global entity and assign a priority to it. You can create simple server configurations by binding a single authentication policy to the system global entity. Or, you can configure a cascade of authentication servers by binding multiple policies to the system global entity. If no authentication policies are bound to the system, users are authenticated by the onboard system.

This document includes the following details:

- Configuring LDAP Authentication
- Configuring RADIUS Authentication
- Configuring TACACS+ Authentication
- Binding the Authentication Policies to the System Global Entity

Configuring LDAP Authentication

Updated: 2014-12-29

You can configure the NetScaler appliance to authenticate user access with one or more LDAP servers. LDAP authorization requires identical group names in Active Directory, on the LDAP server, and on the appliance. The characters and case must also be the same.

By default, LDAP authentication is secured by using SSL/TLS protocol. There are two types of secure LDAP connections. In the first type, the LDAP server accepts the SSL/TLS connection on a port separate from the port used to accept clear LDAP connections. After users establish the SSL/TLS connection, LDAP traffic can be sent over the connection. The second type allows both unsecure and secure LDAP connections and is handled by a single port on the server. In this scenario, to create a secure connection, the client first establishes a clear LDAP connection. Then the LDAP command StartTLS is sent to the server over the connection. If the LDAP server supports StartTLS, the connection is converted to a secure LDAP connection by using TLS.

The port numbers for LDAP connections are:

- 389 for unsecured LDAP connections
- 636 for secure LDAP connections
- 3268 for Microsoft unsecure LDAP connections
- 3269 for Microsoft secure LDAP connections

LDAP connections that use the StartTLS command use port number 389. If port numbers 389 or 3268 are configured on the appliance, it tries to use StartTLS to make the connection. If any other port number is used, connection attempts use SSL/TLS. If StartTLS or SSL/TLS cannot be used, the connection fails.

When configuring the LDAP server, the case of the alphabetic characters must match that on the server and on the appliance. If the root directory of the LDAP server is specified, all of the subdirectories are also searched to find the user attribute. In large directories, this can affect performance. For this reason, Citrix recommends that you use a specific organizational unit (OU).

The following table lists examples of user attribute fields for LDAP servers.

Table 1. User Attribute Fields for LDAP Servers

LDAP server	User attribute	Case sensitive?
Microsoft Active Directory	Server sAMAccountName	No
Novell eDirectory	cn	Yes
IBM Directory Server	uid	Yes
Lotus Domino	CN	Yes
Sun ONE directory (formerly iPlanet)	uid or cn	Yes

The following table lists examples of the base distinguished name (DN).

Table 2. Examples of Base Distinguished Name

LDAP server	Base DN
Microsoft Active Directory	DC=citrix, DC=local
Novell eDirectory	dc=citrix, dc=net
IBM Directory Server	cn=users
Lotus Domino	OU=City, O=Citrix, C=US

© 1999-2017 Citrix Systems, Inc. All rights reserved.

LDAP server	Base DN
Sun ONE directory (formerly iPlanet)	ou=People, dc=citrix, dc=com

The following table lists examples of the bind distinguished name (DN).

Table 3. Examples of Bind Distinguished Name

LDAP server	Bind DN
Microsoft Active Directory	CN=Administrator, CN=Users, DC=citrix, DC=local
Novell eDirectory	cn=admin, dc=citrix, dc=net
IBM Directory Server	LDAP_dn
Lotus Domino	CN=Notes Administrator, O=Citrix, C=US
Sun ONE directory (formerly iPlanet)	uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

To configure LDAP authentication by using the command line interface

At the command prompt, do the following:

1. Create an LDAP action.
   add authentication ldapAction <name> {-serverIP <ip_addr|ipv6_addr|*> | {-serverName <string>}} >] [-authTimeout <positive_integer>] [-ldapBase <string>] [-ldapBindDn <string>] {-ldapBindDnPassword } [-ldapLoginName <string>] [-groupAttrName <string>] [-subAttributeName <string>]

   Example:
   add authentication ldapAction ldap70 -serverIP <IP> -authTimeout 30 -ldapBase "CN=xxxxx,DC=xxxx,DC=xxx" -ldapBindDn "CN=xxxxx,CN=xxxxx,DC=xxxx,DC=xxx" -ldapBindDnPassword abcd -lda
2. Create an LDAP policy.
   add authentication ldapPolicy <name> <rule> [<reqAction>]

   Example:
   add authentication ldappolicy ldap_pol ns_true ldap70
3. Bind the LDAP policy to the following bind points at which the policy will be evaluated.
   - **System Global**: bind system global <policyName> [-priority <positive_integer>]
   - **VPN Global**: bind vpn global <policyName> [-priority <positive_integer>]
   - **Authentication Server**: bind authentication vserver <name> [-policy <string> [-priority <positive_integer>]
   - **VPN Server**: bind vpn vserver <name> [-policy <string> [-priority <positive_integer>]

To configure LDAP authentication by using the configuration utility

Navigate to System > Authentication > LDAP, and create the LDAP authentication policy.

Determining attributes in the LDAP directory

If you need help determining your LDAP directory attributes, you can easily look them up with the free LDAP browser from Softerra.

You can download the LDAP browser from the Softerra LDAP Administrator Web site at http://www.ldapbrowser.com. After the browser is installed, set the following attributes:

- The host name or IP address of your LDAP server.
- The port of your LDAP server. The default is 389.
- The base DN field can be left blank.
- The information provided by the LDAP browser can help you determine the base DN needed for the Authentication tab.
- The Anonymous Bind check determines whether the LDAP server requires user credentials for the browser to connect to it. If the LDAP server requires credentials, leave the check box cleared.

After completing the settings, the LDAP browser displays the profile name in the left pane and connects to the LDAP server.

Configuring RADIUS Authentication

Updated: 2014-08-08

You can configure the NetScaler appliance to authenticate user access with one or more RADIUS servers. If you are using RSA SecurID, SafeWord, or Gemalto Protiva products, use a RADIUS server.

Your configuration might require using a network access server IP address (NAS IP) or a network access server identifier (NAS ID). When configuring the appliance to use a RADIUS authentication server, use the following guidelines:

- If you enable use of the NAS IP, the appliance sends its configured IP address to the RADIUS server, rather than the source IP address used in establishing the RADIUS connection.
- If you configure the NAS ID, the appliance sends the identifier to the RADIUS server. If you do not configure the NAS ID, the appliance sends its host name to the RADIUS server.
- When the NAS IP is enabled, the appliance ignores any NAS ID that was configured by using the NAS IP to communicate with the RADIUS server.

To configure RADIUS authentication by using the configuration utility

Navigate to System > Authentication > Radius, and create the RADIUS authentication policy.

Choosing RADIUS authentication protocols

The NetScaler appliance supports implementations of RADIUS that are configured to use any of several protocols for user authentication, including:

- Password Authentication Protocol
- Challenge-Handshake Authentication Protocol (CHAP)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Microsoft Challenge-Handshake Authentication Protocol (MS-CHAP Version 1 and Version 2)

If your deployment of the appliance is configured to use RADIUS authentication and your RADIUS server is configured to use Password Authentication Protocol, you can strengthen user authentication by assigning a strong shared secret to the RADIUS server. Strong RADIUS shared secrets consist of random sequences of uppercase and lowercase letters, numbers, and punctuation, and are at least 22 characters long. If possible, use a random character generation program to determine RADIUS shared secrets.

To further protect RADIUS traffic, assign a different shared secret to each appliance or virtual server. When you define clients on the RADIUS server, you can also assign a separate shared secret to each client. If you do this, you must configure separately each policy that uses RADIUS authentication.

Shared secrets are configured on the appliance when a RADIUS policy is created.

Configuring IP address extraction

You can configure the appliance to extract the IP address from a RADIUS server. When a user authenticates with the RADIUS server, the server returns a framed IP address that is assigned to the user. The following are attributes for IP address extraction:
- Allows a remote RADIUS server to supply an IP address from the internal network for a user logged on to the appliance.
- Allows configuration for any RADIUS attribute using the type ipaddress, including those that are vendor encoded.

When configuring the RADIUS server for IP address extraction, you configure the vendor identifier and the attribute type.

The vendor identifier enables the RADIUS server to assign an IP address to the client from a pool of IP addresses that are configured on the RADIUS server. The vendor ID and attributes are used to make the association between the RADIUS client and the RADIUS server. The vendor ID is the attribute in the RADIUS response that provides the IP address of the internal network. A value of zero indicates that the attribute is not vendor encoded. The attribute type is the remote IP address attribute in a RADIUS response. The minimum value is one and the maximum value is 255.

A common configuration is to extract the RADIUS attribute *framed IP address*. The vendor ID is set to zero or is not specified. The attribute type is set to eight.

To configure IP address extraction by using the configuration utility

1. Navigate to System > Authentication > Radius, and select a policy.
2. Modify the server parameters and set relevant values in Group Vendor Identifier and Group Attribute Type fields.

Configuring TACACS+ Authentication

Updated: 2014-08-07

You can configure a TACACS+ server for authentication. Similar to RADIUS authentication, TACACS+ uses a secret key, an IP address, and the port number. The default port number is 49. To configure the appliance to use a TACACS+ server, provide the server IP address and the TACACS+ secret. The port needs to be specified only when the server port number in use is something other than the default port number of 49.

To configure TACACS+ authentication by using the configuration utility

Navigate to System > Authentication > TACACS, and create the TACACS authentication policy.
After the TACACS+ server settings are configured on the appliance, bind the policy to the system global entity. For more information about binding authentication policies globally, see "Binding the Authentication Policies to the System Global Entity."

Binding the Authentication Policies to the System Global Entity

Updated: 2014-12-30

When the authentication policies are configured, bind the policies to the system global entity.

To bind an authentication policy to system global using the command line interface

At the command line prompt, do the following:

bind system global <policyName> [-priority <positive_integer>]

Example:
bind system global ldappol1 -priority 10

To bind an authentication policy to system global using the configuration utility

1. Navigate to System > Authentication, and select the authentication type.
2. On the Policies tab, click Global Bindings and bind the authentication policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

TCP Configurations

Nov 01, 2017

TCP configurations for a NetScaler appliance can be specified in an entity called a TCP profile, which is a collection of TCP settings. The TCP profile can then be associated with services or virtual servers that want to use these TCP configurations.

A default TCP profile can be configured to set the TCP configurations that will be applied by default, globally to all services and virtual servers.

Note: When a TCP parameter has different values for service, virtual server, and globally, the value of the most-specific entity (the service) is given the highest precedence.
The NetScaler appliance also provides other approaches for configuring TCP. Read on for more information.

The NetScaler appliance supports the following TCP capabilities:

- Defending TCP against spoofing attacks. The NetScaler implementation of window attenuation is RFC 4953 compliant.
- Explicit Congestion Notification (ECN), which sends notification of the network congestion status to the sender of the data and takes corrective measures for data congestion or data corruption. The NetScaler implementation of ECN is RFC 3168 compliant.
- Round Trip Time Measurement (RTTM) using the TimeStamp option. For the TimeStamp option to work, at least one side of the connection (client or server) must support it. The NetScaler implementation of TimeStamp option is RFC 1323 compliant.
- Detection of spurious retransmissions can be done using TCP duplicate selective acknowledgement (D-SACK) and forward RTO-Recovery (F-RTO). In case of spurious retransmissions, the congestion control configurations are reverted to their original state. The NetScaler implementation of D-SACK is RFC 2883 compliant, and F-RTO is RFC 5682 compliant.
- Congestion control using New-Reno, BIC, CUBIC,, Nile and TCP Westwood algorithms.
- Window scaling to increase the TCP receive window size beyond its maximum value of 65,535 bytes.
  Note: Before configuring window scaling, make sure that:
  - You do not set a high value for the scale factor, because this could have adverse effects on the appliance and the network.
  - You do not configure window scaling unless you clearly know why you want to change the window size.
  - Both hosts in the TCP connection send a window scale option during connection establishment. If only one side of a connection sets this option, window scaling is not used for the connection.
  - Each connection for same session is an independent window scaling session. For example, when a client's request and the server's response flow through the appliance, it is possible to have window scaling between the client and the appliance without window scaling between the appliance and the server.
- TCP maximum congestion window size that is user configurable. The default value is 8190 bytes.
- Selective acknowledgment (SACK), using which the data receiver (either a NetScaler appliance or a client) notifies the sender about all the segments that have been received successfully.
- Forward acknowledgment (FACK) avoids TCP congestion by explicitly measuring the total number of data bytes outstanding in the network, and helping the sender (either a NetScaler ADC or a client) control the amount of data injected into the network during retransmission timeouts.
- TCP connection multiplexing enables reuse of existing TCP connections. The NetScaler appliance stores established TCP connections to the reuse pool. Whenever a client request is received, appliance checks for an available connection in the reuse pool and serves the new client if the connection is available. If it is unavailable, the appliance creates a new connection for the client request and stores the connection to the reuse pool.
  NetScaler supports connection multiplexing for HTTP, SSL, and DataStream connection types.

- Dynamic receive buffering allows the receive buffer to be adjusted dynamically based on memory and network conditions.
- MPTCP connections between client and NetScaler. MPTCP connections are not supported between NetScaler and the backend server. The NetScaler implementation of MPTCP is RFC 6824 compliant.

  Note:
  - To establish an MPTCP connection, both the client and the Netscaler appliance must support the same MPTCP version. If you use the NetScaler appliance as an MPTCP gateway for your servers, the servers do not have to support MPTCP. When the client starts a new MPTCP connection, the appliance identifies the client’s MPTPC version from the MP_CAPABALE option in the SYN packet. If the client’s version is higher than the one supported on the appliance, the appliance indicates its highest version in the MP_CAPABALE option of the SYN-ACK packet. The client then falls back to a lower version and sends the version number in the MP_CAPABALE option

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

of the ACK packet. If that version is supportable, the appliance continues the MPTCP connection. Otherwise, the appliance falls back to a regular TCP.
  - The NetScaler appliance does not initiate subflows (MP_JOIN's). The appliance expects the client to initiate subflows.
- TCP keep-alive to monitor the TCP connections to verify if the peers are up.
- Extracting the TCP/IP path overlay option and inserting client-IP HTTP header. Extracting TCP/IP path overlay and inserting client-IP HTTP header. Data transport through overlay networks often uses connection termination or Network Address Translation (NAT), in which the IP address of the source client is lost. To avoid this, the Netscaler appliance extracts the TCP/IP path overlay option and inserts the source client's IP address into the HTTP header. With the IP address in the header, the web server can identify the source client that made the connection. The extracted data is valid for lifetime of the TCP connection and therefore, this prevents the next hop host from having to interpret the option again. This option is applicable only for web services that have the client-IP insertion option enabled. For more information, see Client Insertion on backend topic.

Additionally, NetScaler provides configuration support for the following:

- TCP segmentation offload.
- Synchronizing cookie for TCP handshake with clients. Disabling this capability prevents SYN attack protection on the NetScaler appliance.
- Learning MSS to enable MSS learning for all the virtual servers configured on the appliance.

This document includes the following details:
- Setting Global TCP Parameters
- Setting Service or Virtual Server Specific TCP Parameters
- Built-in TCP Profiles
- Sample TCP Configurations

Setting Global TCP Parameters

The NetScaler appliance allows you to specify values for TCP parameters that are applicable to all NetScaler services and virtual servers. This can be done using:

- Default TCP profile
- Global TCP command
- TCP buffering feature

Note: The recvBuffSize parameter of the set ns tcpParam command is deprecated from release 9.2 onwards. In later releases, set the buffer size by using the bufferSize parameter of the set ns tcpProfile command. If you upgrade to a release where the recvBuffSize parameter is deprecated, the bufferSize parameter is set to its default value.

Default TCP profile

A TCP profile, named as nstcp_default_profile, is used to specify TCP configurations that will used if no TCP configurations are provided at the service or virtual server level.

Note:
- Not all TCP parameters can be configured through the default TCP profile. Some settings have to be performed by using the global TCP command (see section below).
- The default profile does not have to be explicitly bound to a service or virtual server.

To configure the default TCP profile

- Using the command line interface, at the command prompt enter:

  set ns tcpProfile nstcp_default_profile ...

- On the configuration utility, navigate to System > Profiles, click TCP Profiles and update nstcp_default_profile.

Global TCP command

Another approach you can use to configure global TCP parameters is the global TCP command. In addition to some unique parameters, this command duplicates some parameters that can be set by using a TCP profile. Any update made to these duplicate parameters is reflected in

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

the corresponding parameter in the default TCP profile.

For example, if the SACK parameter is updated using this approach, the value is reflected in the SACK parameter of the default TCP profile (nstcp_default_profile).

Note: Citrix recommends that you use this approach only for TCP parameters that are not available in the default TCP profile.
To configure the global TCP command

- Using the command line interface, at the command prompt enter:

  set ns tcpParam ...

- On the configuration utility, navigate to System > Settings, click Change TCP parameters and update the required TCP parameters.

TCP buffering feature

NetScaler provides a feature called TCP buffering that you can use to specify the TCP buffer size. The feature can be enabled globally or at service level.

Note: The buffer size can also be configured in the default TCP profile. If the buffer size has different values in the TCP buffering feature and the default TCP profile, the greater value is applied.
To configure the TCP buffering feature globally

- At the command prompt enter:

  enable ns mode TCPB

  set ns tcpbufParam -size <positiveInteger> -memLimit <positiveInteger>

- On the configuration utility, navigate to System > Settings, click Configure Modes and select TCP Buffering.
  And, navigate to System > Settings, click Change TCP parameters and specify the values for Buffer size and Memory usage limit.

Setting Service or Virtual Server Specific TCP Parameters

Using TCP profiles, you can specify TCP parameters for services and virtual servers. You must define a TCP profile (or use a built-in TCP profile) and associate the profile with the appropriate service and virtual server.

Note:
- You can also modify the TCP parameters of default profiles as per your requirements. For more information on built-in TCP profiles, see Built-in TCP Profiles.
- You can specify the TCP buffer size at service level using the parameters specified by the TCP buffering feature.

To specify service or virtual server level TCP configurations by using the command line interface

At the command prompt, perform the following:

1. Configure the TCP profile.
   set ns tcpProfile <profile-name>...

2. Bind the TCP profile to the service or virtual server.
   To bind the TCP profile to the service:

   set service <name> ....

   Example:
   > set service service1 -tcpProfileName profile1
   To bind the TCP profile to the virtual server:

   set lb vserver <name> ....

   Example:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> set lb vserver lbvserver1 -tcpProfileName profile1
```

To specify service or virtual server level TCP configurations by using the configuration utility

At the configuration utility, perform the following:

1. Configure the TCP profile.
   Navigate to System > Profiles > TCP Profiles, and create the TCP profile.

2. Bind the TCP profile to the service or virtual server.
   Navigate to Traffic Management > Load Balancing > Services/Virtual Servers, and create the TCP profile, which should be bound to the service or virtual server.

Built-in TCP Profiles

For convenience of configuration, the NetScaler provides some built-in TCP profiles. Review the built-in profiles listed below and select a profile and use it as it is or modify it to meet your requirements. You can bind these profiles to your required services or virtual servers.

Table 1. Built-in TCP Profiles

Built-in profile	Description
nstcp_default_profile	Represents the default global TCP settings on the appliance.
nstcp_default_tcp_lan	Useful for back-end server connections, where these servers reside on the same LAN as the appliance.
nstcp_default_tcp_lan_thin_stream	Similar to the nstcp_default_tcp_lan profile; however, the settings are tuned to small size packet flows.
nstcp_default_tcp_interactive_stream	Similar to the nstcp_default_tcp_lan profile; however, it has a reduced delayed ACK timer and ACK on PUSH packet settings.
nstcp_default_tcp_lfp	Useful for long fat pipe networks (WAN) on the client side. Long fat pipe networks have long delay, high bandwidth lines with minimal packet drops.
nstcp_default_tcp_lfp_thin_stream	Similar to the nstcp_default_tcp_lfp profile; however, the settings are tuned for small size packet flows.
nstcp_default_tcp_lnp	Useful for long narrow pipe networks (WAN) on the client side. Long narrow pipe networks have considerable packet loss once in a while.
nstcp_default_tcp_lnp_thin_stream	Similar to the nstcp_default_tcp_lnp profile; however, the settings are tuned for small size packet flows.
nstcp_internal_apps	Useful for internal applications on the appliance (for example, GSLB sitesyncing). This contains tuned window scaling and SACK options for the desired applications. This profile should not be bound to applications other than internal applications.
nstcp_default_Mobile_profile	Useful for mobile devices.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Built-in profile	Description
nstcp_default_XA_XD_profile	Useful for a XenApp or XenDesktop deployment.

Sample TCP Configurations

Sample command line interface examples for configuring the following:

- Defending TCP against spoofing attacks
- Explicit Congestion Notification (ECN)
- Selective ACKnowledgment (SACK)
- Forward ACKnowledgment (FACK)
- Window Scaling (WS)
- Maximum Segment Size (MSS)
- NetScaler to learn the MSS of a virtual server
- TCP keep-alive
- Buffer size - using TCP profile
- Buffer size - using TCP buffering feature
- MPTCP
- Congestion control
- Dynamic receive buffering

Defending TCP against spoofing attacks

Enable the NetScaler to defend TCP against spoof attacks. By default the "rstWindowAttenuation" parameter is disabled. This parameter is enabled to protect the appliance against spoofing. If you enable, it will reply with corrective acknowledgement (ACK) for an invalid sequence number. Possible values are Enabled, Disabled.

Where RST window attenuation parameter protects the appliance against spoofing. When enabled, will reply with corrective ACK when a sequence number is invalid.

```
> set ns tcpProfile profile1 -rstWindowAttenuate ENABLED -spoofSynDrop ENABLED
 Done
> set lb vserver lbvserver1 -tcpProfileName profile1
 Done
```

Explicit Congestion Notification (ECN)

Enable ECN on the required TCP profile.

```
> set ns tcpProfile profile1 -ECN ENABLED
 Done
> set lb vserver lbvserver1 -tcpProfileName profile1
 Done
```

Selective ACKnowledgment (SACK)

Enable SACK on the required TCP profile.

```
> set ns tcpProfile profile1 -SACK ENABLED
 Done
> set lb vserver lbvserver1 -tcpProfileName profile1
 Done
```

Forward ACKnowledgment (FACK)

Enable FACK on the required TCP profile.

```
> set ns tcpProfile profile1 -FACK ENABLED
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> set lb vserver lbvserver1 -tcpProfileName profile1
 Done
```

Window Scaling (WS)

Enable window scaling and set the window scaling factor on the required TCP profile.

```
> set ns tcpProfile profile1 –WS ENABLED –WSVal 9
 Done
> set lb vserver lbvserver1 -tcpProfileName profile1
 Done
```

Maximum Segment Size (MSS)

Update the MSS related configurations.

```
> set ns tcpProfile profile1 –mss 1460 - maxPktPerMss 512
 Done
> set lb vserver lbvserver1 -tcpProfileName profile1
 Done
```

NetScaler to learn the MSS of a virtual server

Enable the NetScaler to learn the VSS and update other related configurations.

```
> set ns tcpParam -learnVsvrMSS ENABLED –mssLearnInterval 180 -mssLearnDelay 3600
 Done
```

TCP keep-alive

Enable TCP keep-alive and update other related configurations.

```
> set ns tcpProfile profile1 –KA ENABLED –KaprobeUpdateLastactivity ENABLED -KAconnIdleTime 900 -KAmaxProbes 3 -KaprobeInterval 75
 Done
> set lb vserver lbvserver1 -tcpProfileName profile1
 Done
```

Buffer size - using TCP profile

Specify the buffer size.

```
> set ns tcpProfile profile1 –bufferSize 8190
 Done
> set lb vserver lbvserver1 -tcpProfileName profile1
 Done
```

Buffer size - using TCP buffering feature

Enable the TCP buffering feature (globally or for a service) and then specify the buffer size and the memory limit.

```
> enable ns feature TCPB
 Done
> set ns tcpbufParam -size 64 -memLimit 64
 Done
```

MPTCP

Enable MPTCP and then set the optional MPTCP configurations.

```
> set ns tcpProfile profile1 -mptcp ENABLED
 Done
> set ns tcpProfile profile1 -mptcpDropDataOnPreEstSF ENABLED -mptcpFastOpen ENABLED -mptcpSessionTimeout 7200
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> set ns tcpparam -mptcpConCloseOnPassiveSF ENABLED -mptcpChecksum ENABLED -mptcpSFtimeout 0 -mptcpSFReplaceTimeout 10
-mptcpMaxSF 4 -mptcpMaxPendingSF 4 -mptcpPendingJoinThreshold 0 -mptcpRTOsToSwitchSF 2 -mptcpUseBackupOnDSS ENABLED
 Done
```

Congestion control

Set the required TCP congestion control algorithm.

```
> set ns tcpProfile profile1 -flavor Westwood
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

Dynamic receive buffering

Enable dynamic receive buffering on the required TCP profile.

```
> set ns tcpProfile profile1 -dynamicReceiveBuffering ENABLED
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Configurations

Jun 03, 2015

HTTP configurations for a NetScaler appliance can be specified in an entity called an HTTP profile, which is a collection of HTTP settings. The HTTP profile can then be associated with services or virtual servers that want to use these HTTP configurations.

A default HTTP profile can be configured to set the HTTP configurations that will be applied by default, globally to all services and virtual servers.

Note: When a HTTP parameter has different values for service, virtual server, and globally, the value of the most-specific entity (the service) is given the highest precedence.
The NetScaler appliance also provides other approaches for configuring HTTP. Read on for more information.

The NetScaler supports the following HTTP capabilities:

- WebSocket protocol which allows browsers and other clients to create a bi-directional, full duplex TCP connection to the servers. The NetScaler implementation of WebSocket is RFC 6455 compliant.
- SPDY (Speedy). For more information, see SPDY.

This document includes the following details:

- Setting Global HTTP Parameters
- Setting Service or Virtual Server Specific HTTP Parameters
- Built-in HTTP Profiles
- Sample HTTP Configurations

Setting Global HTTP Parameters

Updated: 2014-08-06

The NetScaler appliance allows you to specify values for HTTP parameters that are applicable to all NetScaler services and virtual servers. This can be done using:

- Default HTTP profile
- Global HTTP command

Default HTTP profile

A HTTP profile, named as nshttp_default_profile, is used to specify HTTP configurations that will be used if no HTTP configurations are provided at the service or virtual server level.

Note:

- Not all HTTP parameters can be configured through the default HTTP profile. Some settings have to be performed by using the global HTTP command (see section below).
- The default profile does not have to be explicitly bound to a service or virtual server.

To configure the default HTTP profile

- Using the command line interface, at the command prompt enter:

  set ns httpProfile nshttp_default_profile ...

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- On the configuration utility, navigate to System > Profiles, click HTTP Profiles and update nshttp_default_profile.

Global HTTP command

Another approach you can use to configure global HTTP parameters is the global HTTP command. In addition to some unique parameters, this command duplicates some parameters that can be set by using a HTTP profile. Any update made to these duplicate parameters is reflected in the corresponding parameter in the default HTTP profile.

For example, if the maxReusePool parameter is updated using this approach, the value is reflected in the maxReusePool parameter of the default HTTP profile (nshttp_default_profile).

Note: Citrix recommends that you use this approach only for HTTP parameters that are not available in the default HTTP profile.
To configure the global HTTP command

- Using the command line interface, at the command prompt enter:

  set ns httpParam ...

- On the configuration utility, navigate to System > Settings, click Change HTTP parameters and update the required HTTP parameters.

Setting Service or Virtual Server Specific HTTP Parameters

Using HTTP profiles, you can specify HTTP parameters for services and virtual servers. You must define a HTTP profile (or use a built-in HTTP profile) and associate the profile with the appropriate service and virtual server.

Note: You can also modify the HTTP parameters of default profiles as per your requirements. For more information on built-in HTTP profiles, see Built-in HTTP Profiles.

To specify service or virtual server level HTTP configurations by using the command line interface

At the command prompt, perform the following:

1. Configure the HTTP profile.
   set ns httpProfile <profile-name>...

2. Bind the HTTP profile to the service or virtual server.
   To bind the HTTP profile to the service:

   set service <name> .....

   Example:
   > set service service1 -httpProfileName profile1
   To bind the HTTP profile to the virtual server:

   set lb vserver <name> .....

   Example:
   > set lb vserver lbvserver1 -httpProfileName profile1

To specify service or virtual server level HTTP configurations by using the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

configuration utility

At the configuration utility, perform the following:

1. Configure the HTTP profile.
   Navigate to System > Profiles > HTTP Profiles, and create the HTTP profile.

2. Bind the HTTP profile to the service or virtual server.
   Navigate to Traffic Management > Load Balancing > Services/Virtual Servers, and create the HTTP profile, which should be bound to the service/virtual server.

Built-in HTTP Profiles

For convenience of configuration, the NetScaler provides some built-in HTTP profiles. Review the profiles listed below and use it as it is or modify it to meet your requirements. You can bind these profiles to the required services or virtual servers.

Table 1. Built-in HTTP Profiles

Built-in profile	Description
nshttp_default_profile	Represents the default global HTTP settings on the appliance.
nshttp_default_strict_validation	Settings for deployments that require strict validation of HTTP requests and responses.

Sample HTTP Configurations

Sample command line interface examples to configure the following:

- HTTP band statistics
- WebSocket connections

HTTP band statistics

Specify the band size for HTTP requests and responses.

```
> set protocol httpBand reqBandSize 300 respBandSize 2048
Done
> show protocol httpband -type REQUEST
```

WebSocket connections

Enable webSocket on the required HTTP profile.

```
> set ns httpProfile http_profile1 -webSocket ENABLED
Done
> set lb vserver lbvserver1 -httpProfileName profile1
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring HTTP/2 on the NetScaler Appliance

Dec 02, 2016

Note: The HTTP/2 functionality is supported on the NetScaler MPX, VPX, and SDX models. For NetScaler VPX, the HTTP/2 functionality is supported from 11.0-66.x release onwards.

The problem with web application performance is directly related to trend toward increasing the page size and the number of objects on the web pages. HTTP/1.1 was developed to support smaller web pages, slower Internet connections, and more limited server hardware than are common today. It is not well suited for newer technologies such as JavaScript and cascading style sheets (CSS), nor for new media types such as Flash videos and graphics-rich images, because it can request only one resource per connection to the server. This limitation significantly increases the number of round trips, causing longer page-rendering time and reduced network performance.

The HTTP/2 protocol addresses these limitations by allowing communication to occur with less data transmitted over the network, and providing the ability to send multiple requests and responses across a single connection. At its core, HTTP/2 addresses the key limitations of HTTP/1.1 by using the underlying network connections more efficiently. It changes the way requests and responses travel over the network.

HTTP/2 is a binary protocol. It is more efficient to parse, more compact on the wire, and most importantly, it is less error-prone, compared to textual protocols like HTTP/1.1. The HTTP/2 protocol uses a binary framing layer that defines the frame type and how HTTP messages are encapsulated and transferred between the client and server.

The HTTP/2 protocol includes a lot of performance-enhancing changes that significantly improve performance, particularly for clients connecting over a mobile network.

The following table lists the major improvements in HTTP/2 over HTTP/1.1:

HTTP/2 Features	Description
Header Compression	HTTP headers have a lot of repetitive information and therefore consume unnecessary bandwidth during data transmission. HTTP/2 reduces bandwidth requirements by compressing the header and minimizing the requirement to transport HTTP headers with every request and response.
Connection Multiplexing	Latency can have a huge impact on page load times and the end user experience. Connection multiplexing overcomes this problem by sending multiple requests and responses across a single connection.
Server Push	Server push enables the server to proactively push content to the client browser, avoiding round trip delay. This feature caches the responses it thinks the client will need, reduces the number round trips, and improves the page rendering time. **Important**: The NetScaler appliance does not support the server push functionality.
No Head-of-line Blocking	Under HTTP/1.1, browsers can download one resource at a time per connection. When a browser has to download a large resource, it blocks all other resources from downloading until the first download is complete. HTTP/2 overcomes this problem with a multiplexing approach. It allows the client browser to download other web components in parallel over the same connection and display them as they become available.
Request Prioritization	Not all resources have equal priority when the browser renders a web page. To accelerate the load time, all modern browsers prioritize requests by type of asset, their location on the page, and even by learned priority from previous visits. With HTTP/1.1, the browser has limited ability to leverage the priority data, because this protocol does not support multiplexing, and there is no way to communicate request prioritization by the server. The result is unnecessary network latency. HTTP/2 overcomes this problem by allowing the browser to dispatch all requests. The browser can communicate its stream prioritization preference via stream dependencies and weights, enabling the servers to optimize response delivery. **Important**: The NetScaler appliance does not support the request prioritization functionality.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How HTTP/2 Works on a NetScaler appliance

A NetScaler appliance supports HTTP/2 connections with clients. It converts the HTTP/2 headers to HTTP/1.1 for communication with servers. When a service or virtual server associated with a profile configured for HTTP/2 receives a request from a client, the NetScaler appliance accumulates the HTTP/2 headers and decodes them to form HTTP/1.1 headers. After decoding the headers, the appliance validates the HTTP/1.1 headers and ensures that the data is intact. Once the validation is complete, the appliance forwards the HTTP/1.1 headers to the server.

When the appliance receives the response headers from the server, it converts all the HTTP/1.1 headers to HTTP/2 headers and sends them to the client.

HTTP/2 over an SSL (HTTPS) virtual server

If the HTTP/2 setting is enabled on the HTTP profile associated with an HTTPS virtual server, the NetScaler appliance uses the TLS ALPN extension (RFC 7301) to determine whether the HTTP client supports HTTP/2. If it does, the appliance chooses HTTP/2 as the application-layer protocol to transmit data between the client and the server NetScaler appliance (as described in RFC 7540 - Section 3.3).

The appliance uses the following order of preference when choosing the application-layer protocol through the TLS ALPN extension:

- HTTP/2 (if enabled in the HTTP profile)
- SPDY (if enabled in the HTTP profile)
- HTTP/1.1

HTTP/2 over an HTTP virtual server

If the HTTP/2 setting is enabled on the HTTP profile associated with the HTTP virtual server, the NetScaler appliance uses the HTTP protocol upgrade mechanism (as described in RFC 7540 - Section 3.2) to upgrade the connection to HTTP/2. The HTTP client requests a protocol upgrade and the appliance will upgrades the connection to HTTP/2.

Note: If the HTTP/2 setting is enabled on the HTTP profile associated with the virtual server, the NetScaler appliance does ***NOT*** support the direct use of HTTP/2, but still it acts as a TCP proxy by forwarding the HTTP/2 requests to servers.

Configuring HTTP/2 on the NetScaler Appliance

The HTTP/2 feature is part of the HTTP configurations on the NetScaler appliance. By default, the HTTP/2 feature is disabled. You have to enable the feature on a specific HTTP profile and associate the HTTP profile with services or virtual servers on which you want to use the HTTP/2 feature.

To know more about binding a HTTP profile to a virtual server, see Setting Service or Virtual Server Specific HTTP Parameters section on HTTP Configuration page.

Note: The HTTP/2 functionality will not work, if User Source IP (USIP) mode is enabled and the Proxy mode is disabled on the NetScaler appliance.

To configure HTTP/2 by using the command line

At the command prompt, type:

set ns httpProfile <name> [-**http2** (ENABLED | DISABLED)] [-**http2MaxHeaderListSize** <positive_integer>] [-**http2MaxFrameSize** <positive_integer>] [-**http2MaxConcurrentStreams** <positive_integer>] [-**http2InitialWindowSize** <positive_integer>] [-**http2HeaderTableSize** <positive_integer>]

To configure HTTP/2 by using the configuration utility

1. Create a HTTP/2 profile on the NetScaler appliance. Navigate to **System** > **Profiles**, select a HTTP Profile and then click **Add** to create a HTTP profile with **HTTP/2** checkbox selected.

2. [Optional] Configure HTTP/2 header table size.
   Navigate to **System** > **Profiles**, select a HTTP Profiles and click **Edit** to specify the maximum size of the header compression table in the **HTTP/2 Header Table Size** text box.

3. [Optional] Configure HTTP/2 initial window size
   Navigate to **System** > **Profiles**, select a HTTP Profiles and click **Edit** to specify the initial window size for an HTTP/2 stream in the **HTTP/2 Initial Window Size** text box.

4. [Optional] Configure HTTP/2 maximum concurrent streams.
   Navigate to **System** > **Profiles**, select a HTTP Profiles and click **Edit** to specify the maximum number of concurrent streams per connection in the **HTTP/2 Maximum Concurrent Streams** text box .

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

5. [Optional] Configure HTTP/2 maximum frame size.
   Navigate to **System** > **Profiles**, select a HTTP Profiles and click **Edit** to specify the maximum size of the frame in the **HTTP/2 Maximum Frame Size** text box.

6. [Optional] Configure HTTP/2 maximum header list size.
   Navigate to **System** > **Profiles**, select a HTTP Profiles and click **Edit** to specify the header size the **Maximum Header List Size** text box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SNMP

Nov 01, 2017

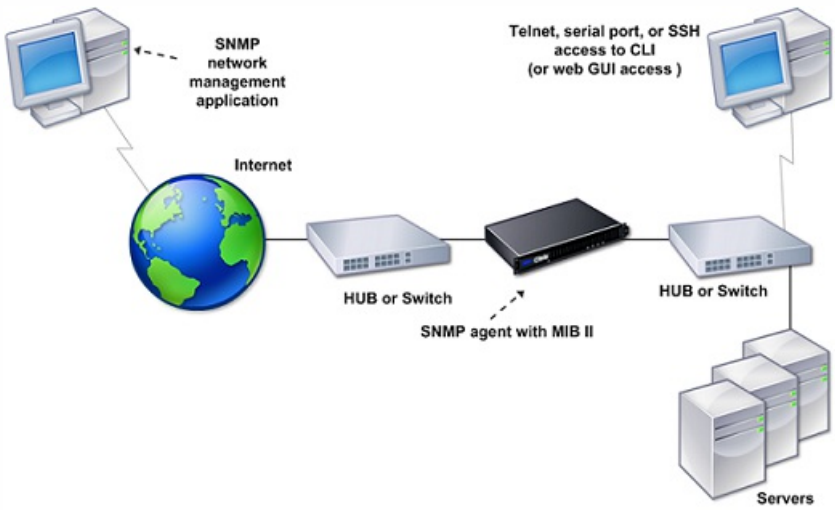
You can use Simple Network Management Protocol (SNMP) to configure the SNMP agent on the Citrix NetScaler appliance to generate asynchronous events, which are called *traps*. The traps are generated whenever there are abnormal conditions on the NetScaler. The traps are then sent to a remote device called a *trap listener*, which signals the abnormal condition on the NetScaler appliance. Or, you can query the SNMP agent for System-specific information from a remote device called an *SNMP manager*. The agent then searches the management information base (MIB) for the data requested and sends the data to the SNMP manager.

For information about SNMP parameters, traps, and its descriptions, see NetScaler SNMP OID reference.

The SNMP agent on the NetScaler can generate traps compliant with SNMPv1, SNMPv2, and SNMPv3. For querying, the SNMP agent supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3).

The following figure illustrates a network with a NetScaler that has SNMP enabled and configured. In the figure, each SNMP network management application uses SNMP to communicate with the SNMP agent on the NetScaler. The SNMP agent searches its management information base (MIB) to collect the data requested by the SNMP Manager and provides the information to the application.

Figure 1. *NetScaler Supporting SNMP*



Importing MIB Files to the SNMP Manager and Trap Listener

To monitor a NetScaler appliance, you must download the MIB object definition files. The MIB files include the following:

- MIB-2 groups SYSTEM, IF, ICMP, UDP, and SNMP.
- NetScaler-specific configuration and statistics.

You can obtain the MIB object definition files from the /netscaler/snmp directory or from the Downloads tab of the NetScaler GUI.

If the SNMP management application is other than WhatsUpGold, download the following files to the SNMP management application:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- NS-MIB-smiv1.mib. Used by SNMPv1 managers and trap listeners.
- NS-MIB-smiv2.mib. Used by SNMPv2 and SNMPv3 managers and SNMPv2 trap listeners.

If the SNMP management application is WhatsUpGold, download the following files to the SNMP management application:

- mib.txt
- traps.txt

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NetScaler to Generate SNMP Traps

Sep 14, 2016

You can configure the NetScaler appliance to generate asynchronous events, which are called *traps*. The traps are generated whenever there are abnormal conditions on the appliance. The traps are sent to a remote device called a *trap listener*. This helps administrators monitor the appliance and respond promptly to any issues.

The NetScaler appliance provides a set of condition entities called *SNMP alarms*. When the condition in any SNMP alarm is met, the appliance generates SNMP trap messages that are sent to the configured trap listeners. For example, when the LOGIN-FAILURE alarm is enabled, a trap message is generated and sent to the trap listener whenever there is a login failure on the appliance.

To configure the NetScaler appliance to generate traps, you need to enable and configure alarms. Then, you specify trap listeners to which the appliance will send the generated trap messages.

This document includes the following details:

- Enabling an SNMP Alarm
- Configuring Alarms
- Configuring SNMPv1 or SNMPv2 Traps
- Configuring SNMPv3 Traps
- Enabling Unconditional SNMP Trap Logging

Enabling an SNMP Alarm

The NetScaler appliance generates traps only for SNMP alarms that are enabled. Some alarms are enabled by default, but you can disable them.

When you enable an SNMP alarm, the appliance generates corresponding trap messages when some events occur. Some alarms are enabled by default.

To enable an SNMP alarm by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- enable snmp alarm <trapName>
- show snmp alarm <trapName>

To enable an SNMP alarm by using the configuration utility

1. Navigate to System > SNMP > Alarms, and select the alarm.
2. Click Actions and select Enable.

Configuring Alarms

The NetScaler appliance provides a set of condition entities called *SNMP alarms*. When the condition set for an SNMP alarm is met, the appliance generates SNMP traps messages that are sent to the configured trap listeners. For example, when the LOGIN-FAILURE alarm is enabled, a trap message is generated and sent to the trap listener whenever there is a login failure on the appliance.

You can assign an SNMP alarm with a severity level. When you do this, the corresponding trap messages are assigned that

© 1999-2017 Citrix Systems, Inc. All rights reserved.

severity level.

The following are the severity levels, defined on the appliance, in decreasing order of severity.

- Critical
- Major
- Minor
- Warning
- Informational

For example, if you set a warning severity level for the SNMP alarm named LOGIN-FAILURE, the trap messages generated when there is a login failure will be assigned with the warning severity level.

You can also configure an SNMP alarm to log the corresponding trap messages generated whenever the condition on that alarm is met.

To configure an SNMP alarm by using the command line interface

At the command prompt, type the following commands to configure an SNMP alarm and verify the configuration:

- set snmp alarm <trapName> [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-time <secs>] [-state (ENABLED | DISABLED)] [-severity <severity>] [-logging (ENABLED | DISABLED)]
- show snmp alarm <trapName>

Where,
ThresholdValue: Value for the high threshold. The NetScaler appliance generates an SNMP trap message when the value of the attribute associated with the alarm is greater than or equal to the specified high threshold value.

NormalValue. Value for the normal threshold. A trap message is generated if the value of the respective attribute falls to or below this value after exceeding the high threshold.

To configure SNMP alarms by using the configuration utility

Navigate to System > SNMP > Alarms, select an alarm and configure the alarm parameters.

Configuring SNMPv1 or SNMPv2 Traps

After configuring the alarms, you need to specify the trap listener to which the appliance sends the trap messages. Apart from specifying parameters such as IP or IPv6 address and the destination port of the trap listener, you can specify the type of trap (either generic or specific) and the SNMP version.

You can configure a maximum of 20 trap listeners for receiving either generic or specific traps.

You can also configure the appliance to send SNMP trap messages with a source IP address other than the NetScaler IP (NSIP or NSIP6) address to a particular trap listener. For a trap listener that has an IPv4 address, you can set the source IP to either a mapped IP (MIP) address or a subnet IP (SNIP) address configured on the appliance. For a trap listener that has an IPv6 address, you can set the source IP to subnet IPv6 (SNIP6) address configured on the appliance.

You can also configure the appliance to send trap messages to a trap listener on the basis of a severity level. For example, if you set the severity level as Minor for a trap listener, all trap messages of the severity level equal to or greater than Minor (Minor, Major, and Critical) are sent to the trap listener.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

If you have defined a community string for the trap listener, you must also specify a community string for each trap that is to be sent to the listener. A trap listener for which a community string has been defined accepts only trap messages that include a community string matching the community string defined in the trap listener. Other trap messages are dropped.

To add an SNMP trap by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp trap <trapClass> <trapDestination> -version (V1 | V2) -destPort <port> -communityName <string> -srcIP <ip_addr> -severity <severity>
- show snmp trap

Example
> add snmp trap specific 10.102.29.3 -version V2 -destPort 80 -communityName com1 -severity Major

To configure SNMP Traps by using the configuration utility

Navigate to System > SNMP > Traps, and create the SNMP trap.

Configuring SNMPv3 Traps

SNMPv3 provides security capabilities such as authentication and encryption by using the credentials of SNMP users. An SNMP manager can receive SNMPv3 trap messages only if its configuration includes the password assigned to the SNMP user.

The trap destination can now receive SNMPv1, SNMPv2, and SNMPv3 trap messages.

To configure an SNMPv3 trap by using the command line interface

At the command prompt, do the following:

1. Add an SNMPv3 trap.
   add snmp trap <trapClass> <trapDestination> -version (V1 | V2 | V3) -destPort <port> -communityName <string> -srcIP <ip_addr> -severity <severity>

   Note: Once set, the SNMP trap version cannot not be modified.
   Example
   > add snmp trap specific 10.102.29.3 -version V3 -destPort 80 -communityName com1 -severity Major
2. Add an SNMP user.
   add snmp user <name> -group <string> [-authType (MD5 | SHA) { -authPasswd } [-privType (DES | AES) { -privPasswd }]]

   Example
   > add snmp user edocs_user -group edocs_group
3. Bind the SNMPv3 trap to the SNMP user.
   bind snmp trap <trapClass> <trapDestination> [-version <version>] (-userName <string> [-securityLevel <securityLevel>])

   Example
   > bind snmp trap specific 10.102.29.3 -version V3 -userName edocs_user -securityLevel authPriv

To configure an SNMPv3 trap by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Add an SNMPv3 trap.
   Navigate to System > SNMP > Traps, and create the SNMP trap by selecting V3 as the SNMP version.

2. Add an SNMP user.
   Navigate to System > SNMP > Users, and create the SNMP user.

3. Bind the SNMPv3 trap to the SNMP user.
   - Navigate to System > SNMP > Traps, and select the SNMP version 3 trap.
   - Select the user to which the trap should be bound and define the appropriate Security Level.

Enabling Unconditional SNMP Trap Logging

By default, the NetScaler appliance logs any SNMP trap messages (for SNMP alarms in which logging is enabled) when at least one trap listener is specified on the appliance. However, you can specify that SNMP trap messages be logged even when no trap listeners are configured.

To enable unconditional SNMP trap logging by using the command line interface

At the command prompt, type the following commands to configure unconditional SNMP trap logging and verify the configuration:

- set snmp option -snmpTrapLogging (ENABLED | DISABLED)
- show snmp option

To enable unconditional SNMP trap logging by using the configuration utility

Navigate to System > SNMP, click Change SNMP Options and select SNMP Trap Logging.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NetScaler for SNMP v1 and v2 Queries

Jun 01, 2015

You can query the NetScaler SNMP agent for system-specific information from a remote device called *SNMP managers*. The agent then searches the management information base (MIB) for the data requested and sends the data to the SNMP manager.

The following types of SNMP v1 and v2 queries are supported by the SNMP agent:

- GET
- GET NEXT
- ALL
- GET BULK

You can create strings called community strings and associate each of these to query types. You can associate one or more community strings to each query type. Community string are passwords and used to authenticate SNMP queries from SNMP managers.

For example, if you associate two community strings, such as **abc** and **bcd**, to the query type GET NEXT, the SNMP agent on the NetScaler appliance considers only those GET NEXT SNMP query packets that contain **abc** or **bcd** as the community string.

This document includes the following details:

- Specifying an SNMP Manager
- Specifying an SNMP Community

Specifying an SNMP Manager

Updated: 2014-08-06

You must configure the NetScaler appliance to allow the appropriate SNMP managers to query it. You must also provide the SNMP manager with the required NetScaler-specific information. You can add up to a maximum of 100 SNMP managers or networks.

For an IPv4 SNMP manager you can specify a host name instead of the manager's IP address. If you do so, you must add a DNS name server that resolves the host name of the SNMP manager to its IP address. You can add up to a maximum of five host-name based SNMP managers.

Note: The appliance does not support use of host names for SNMP managers that have IPv6 addresses. You must specify the IPv6 address.
If you do not configure at least one SNMP manager, the appliance accepts and responds to SNMP queries from all IP addresses on the network. If you configure one or more SNMP managers, the appliance accepts and responds only to SNMP queries from those specific IP addresses.

If you remove an SNMP manager from the configuration, that manager can no longer query the appliance.

To add SNMP managers by specifying IP addresses by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp manager <IPAddress> ... [-netmask <netmask>]
- show snmp manager

Example

```
> add snmp manager 10.102.29.10 10.102.29.15 10.102.29.30
```

To add an SNMP manager by specifying its host name by using the command line interface

Important: If you specify the SNMP manager's host name instead of its IP address, you must configure a DNS name server to resolve the host name to the SNMP manager's IP address. For more information, see "Adding a Name Server."
At the command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp manager <IPAddress> [-domainResolveRetry **<integer>]**
- show snmp manager

Example

```
> add nameserver 10.103.128.15

> add snmp manager engwiki.eng.example.net –domainResolveRetry 10
```

To add an SNMP manager by using the configuration utility

1. Navigate to System > SNMP > Managers, and create the SNMP manager.
   Important: If you specify the SNMP manager's host name instead of its IPv4 address, you must configure a DNS name server to resolve the host name to the SNMP manager's IP address. For more information, see "Adding a Name Server."
   Note: The appliance does not support host names for SNMP managers that have IPv6 addresses.

Specifying an SNMP Community

You can create strings called community strings and associate them with the following SNMP query types on the appliance:

- GET
- GET NEXT
- ALL
- GET BULK

You can associate one or more community strings to each query types. For example, when you associate two community strings, such as **abc** and **bcd**, to the query type GET NEXT, the SNMP agent on the appliance considers only those GET NEXT SNMP query packets that contain **abc** or **bcd** as the community string.

If you do not associate any community string to a query type then the SNMP agent responds to all SNMP queries of that type.

To specify an SNMP community by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp community <communityName> <permissions>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- show snmp community

Example

```
> add snmp community com all
```

To configure an SNMP community string by using the configuration utility

Navigate to System > SNMP > Community, and create the SNMP community.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NetScaler for SNMPv3 Queries

Oct 09, 2016

Simple Network Management Protocol Version 3 (SNMPv3) is based on the basic structure and architecture of SNMPv1 and SNMPv2. However, SNMPv3 enhances the basic architecture to incorporate administration and security capabilities, such as authentication, access control, data integrity check, data origin verification, message timeliness check, and data confidentiality.

To implement message level security and access control, SNMPv3 introduces the user-based security model (USM) and the view-based access control model (VACM).

- **User-Based Security Model.** The user-based security model (USM) provides message-level security. It enables you to configure users and security parameters for the SNMP agent and the SNMP manager. USM offers the following features:
  - **Data integrity:** To protect messages from being modified during transmission through the network.
  - **Data origin verification:** To authenticate the user who sent the message request.
  - **Message timeliness:** To protect against message delays or replays.
  - **Data confidentiality:** To protect the content of messages from being disclosed to unauthorized entities or individuals.
- **View-Based Access Control Model.** The view-based access control model (VACM) enables you to configure access rights to a specific subtree of the MIB based on various parameters, such as security level, security model, user name, and view type. It enables you to configure agents to provide different levels of access to the MIB to different managers.

The Citrix NetScaler supports the following entities that enable you to implement the security features of SNMPv3:

- SNMP Engines
- SNMP Views
- SNMP Groups
- SNMP Users

These entities function together to implement the SNMPv3 security features. Views are created to allow access to subtrees of the MIB. Then, groups are created with the required security level and access to the defined views. Finally, users are created and assigned to the groups.

Note: The view, group, and user configuration are synchronized and propagated to the secondary node in a high availability (HA) pair. However, the engine ID is neither propagated nor synchronized as it is unique to each NetScaler appliance.
To implement message authentication and access control, you need to:

- Set the Engine ID
- Configure Views
- Configure Groups
- Configure Users

Setting the Engine ID

Updated: 2014-08-06

SNMP engines are service providers that reside in the SNMP agent. They provide services such as sending, receiving, and authenticating messages. SNMP engines are uniquely identified using engine IDs.

The NetScaler appliance has a unique engineID based on the MAC address of one of its interfaces. It is not necessary to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

override the engineID. However, if you want to change the engine ID, you can reset it.

To set the engine ID by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- set snmp engineId <engineID>
- show snmp engineId

Example

```
> set snmp engineId 8000173f0300c095f80c68
```

To set the engine ID by using configuration utility

Navigate to System > SNMP > Users, click Configure Engine ID and type an engine ID.

Configuring a View

Updated: 2014-08-06

SNMP views restrict user access to specific portions of the MIB. SNMP views are used to implement access control.

To add an SNMP view by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp view <name> <subtree> -type (included | excluded)
- show snmp view <name>

Where,

Name. Name for the SNMPv3 view. It can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the SNMPv3 view.

Subtree. A particular branch (subtree) of the MIB tree that you want to associate with this SNMPv3 view. You must specify the subtree as an SNMP OID. This is an argument of maximum Length: 99.

type. Include or exclude the subtree, specified by the subtree parameter, in or from this view. This setting can be useful when you have included a subtree, such as A, in an SNMPv3 view and you want to exclude a specific subtree of A, such as B, from the SNMPv3 view. This is a mandatory argument. Possible values: included, excluded.

To configure an SNMP view by using the configuration utility

Navigate to System > SNMP > Views, and create the SNMP view.

Configuring a Group

Updated: 2014-08-06

SNMP groups are logical aggregations of SNMP users. They are used to implement access control and to define the security levels. You can configure an SNMP group to set access rights for users assigned to that group, thereby restricting the users to specific views.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You need to configure an SNMP group to set access rights for users assigned to that group.

To add an SNMP group by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp group <name> <securityLevel> -readViewName <string>
- show snmp group <name> <securityLevel>

Where,

Name. Name for the SNMPv3 group. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the SNMPv3 group.

securityLevel. Security level required for communication between the NetScaler appliance and the SNMPv3 users who belong to the group. Specify one of the following options:

noAuthNoPriv. Require neither authentication nor encryption.

authNoPriv. Require authentication but no encryption.

authPriv. Require authentication and encryption. Note: If you specify authentication, you must specify an encryption algorithm when you assign an SNMPv3 user to the group. If you also specify encryption, you must assign both an authentication and an encryption algorithm for each group member. This is a mandatory argument. Possible values: noAuthNoPriv, authNoPriv, authPriv.

readViewName. Name of the configured SNMPv3 view that you want to bind to this SNMPv3 group. An SNMPv3 user bound to this group can access the subtrees that are bound to this SNMPv3 view as type INCLUDED, but cannot access the ones that are type EXCLUDED. If the NetScaler appliance has multiple SNMPv3 view entries with the same name, all such entries are associated with the SNMPv3 group. This is a mandatory argument. Maximum Length: 31

To configure an SNMP group by using the configuration utility

Navigate to System > SNMP > Groups, and create the SNMP group.

Configuring a User

Updated: 2014-08-06

SNMP users are the SNMP managers that the agents allow to access the MIBs. Each SNMP user is assigned to an SNMP group.

You need to configure users at the agent and assign each user to a group.

To configure a user by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add snmp user <name> -group <string> [-authType (MD5 | SHA) {-authPasswd } [-privType (DES | AES) {-privPasswd }]]
- show snmp user <name>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authType is the authentication option available while configuring an user. There are two authentication types such as MD5 and SHA.

privType is the encryption option available while configuring an user. There are two types of encryption such as DES of key size 128 bit, and AES of key size 128 bit.

Example

```
> add snmp user edocs_user -group edocs_group
```

To configure an SNMP user by using the configuration utility

Navigate to System > SNMP > Users, and create the SNMP user.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SNMP Alarms for Rate Limiting

Aug 24, 2016

Citrix NetScaler appliances such as the NetScaler MPX 10500, 12500, and 15500 are rate limited. The maximum throughput (Mbps) and packets per second (PPS) are determined by the license purchased for the appliance. For rate-limited platforms, you can configure SNMP traps to send notifications when throughput and PPS approach their limits and when they return to normal.

Throughput and PPS are monitored every seven seconds. You can configure traps with high-threshold and normal-threshold values, which are expressed as a percentage of the licensed limits. The appliance then generates a trap when throughput or PPS exceeds the high threshold, and a second trap when the monitored parameter falls to the normal threshold. In addition to sending the traps to the configured destination device, the NetScaler logs the events associated with the traps in the /var/log/ns.log file as EVENT ALERTSTARTED and EVENT ALERTENDED.

Exceeding the throughput limit can result in packet loss. You can configure SNMP alarms to report packet loss.

For more information about SNMP alarms and traps, see "Configuring the NetScaler to generate SNMP v1 and v2 Traps."

This document includes the following details:

- Configuring an SNMP Alarm for Throughput or PPS
- Configuring SNMP Alarm for Dropped Packets

Configuring an SNMP Alarm for Throughput or PPS

Updated: 2014-08-12

To monitor both throughout and PPS, you must configure seperate alarms and set threshold pps value in Mbps.

To configure an SNMP alarm for the throughput rate by using the command line interface

At the command prompt, type the following commands to configure the SNMP alarm and verify the configuration:

- set snmp alarm PF-RL-RATE-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state (**ENABLED** | **DISABLED**)] [-severity <severity>] [-logging (**ENABLED** | **DISABLED**)]
- show snmp alarm PF-RL-RATE-THRESHOLD

Example

```
> set snmp alarm PF-RL-RATE-THRESHOLD -thresholdValue 70 -normalValue 50
```

To configure an SNMP alarm for PPS by using the command line interface

At the command prompt, type the following commands to configure the SNMP alarm for PPS and verify the configuration:

- set snmp alarm PF-RL-PPS-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state (**ENABLED** | **DISABLED**)] [-severity <severity>] [-logging (**ENABLED** | **DISABLED**)]
- show snmp alarm PF-RL-PPS-THRESHOLD

Example

```
> set snmp alarm PF-RL-PPS-THRESHOLD -thresholdValue 70 -normalValue 50
```

To configure an SNMP alarm for throughput or PPS by using the configuration utility

1. Navigate to System > SNMP > Alarms, and select PF-RL-RATE-THRESHOLD (for throughput rate) or PF-RL-PPS-THRESHOLD (for packets per second).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Set the alarm parameters and enable the selected SNMP alarm.

Configuring SNMP Alarm for Dropped Packets

Updated: 2014-08-12

You can configure an alarm for packets dropped as a result of exceeding the throughput limit and an alarm for packets dropped as a result of exceeding the PPS limit.

To configure an SNMP alarm for packets dropped because of excessive throughput, by using the command line interface

At the command prompt, type:
set snmp alarm PF-RL-RATE-PKTS-DROPPED [-state (**ENABLED** | **DISABLED**)] [-severity <severity>] [-logging (**ENABLED** | **DISABLED**)]

To configure an SNMP alarm for packets dropped because of excessive PPS, by using the command line interface

At the command prompt, type:
set snmp alarm PF-RL-PPS-PKTS-DROPPED [-state (**ENABLED** | **DISABLED**)] [-severity <severity>] [-logging (**ENABLED** | **DISABLED**)]

To configure an SNMP alarm for dropped packets by using the configuration utility

1. Navigate to System > SNMP > Alarms, and select PF-RL-RATE-PKTS-DROPPED (for packets dropped because of excessive throughput) or PF-RL-PPS-PKTS-DROPPED (for packets dropped because of excessive PPS).
2. Set the alarm parameters and enable the selected SNMP alarm.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Audit Logging

Jun 03, 2015

Auditing is a methodical examination or review of a condition or situation. The Audit Logging feature enables you to log the NetScaler states and status information collected by various modules in the kernel and in the user-level daemons. For audit logging, you can use the SYSLOG protocol, the native NSLOG protocol, or both.

SYSLOG is a standard protocol for logging. It has two components: the SYSLOG auditing module, which runs on the NetScaler appliance, and the SYSLOG server, which can run on the underlying FreeBSD operating system (OS) of the NetScaler appliance or on a remote system. SYSLOG uses user data protocol (UDP) for data transfer.

Similarly, the native NSLOG protocol has two components— the NSLOG auditing module, which runs on the NetScaler appliance, and the NSLOG server, which can run on the underlying FreeBSD OS of the NetScaler appliance or on a remote system. NSLOG uses transmission control protocol (TCP) for data transfer.

When you run a SYSLOG or NSLOG server, it connects to the NetScaler appliance. The NetScaler appliance then starts sending all the log information to the SYSLOG or NSLOG server, and the server can filter the log entries before storing them in a log file. An NSLOG or SYSLOG server can receive log information from more than one NetScaler appliance, and a NetScaler appliance can send log information to more than one SYSLOG server or NSLOG server.

If multiple SYSLOG servers are configured, the NetScaler appliance send its SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the NetScaler appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastPackets, and AuditlogHash.

The log information that a SYSLOG or NSLOG server collects from a NetScaler appliance is stored in a log file in the form of messages. These messages typically contain the following information:

- The IP address of a NetScaler appliance that generated the log message
- A time stamp
- The message type
- The predefined log levels (Critical, Error, Notice, Warning, Informational, Debug, Alert, and Emergency)
- The message information

To configure audit logging, you first configure the audit modules on the NetScaler appliance. That involves creating audit policies and specifying the NSLOG server or SYSLOG server information. You then install and configure the SYSLOG or the NSLOG server on the underlying FreeBSD OS of the NetScaler appliance or on a remote system.

Note: Because SYSLOG is an industry standard for logging program messages, and various vendors provide support, this documentation does not include SYSLOG server configuration information.
The NSLOG server has its own configuration file (auditlog.conf). You can customize logging on the NSLOG server system by making additional modifications to the configuration file (auditlog.conf).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NetScaler Appliance for Audit Logging

Oct 26, 2017

On the NetScaler appliance, you configure SYSLOG and/or NSLOG policies. Each policy includes a rule, which is an expression identifying the messages to be logged, and a SYSLOG or NSLOG (depending on the type of policy) action. The action specifies the server to which to send the log message, the level of the messages to be logged, and the data format of the logged messages. You can bind the policies globally or to individual virtual servers.

The appliance logs the following information related to TCP connections:

- Source port
- Destination port
- Source IP
- Destination IP
- Number of bytes transmitted and received
- Time period for which the connection is open

Note:

- You can enable TCP logging on individual load balancing virtual servers. You must bind the audit log policy to a specific load balancing virtual server that you want to log.
- When using the NetScaler as the audit log server, by default, the ns.log file is rotated (new file is created) when the file size reaches 100K and the last 25 copies of the ns.log are archived and compressed with gzip. To accommodate more archived files after 25 files, the oldest archive is deleted. You can modify the 100K limit or the 25 file limit by updating the following entry in the /etc/newsyslog.conf file:
  /var/log/ns.log 600 25 100 * Z
  where, 25 is the number of archived files to be maintained and 100K is the size of the ns.log file after which the file will be archived.

This document includes the following details:

- Configuring SYSLOG and NSLOG Actions
- Configuring Audit Policies
- Binding the Audit Policies Globally
- Configuring Policy-Based Logging

Configuring SYSLOG and NSLOG Actions

Updated: 2015-06-03

You can configure audit server actions for different servers and for different log levels.

By default SYSLOG uses UDP, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data. When using TCP for SYSLOG, you can set the buffer limit on the NetScaler appliance to store the logs. After which the logs are sent to the SYSLOG server.

To configure a SYSLOG server action by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> [-dateFormat (MMDDYYYY |

© 1999-2017 Citrix Systems, Inc. All rights reserved.

DDMMYYYY)] [-transport (TCP | UDP)]
- show audit syslogAction [<name>]

Example

```
> add audit syslogaction audit-action1 10.102.1.1 -loglevel INFORMATIONAL -dateformat MMDDYYYY
```

To configure an NSLOG server action by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add audit nslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> [-dateFormat (MMDDYYYY | DDMMYYYY)]
- show audit nslogAction [<name>]

Example

```
> add audit nslogAction nslog-action1 10.102.1.3 -serverport 520 -loglevel INFORMATIONAL -dateFormat MMDDYYYY
```

To configure an auditing server action by using the configuration utility

Navigate to System > Auditing > Syslog or Nslog, click Servers tab and create the auditing server.

Specifying a domain name for a logging server

When configuring an auditlog action, you can specify the domain name of a syslog or nslog server instead of its IP address. Then, if the server's IP address changes, you don't have to change it on the NetScaler appliance.

The domain name resolution is done through a DNS name server that resolves the host name of the syslog or nslog server to its IP address.

Specifying a domain name for a logging server consists of the following tasks:

- Set the domain server name. Set the domain name parameter while adding or modifying the syslog/nslog action as a part of syslog/nslog configuration. For instructions on configuring SYSLOG or NSLOG logging on a NetScaler appliance, see http://docs.citrix.com/en-us/netscaler/11/system/audit-logging/configuring-audit-logging.html.
- Add a name server: Specify a DNS name server to resolve the host name to the syslog server's IP address. For more information, see http://docs.citrix.com/en-us/netscaler/10-1/ns-tmg-wrapper-10-con/ns-tmg-dns-gen-wrapper-10-con/ns-tmg-dns-config-ns-forwrdr-con/ns-tmg-dns-add-name-srvr-tsk.html.

  Note: If you would like to use another Netscaler as the DNS server, configure the following in that Netscaler

1. add service service1 10.102.146.48 adns 53
2. add addrec bsdserver.com 10.102.146.4

Here service1 is the DNS service which is to be configured and this service will resolve the bsdserver.com to 10.102.146.4 when queried.

To configure the auditlog action to accept the domain name of a syslog server by using the command line

- For a new syslog configuration, at the command prompt, type:
  - add audit syslogAction <name> (<serverIP> | ((<serverDomainName> [-domainResolveRetry <integer>]) | -lbVserverName <string>)) -logLevel <logLevel>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- For an existing syslog configuration, at the command prompt, type:
    - set audit syslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>] [-serverDomainName <string>] [-lbVserverName <string>] [-domainResolveRetry <integer>] [-domainResolveNow]

To configure the auditlog action to accept the domain name of an nslog server by using the command line

- For a new nslog configuration, at the command prompt, type:
    - add audit nslogAction <name> (<serverIP> | (<serverDomainName> [-domainResolveRetry <integer>])) -logLevel <logLevel> ...
- For an existing nslog configuration, at the command prompt, type:
    - set audit nslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>] [-serverDomainName <string>] [-domainResolveRetry <integer>] [-domainResolveNow]

Parameter Description

serverDomainName

Instead of an IP address, you can specify a host name that has been assigned to an logging server. If you do so, you must add a DNS name server that resolves the host name of the logging server to its IP address.

domainResolveRetry

Amount of time, in seconds, for which the NetScaler appliance waits before sending another DNS query to resolve the host name of the logging server if the last query failed. This parameter is valid for host-name based logging servers only.

Minimum value: 5

Maximum value: 20939

Sample Configuration

Syslog:

In the following sample configuration, syslog action (SYS-ACTION-1) and syslog policy (SYSLOG-POLICY-1) are configured. SYS-ACTION-1 specifies the domain name (EXAMPLE.DOMAIN.NAME.SYSLOG) of a syslog server having IP address, 192.0.2.10.

The NetScaler sends DNS queries to the DNS server having IP address 192.0.2.20 to resolve the domain name (EXAMPLE.DOMAIN.NAME.SYSLOG) of the syslog server to its ip address (192.0.2.10)

```
> add audit syslogAction SYS-ACTION-1 EXAMPLE.DOMAIN.NAME.SYSLOG – domainResolveRetry 500
Done

> add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1

Done

>bind system global SYSLOG-POLICY-1

Done
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>add nameserver 192.0.2.20

Done
```

Nslog:

In the following sample configuration, nslog action (NSLOG-ACTION-1) and nslog policy (NSLOG-POLICY-1) are configured. NSLOG-ACTION-1 specifies the domain name (EXAMPLE.DOMAIN.NAME.NSLOG) of a nslog server having IP address, 192.0.2.10.The NetScaler sends DNS queries to the DNS server having IP address 192.0.2.20 to resolve the domain name of the nslog server.

```
>add audit nslogAction NSLOG-ACTION-1  EXAMPLE.DOMAIN.NAME.NSLOG – domainResolveRetry 500

Done

> add audit nslogPolicy NSLOG-POLICY-1 ns_true NSLOG-ACTION-1

Done

> bind system global NSLOG-POLICY-1

Done

>add nameserver 192.0.2.20

Done
```

Configuring Audit Policies

Updated: 2015-04-29

Configure SYSLOG policies to log messages to a SYSLOG server, and/or NSLOG policy to log messages to an NSLOG server. Each policy includes a rule identifying the messages to be logged, and a SYSLOG or NS LOG (depending on the type of policy) action.

To configure a SYSLOG policy by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add audit syslogPolicy <name> <rule> <action>
- show audit syslogPolicy [<name>]

Example

```
> add audit syslogpolicy syslog-pol1 ns_true audit-action1
```

To configure an NSLOG policy by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add audit nslogPolicy <name> <rule> <action>
- show audit nslogPolicy [<name>]

Example

```
> add audit nslogPolicy nslog-pol1 ns_true nslog-action1
```

To configure an audit server policy by using the configuration utility

Navigate to System > Auditing > Syslog or Nslog, click Policies tab and create the auditing policy.

Binding the Audit Policies Globally

Updated: 2015-06-02

You must globally bind the audit log policies to their respective global entities (SYSTEM, RNAT, VPN) to enable logging of all NetScaler system events. By defining the priority level, you can set the evaluation order of the audit server logging. Priority 0 is the highest and is evaluated first. The higher the priority number, the lower is the priority of evaluation.

To configure a SYSLOG policy by using the command line interface

At the command prompt, type:

- bind system global [<policyName> [-priority <positive_integer>]]
- show system global

Example

```
> bind system global nslog-pol1 -priority 20
```

To globally bind the audit policy by using the configuration utility

1. Navigate to System > Auditing > Syslog or Nslog.
2. On Policies tab, click Action, and select Global Bindings to bind the audit global policies.

Configuring Policy-Based Logging

Updated: 2015-06-26

The audit logging feature dynamically captures audit logs in a user-defined format for rewrite and responder policies. The audit server (SYSLOG or NSLOG) collects and stores the logs in chronological order on the NetScaler appliance. Many enterprises use these logs to tune feature behavior and troubleshoot problems.

The NetScaler audit logging feature provides detailed control of the information to be logged and specifies the location where the log messages are stored. Each log message can have one of the severity levels described in the following table.

Log Level	Indicates
EMERGENCY	Critical problem that might make the system unusable.
ALERT	Noncritical problem that might cause the NetScaler appliance to function incorrectly. You must take immediate corrective action to prevent the appliance from experiencing a critical problem.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Log Level	Indicates
CRITICAL	Critical condition that does not restrict operation but might escalate to a larger problem.
ERROR	Failure of a NetScaler operation.
WARNING	Issue that might result in critical errors.
NOTICE	Same as INFORMATION, but in greater detail.
INFORMATION	Any action taken by the NetScaler appliance. This level of logging can help troubleshoot problems on the Citrix NetScaler product line.
DEBUG	Extensive, detailed information. Developers use this level of logging to troubleshoot problems.

Configuring Audit Logging for Rewrite and Responder Policies

To process logs for rewrite and responder policies, do the following:

1. Create a SYSLOG action and associate the action with SYSLOG policy. A SYSLOG action specifies the configuration of SYSLOG server. You can define the server IP, port and the kind of logs to be sent to the SYSLOG server.
2. Create SYSLOG policy. The SYSLOG policy includes a rule, which is an expression identifying the messages to be logged, depending on the type of policy action.
3. Bind the policy to system global. By binding to system global, the logs generated at the system level will be sent to the respective syslog servers as defined in their syslog action.
4. Add an audit message action. The audit message action specifies the log format, the level of the messages to be logged, and the data format of the logged messages.
5. Create a rewrite or responder policy and associate the audit message action.
6. Bind the rewrite or responder policy globally or to a load balancing or content switching virtual server so that the policy applies to all the HTTP traffic specific to that virtual server.

To configure policy based logging by using the command line interface

At the command prompt, do the following:

1. Create a syslog action.
   add audit syslogAction <name> <serverIP> -logLevel <logLevel> [-userDefinedAuditlog (YES | NO)]

   Example:

   add audit syslogAction LogServerAction 10.102.115.34 -logLevel ALL -userDefinedAuditlog YES

   Note: The parameter “-userDefinedAuditlog” should be set to “YES” to capture audit logs in a user-defined format.
2. Add a syslog policy
   add audit syslogPolicy <name> <rule> <action>

   Example:

   add audit syslogPolicy customlog-log-pol ns_true syslogAction

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. Bind the syslog policy to system global.
   bind system global <policy name> [-priority <positive_integer>]

   Example:

   bind system global customlog-log-pol -priority 100

After configuring the SYSLOG settings, do the following to generate the user-defined logs:

1. Add an audit message action
   Note: Currently, NetScaler appliance generates logs for packets belonging to a specific virtual server.
   add audit messageaction <name> <logLevel> <stringBuilderExpr> [-logtoNewnslog (YES | NO)] [-bypassSafetyCheck (YES | NO)]

   Example:

   add audit messageaction audit_log_action <LogLevel> "\"alertType=NetScaler Log - clientIP=\" + CLIENT.IP.SRC" -logtoNewnslog [YES|NO] -bypassSafetyCheck YES

   Where

   <logLevel> = (EMERGENCY | ALERT | CRITICAL | ERROR | WARNING | NOTICE | INFORMATIONAL | DEBUG)

2. Add a rewrite or responder policy
   add rewrite policy <name> <rule> <action> [-logAction <string>]

   add responder policy <name> <rule> <action> [-logAction <string>]

   Example:

   add rewrite policy log-rewr-pol true NOREWRITE -logAction audit_log_action

   add responder policy log-resp-pol true NOOP -logAction audit_log_action

   Note: By default, the policy action is set as "NO ACTION" so that no action is performed to the packet but to generate log for the packet.
   Use "logAction" policy parameter to link the audit message action to the policy.

3. Bind the responder or rewrite policy to a virtual server
   bind lb vserver <name> (-policyName <string>@ [-priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type (REQUEST | RESPONSE)]

   Example:

   bind lb vserver <VSERVER NAME> -policyName log-rewr-pol -priority 5 -gotoPriorityExpression END -type REQUEST

Configuring policy based logging using the configuration utility

1. Add a syslog action
   Navigate to **System**>**Auditing**>**Syslog**, click **Servers**tab and create the auditing server.

2. Add a syslog policy
   Navigate to **System**>**Auditing**>**Syslog**, click **Policies**tab and create the auditing server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. Bind the syslog policy to system global
   1. Navigate to **System** > **Auditing** > **Syslog**.
   2. On **Policies** tab, click **Action**, and select **Global Bindings** to bind the audit global policies.

After configuring the SYSLOG configurations, do the following to generate the user-defined logs:

1. Add an audit message action
   Navigate to **System**>**Auditing**>**Message Actions**, and create the audit message action.

2. Add a rewrite or responder policy
   For rewrite policy.Navigate to **AppExpert** > **Rewrite** > **Policies**, and create a rewrite or a responder policy.

   For responder policy. Navigate to **AppExpert** > **Responder** > **Policies**, and create a rewrite or a responder policy.

   Note: When creating the policy, specify the audit message action in the **Log Action** field.
3. Bind the responder or rewrite policy to a virtual server
   1. For load balancing. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, select the required virtual server and click **Policies** to bind the relevant policy.
   2. For content switching. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers**, select the required virtual server and click **Policies** to bind the relevant policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing and Configuring the NSLOG Server

Jul 23, 2015

During installation, the NSLOG server executable file (auditserver) is installed along with other files. The auditserver executable file includes options for performing several actions on the NSLOG server, including running and stopping the NSLOG server. In addition, you use the auditserver executable to configure the NSLOG server with the IP addresses of the NetScaler appliances from which the NSLOG server will start collecting logs. Configuration settings are applied in the NSLOG server configuration file (auditlog.conf).

Then, you start the NSLOG server by executing the auditserver executable. The NSLOG server configuration is based on the settings in the configuration file. You can further customize logging on the NSLOG server system by making additional modifications to the NSLOG server configuration file (auditlog.conf).

Attention: The version of the NSLOG server package must be the same as that of the NetScaler. For example, if the version of the NetScaler is 10.1 Build 125.9, the NSLOG server must also be of the same version.
The following table lists the operating systems on which the NSLOG server is supported.

Table 1. Supported Platforms for the NSLOG Server

Operating system	Software requirements	Remarks
Windows	• Windows XP Professional • Windows Server 2003 • Windows 2000/NT • Windows Server 2008 • Windows Server 2008 R2	
Linux	• RedHat Linux 4 or later • SUSE Linux Enterprise 9.3 or later	
FreeBSD	FreeBSD 6.3 or later	For NetScaler 10.5, use only FreeBSD 8.4.
Mac OS	Mac OS 8.6 or later	Not supported on NetScaler 10.1 and later releases.

The minimum hardware specifications for the platform running the NSLOG server are as follows:

- Processor- Intel x86 ~501 megahertz (MHz)
- RAM - 512 megabytes (MB)
- Controller - SCSI

This document includes the following details:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Installing NSLOG Server on the Linux Operating System
- Installing NSLOG Server on the FreeBSD Operating System
- Installing NSLOG Server Files on the Windows Operating System
- NSLOG Server Command Options
- Adding the NetScaler Appliance IP Addresses on the NSLOG Server
- Verifying the NSLOG Server Configuration File

Installing NSLOG Server on the Linux Operating System

Log on to the Linux system as an administrator. Use the following procedure to install the NSLOG server executable files on the system.

To install the NSLOG server package on a Linux operating system

1. At a Linux command prompt, type the following command to copy the NSauditserver.rpm file to a temporary directory:
   cp <path_to_cd>/Utilities/auditserver/Linux/NSauditserver.rpm /tmp

2. Type the following command to install the NSauditserver.rpm file:
   rpm -i NSauditserver.rpm

   This command extracts the files and installs them in the following directories:

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

To uninstall the NSLOG server package on a Linux operating system

1. At a command prompt, type the following command to uninstall the audit server logging feature:
   rpm -e NSauditserver

2. For more information about the NSauditserver RPM file, use the following command:
   rpm -qpi *.rpm

3. To view the installed audit server files use the following command:
   rpm -qpl *.rpm

   *.rpm: Specifies the file name.

Installing NSLOG Server on the FreeBSD Operating System

Before you can install the NSLOG server, you have to copy the NSLOG package from the NetScaler product CD or download it from www.citrix.com. The NSLOG package has the following name format:

AuditServer_<release number>-<build number>.zip

For example: AuditServer_10.5-58.11.zip

This package contains files for all supported platforms: Linux, Windows, and FreeBSD. On a FreeBSD operating system, install the NSLOG package that has the following name format:

audserver_bsd-<release number>-<build number>.tgz

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

For example: audserver_bsd-10.5-58.11.tgz

To download NSLOG package from www.citrix.com

1. In a web browser, go to www.citrix.com.
2. In the menu bar, click **Log In**.
3. Enter your login credentials, and then click **Log In**.
4. In the menu bar, click **Downloads**.
5. From the **Select a product** list, select **NetScaler ADC**.
6. On the **NetScaler ADC** page, select the release for which you want to download the NSLOG package (for example, Release 10.5), and then select **Firmware**.
7. Under **Firmware**, select the NetScaler firmware for the build number for which you want to download the NSLOG package.
8. On the page that appears, scroll down, select **Audit Servers**, and click **Download File** next to the package that you want to download.

To install the NSLOG server package on a FreeBSD operating system

1. On the system to which you have downloaded the NSLOG package AuditServer_<release number>-<build number>.zip (for example, AuditServer_9.3-51.5.zip), extract the FreeBSD NSLOG server package audserver_bsd-<release number>-<build number>.tgz (for example, audserver_bsd-9.3-51.5.tgz) from the package.
2. Copy the FreeBSD NSLOG server package audserver_bsd-<release number>-<build number>.tgz (for example, audserver_bsd-9.3-51.5.tgz) to a directory on a system running FreeBSD OS.
3. At a command prompt for the directory into which the FreeBSD NSLOG server package was copied, run the following command to install the package:
   pkg_add audserver_bsd-<release number>-<build number>.tgz

   Example
   pkg_add audserver_bsd-9.3-51.5.tgz
   The following directories are extracted:

   - <root directory extracted from the FreeBSD NSLOG server package tgz file>\netscaler\bin (for example, /var/auditserver/netscaler/bin)
   - <root directory extracted from the FreeBSD NSLOG server package tgz file>\netscaler\etc (for example, /var/auditserver/netscaler/etc)
   - <root directory extracted from the FreeBSD NSLOG server package tgz file>\netscaler\samples (for example, /var/auditserver/samples)
4. At a command prompt, type the following command to verify that the package is installed:
   pkg_info | grep NSaudserver

To uninstall the NSLOG server package on a FreeBSD operating system

At a command prompt, type:

pkg_delete NSaudserver

Installing NSLOG Server Files on the Windows Operating System

Before you can install the NSLOG server, you have to copy the NSLOG package from the NetScaler product CD or download it from www.citrix.com. The NSLOG package has the following name format AuditServer _<release number>-<build number>.zip (for example, AuditServer_9.3-51.5.zip). This package contains NSLOG installation packages for all

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

supported platforms.

To download NSLOG package from www.Citrix.com

1. In a web browser, go to www.citrix.com.
2. In the menu bar, click Log In.
3. Enter your login credentials, and then click Log In.
4. In the menu bar, click Downloads.
5. Search to find the page that provides the appropriate release number and build.
6. On that page, under Audit Servers, click Download to download the NSLOG package, having the format AuditServer_<release number>-<build number>.zip , to your local system (for example, AuditServer_9.3-51.5.zip).

To install NSLOG server on a Windows operating system

1. On the system, where you have downloaded the NSLOG package AuditServer_<release number>-<build number>.zip (for example, AuditServer_9.3-51.5.zip), extract audserver_win-<release number>-<build number>.zip (for example, audserver_win-9.3-51.5.zip) from the package.
2. Copy the extracted file audserver_<release number>-<build number>.zip (for example, audserver_win-9.3-51.5.zip) to a Windows system on which you want to install the NSLOG server.
3. Unzip the audserver_<release number>-<build number>.zip file (for example, audserver_win-9.3-51.5.zip).
4. The following directories are extracted:
   1. <root directory extracted from the Windows NSLOG server package zip file>\bin (for example, C:\audserver_win-9.3-51.5\bin)
   2. <root directory extracted from the Windows NSLOG server package zip file>\etc (for example, C:\audserver_win-9.3-51.5\ etc)
   3. < root directory extracted from the Windows NSLOG server package zip file >\samples (for example, C:\audserver_win-9.3-51.5\ samples)
5. At a command prompt, run the following command from the <root directory extracted from the Windows NSLOG server package zip file>\bin path:
   audserver -install -f <directorypath>**auditlog.conf**

   <directorypath>: Specifies the path to the configuration file (auditlog.conf). By default, log.conf is under <root directory extracted from Windows NSLOG server package zip file>\samples directory. But you can copy auditlog.conf to your desired directory.

To uninstall the NSLOG server on a Windows operating system

At a command prompt, run the following from the <root directory extracted from Windows NSLOG server package zip file>\bin path:

audserver -remove

NSLOG Server Command Options

The following table describes the commands that you can use to configure audit server options.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 2. Audit Server Options

Audit server commands	Specifies
audserver -help	The available Audit Server options.
audserver -addns -f <path to configuration file>	The system that gathers the log transaction data. You are prompted to enter the IP address of the NetScaler appliance. Enter the valid user name and password.
audserver -verify -f <path to configuration file>	Check for syntax or semantic errors in the configuration file (for example, auditlog.conf).
audserver -start -f <path to configuration file>	Start audit server logging based on the settings in the configuration file (auditlog.conf). Linux only: To start the audit server as a background process, type the ampersand sign (&) at the end of the command.
audserver -stop (Linux only)	Stops audit server logging when audit server is started as a background process. Alternatively, use the Ctrl+C key to stop audit server logging.
audserver -install -f <path to configuration file> (Windows only)	Installs the audit server logging client as a service on Windows.
audserver -startservice (Windows Only)	Start the audit server logging service, when you enter this command at a command prompt. You can also start audit server logging from Start > Control Panel > Services. Note: Audit server logging starts by using the configuration settings in the configuration file, for example, auditlog.conf file specified in the audit server install option.
audserver -stopservice (Windows Only)	Stop audit server logging.
audserver -remove	Removes the audit server logging service from the registry.

Run the audserver command from the directory in which the audit server executable is present:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- On Windows: \ns\bin
- On Solaris and Linux: \usr\local\netscaler\bin

The audit server configuration files are present in the following directories:

- On Windows: \ns\etc
- On Linux: \usr\local\netscaler\etc

The audit server executable is started as ./auditserver in Linux and FreeBSD.

Adding the NetScaler Appliance IP Addresses on the NSLOG Server

In the configuration file (auditlog.conf), add the IP addresses of the NetScaler appliances whose events must be logged.

To add the IP addresses of the NetScaler appliance

At a command prompt, type the following command:

audserver -addns -f <directorypath>\auditlog.conf

<directorypath>: Specifies the path to the configuration file (auditlog.conf).

You are prompted to enter the information for the following parameters:

NSIP: Specifies the IP address of the NetScaler appliance, for example, 10.102.29.1.

Userid: Specifies the user name, for example, nsroot.

Password: Specifies the password, for example, nsroot.

If you add multiple NetScaler IP addresses (NSIP), and later you do not want to log all of the NetScaler appliance event details, you can delete the NSIPs manually by removing the NSIP statement at the end of the auditlog.conf file. For a high availability (HA) setup, you must add both primary and secondary NetScaler IP addresses to auditlog.conf by using the audserver command. Before adding the IP address, make sure the user name and password exist on the system.

Verifying the NSLOG Server Configuration File

Check the configuration file (audit log.conf) for syntax correctness to enable logging to start and function correctly.

To verify configuration, at a command prompt, type the following command:

audserver -verify -f <directorypath>\auditlog.conf

<directorypath>: Specifies the path to the configuration file (audit log.conf).

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Running the NSLOG Server

Jun 20, 2013

To start audit server logging

Type the following command at a command prompt:

audserver -start -f <directorypath>\auditlog.conf
<directorypath>: Specifies the path to the configuration file (audit log.conf).

To stop audit server logging that starts as a background process in FreeBSD or Linux

Type the following command:

audserver -stop

To stop audit server logging that starts as a service in Windows

Type the following command:

audserver -stopservice

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Customizing Logging on the NSLOG Server

Jun 01, 2015

You can customize logging on the NSLOG server by making additional modifications to the NSLOG server configuration file (log.conf). Use a text editor to modify the log.conf configuration file on the server system.

To customize logging, use the configuration file to define filters and log properties.

- **Log filters.** Filter log information from a NetScaler appliance or a set of NetScaler appliances.
- **Log properties.** Each filter has an associated set of log properties. Log properties define how to store the filtered log information.

This document includes the following details:

- Creating Filters
- Specifying Log Properties

Creating Filters

Updated: 2013-11-14

You can use the default filter definition located in the configuration file (audit log.conf), or you can modify the filter or create a new filter. You can create more than one log filter.

Note: For consolidated logging, if a log transaction occurs for which there is no filter definition, the default filter is used (if it is enabled.) The only way you can configure consolidated logging of all the NetScaler appliances is by defining the default filter.

To create a filter

At the command prompt, type the following command in the configuration file (auditlog.conf) :

filter <filterName> [IP <ip>] [NETMASK <mask>] [ON | OFF]

<filterName>: Specify the name of the filter (maximum of 64 alphanumeric characters).

<ip>: Specify the IP addresses.

<mask>: Specify the subnet mask to be used on a subnet.

Specify ON to enable the filter to log transactions, or specify OFF to disable the filter. If no argument is specified, the filter is ON

Examples

filter F1 IP 192.168.100.151 ON

To apply the filter F2 to IP addresses 192.250.100.1 to 192.250.100.254:

filter F2 IP 192.250.100.0 NETMASK 255.255.255.0 ON

filterName is a required parameter if you are defining a filter with other optional parameters, such as IP address, or the combination of IP address and Netmask.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Specifying Log Properties

Updated: 2013-11-13

Log properties associated with the filter are applied to all the log entries present in the filter. The log property definition starts with the key word BEGIN and ends with END as illustrated in the following example:

```
BEGIN <filtername>
 logFilenameFormat ...
 logDirectory ...
 logInterval ...
 logFileSizeLimit ....
END
```

Entries in the definition can include the following:

- **LogFilenameFormat** specifies the file name format of the log file. The name of the file can be of the following types:
  - Static: A constant string that specifies the absolute path and the file name.
  - Dynamic: An expression that includes the following format specifiers:
    - Date (%{format}t)
    - % creates file name with NSIP

  Example

  LogFileNameFormat Ex%{%m%d%y}t.log

  This creates the first file name as Exmmddyy.log. New files are named: Exmmddyy.log.0, Exmmddyy.log.1, and so on. In the following example, the new files are crated when the file size reaches 100MB.

  Example

  LogInterval size
  LogFileSize 100
  LogFileNameFormat Ex%{%m%d%y}t

  Caution: The date format %t specified in the LogFilenameFormat parameter overrides the log interval property for that filter. To prevent a new file being created every day instead of when the specified log file size is reached, do not use %t in the LogFilenameFormat parameter.
- **logDirectory** specifies the directory name format of the log file. The name of the file can be either of the following:
  - Static: Is a constant string that specifies the absolute path and file name.
  - Dynamic: Is an expression containing the following format specifiers:
    - Date (%{format}t)
    - % creates directory with NSIP

  The directory separator depends on the operating system. In Windows, use the directory separator \.

  Example:

  LogDirectory dir1\dir2\dir3

  In the other operating systems (Linux, FreeBsd, etc.), use the directory separator /.

- **LogInterval** specifies the interval at which new log files are created. Use one of the following values:
  - Hourly: A file is created every hour. Default value.
  - Daily: A file is created very day at midnight.
  - Weekly: A file is created every Sunday at midnight.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Monthly : A file is created on the first day of the month at midnight.
- None: A file is created only once, when audit server logging starts.
- Size: A file is created only when the log file size limit is reached.

Example

LogInterval Hourly

- **LogFileSizeLimit** specifies the maximum size (in MB) of the log file. A new file is created when the limit is reached.

  Note that you can override the loginterval property by assigning size as its value.

  The default LogFileSizeLimit is 10 MB.

  Example

  LogFileSizeLimit 35

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Reliable SYSLOG

Sep 29, 2016

The NetScaler appliance supports sending SYSLOG messages to external log servers over TCP for a more reliable transport mechanism. Unlike the UDP protocol, TCP provides receipt acknowledgement, reliability, and flow control over.

To configure the NetScaler appliance to send SYSLOG messages over TCP using the command line interface

At the command prompt, type the following:

add audit syslogAction <name> <serverIP> [-serverPort <port>] [-logLevel <logLevel>] [-dateFormat (MMDDYYYY | DDMMYYYY)] [-transport (TCP)]

Example

add audit syslogaction audit-action1 10.102.1.1 -serverPort 514 -loglevel INFORMATIONAL -dateformat MMDDYYYY -transport TCP

Note: 514 is the standard server port number for UDP while there is no standard port for TCP. The appliance by default takes 514 as the standard server port number for both UDP and TCP protocols.

To configure the NetScaler appliance to send SYSLOG messages over TCP using the configuration utility

1. Navigate to **System** > **Auditing** > **Syslog**, and select the **Servers** tab.
2. Click **Add** and select **TCP Logging**.

Limitations

1. The log messages are sourced through the NetScaler appliance's NSIP address, and currently there is no SNIP support for SYSLOG.
2. The reliable SYSLOG option does not provide authentication, integrity check, or privacy.
3. The NetScaler appliance relies on the TCP protocol to provide confirmation for SYSLOG message delivery to external log servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing SYSLOG Servers

Oct 09, 2016

The NetScaler appliance send its SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the NetScaler appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastConnection, LeastPackets, and AuditlogHash.

Load balancing of SYSLOG servers using the command line interface

At the command prompt, type:

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.
   add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP | SYSLOGUDP)> <port>

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGUDP, and load balancing method as AUDITLOGHASH.
   add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod <AUDITLOGHASH>]

3. Bing the service to the load balancing virtual server.
   Bind lb vserver <name> <serviceName>

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.
   add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel <logLevel>]

5. Add a SYSLOG policy by specifying the rule and action.
   add syslogpolicy <name> <rule> <action>

6. Bind the SYSLOG policy to the system global for the policy to take effect.
   bind system global <policyName>

Load balancing of SYSLOG servers using the configuration utility

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.
   Navigate to **Traffic Management** > **Services**, click **Add** and select **SYLOGTCP** or **SYSLOGUDP** as protocol.

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.
   Navigate to **Traffic Management** > **Virtual Servers**, click **Add** and select **SYLOGTCP** or **SYSLOGUDP** as protocol.

3. Bing the service to the load balancing virtual server to the service.
   Bing the service to the load balancing virtual server.

   Navigate to **Traffic Management** > **Virtual Servers**, select a virtual server and then select **AUDITLOGHASH** in the **Load Balancing Method.**

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.
   Navigate to **System** > **Auditing**, click **Servers** and add a server by selecting **LB Vserver** option in **Servers**.

5. Add a SYSLOG policy by specifying the rule and action.
   Navigate to **System** > **Syslog**, click **Policies** and add a SYSLOG policy.

6. Bind the SYSLOG policy to the system global for the policy to take effect.
   Navigate to **System** > **Syslog**, select a SYSLOG policy and click **Action**, and then click **Global Bindings** and bind the policy to system global.

Example

The following configuration specifies load balance of SYSLOG messages among the external log servers using the AUDITLOGHASH as load balancing method. AUDITLOGHASH method load balances the traffic based on input hash value from the audit agents. The agents are the modules which generate auditlog in a NetScaler appliance. For example, if an agent LSN wants to load balance auditlogs based on client IP address, LSN module generates the hash value based on clientIP and passes the hash value to auditlog module. The auditlog module sends the auditlog messages which have same hash value to the external syslog server.

The NetScaler appliance generates SYSLOG events and messages that are load balanced amongst the services, service1, service2, and service 3.

```
add service service1 192.0.2.10 SYSLOGUDP 514
add service service2 192.0.2.11 SYSLOGUDP 514
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add service service3 192.0.2.11 SYSLOGUDP 514
add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
bind lb vserver lbvserver1 service1
bind lb vserver lbvserver1 service2
bind lb vserver lbvserver1 service3
add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
add syslogpolicy syspol1 ns_true sysaction1
bind system global syspol1
```

Limitations

- The NetScaler appliance does not support an external load balancing virtual server load balancing the SYSLOG messages among the log servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Default Settings for the Log Properties

Mar 28, 2012

The following is an example of the default filter with default settings for the log properties:

```
begin default
 logInterval Hourly
 logFileSizeLimit 10
 logFilenameFormat auditlog%{%y%m%d}t.log
 end default
```

Following are two examples of defining the default filters:

Example 1

```
Filter f1 IP 192.168.10.1
```

This creates a log file for NSI 192.168.10.1 with the default values of the log in effect.

Example 2

```
Filter f1 IP 192.168.10.1
begin f1
 logFilenameFormat logfiles.log
end f1
```

This creates a log file for NSIP 192.168.10.1. Since the log file name format is specified, the default values of the other log properties are in effect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Sample Configuration File (audit.conf)

Mar 28, 2012

Following is a sample configuration file:

```
##############################
# This is the Auditserver configuration file
# Only the default filter is active
# Remove leading # to activate other filters
##############################
MYIP <NSAuditserverIP>
MYPORT 3023
# Filter filter_nsip  IP <Specify the NetScaler IP address to filter on > ON
# begin filter_nsip
#  logInterval   Hourly
#  logFileSizeLimit  10
#  logDirectory   logdir\%A\
#  logFilenameFormat  nsip%{%d%m%Y}t.log
# end filter_nsip
Filter default
begin default
 logInterval   Hourly
 logFileSizeLimit  10
 logFilenameFormat  auditlog%{%y%m%d}t.log
end default
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Web Server Logging

Jun 25, 2014

You can use the Web server logging feature to send logs of HTTP and HTTPS requests to a client system for storage and retrieval. This feature has two components:

- The Web log server, which runs on the NetScaler.
- The NetScaler Web Logging (NSWL) client, which runs on the client system.

When you run the NetScaler Web Logging (NSWL) client:

1. It connects to the NetScaler.
2. The NetScaler buffers the HTTP and HTTPS request log entries before sending them to the client.
3. The client can filter the entries before storing them.

To configure Web server logging, you first enable the Web logging feature on the NetScaler and configure the size of the buffer for temporarily storing the log entries. Then, you install NSWL on the client system. You then add the NetScaler IP address (NSIP) to the NSWL configuration file. You are now ready to start the NSWL client to begin logging. You can customize Web server logging by making additional modifications to the NSWL configuration file (log.conf).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NetScaler for Web Server Logging

Oct 09, 2016

To configure the NetScaler for web server logging you are required to only enable the Web Server Logging feature. Optionally, you can perform the following configurations:

- Modify the size of the buffer (default size is 16 MB) that stores the logged information before it is sent to the NetScaler Web Logging (NSWL) client.
- Specify the custom HTTP headers that you want to export to the NSWL client. You can configure a maximum of two HTTP request and two HTTP response header names.

To configure web server logging by using the command line interface

At the command prompt, perform the following operations:

- Enable the web server logging feature.
  enable ns feature WL

- [Optional] Modify the buffer size for storing the logged information.
  set ns weblogparam -bufferSizeMB <size>

  Note: To activate your modification, you must disable and then re-enable the Web server logging feature.
- [Optional] Specify the custom HTTP header names that you want to export.
  set ns weblogparam [-customReqHdrs <string> ...] [-customRspHdrs <string> ...]

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> enable ns feature WL

Done

> set ns weblogparam -bufferSizeMB 60

Done

> show ns weblogparam

    Web Logging parameters:

    Log buffer size:  60MB

    Custom HTTP request headers: (none)

    Custom HTTP response headers: (none)

Done

> set ns weblogparam -customReqHdrs req1 req2 -customRspHdrs res1 res2

Done

> show ns weblogparam

    Web Logging parameters:

    Log buffer size:  60MB

    Custom HTTP request headers: req1, req2

    Custom HTTP response headers: res1, res2

Done
```

To configure web server logging by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Navigate to System > Settings and perform the following operations:

- To enable the web server logging feature, click Change Advanced Features and select Web Logging.
- To modify the buffer size, click Change Global System Settings and under Web Logging, enter the buffer size.
- To specify the custom HTTP headers to be exported, click Change Global System Settings and under Web Logging, specify the header values.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing the NetScaler Web Logging (NSWL) Client

Jun 01, 2015

During installation, the NSWL client executable file (nswl) is installed along with other files. The nswl executable file provides a list of options that you can use. For details, see Configuring the NSWL Client.

Attention: The version of the NSWL client must be the same as that of the NetScaler. For example, if the version of the NetScaler is 10.1 Build 125.9, the NSWL client must also be of the same version.
The following table lists the operating systems on which the NSWL client can be installed.

Table 1. Supported Platforms for the NSWL Client with hardware requirements

Operating system	Version	Hardware requirements	Remarks
Windows	• Windows XP Professional • Windows Server 2003 • Windows 2000/NT • Windows Server 2008 • Windows Server 2008 R2	Processor - Intel x86 ~501 MHz RAM - 512 MB Controller - SCSI	
Mac OS	Mac OS 8.6 or later	-	Not supported on NetScaler 10.1 and later releases.
Linux	• RedHat Linux 4 or later • SUSE Linux Enterprise 9.3 or later	Processor - Intel x86 ~501 MHz RAM - 512 MB Controller - SCSI	
Solaris	Solaris Sun OS 5.6 or later	Processor - UltraSPARC-IIi 400 MHz RAM - 512 MB Controller - SCSI	Not supported on NetScaler 10.5 and later releases.
FreeBSD	FreeBSD 6.3 or later	Processor - Intel x86 ~501 MHz RAM - 512 MB Controller - SCSI	For NetScaler 10.5, use only FreeBSD 8.4.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operating system	Version	Hardware requirements	Remarks
AIX	AIX 6.1		Not supported on NetScaler 10.5 and later releases.

If the NSWL client system cannot process the log transaction because of a CPU limitation, the Web log buffer overruns and the logging process reinitiates.

Caution: Reinitiation of logging can result in loss of log transactions.
To temporarily solve a NSWL client system bottleneck caused by a CPU limitation, you can tune the Web server logging buffer size on the NetScaler appliance. To solve the problem, you need a client system that can handle the site's throughput.

This document includes the following details:

- Downloading the NSWL Client
- Installing the NSWL Client on a Solaris System
- Installing the NSWL Client on a Linux System
- Installing the NSWL Client on a FreeBSD System
- Installing the NSWL Client on a Mac System
- Installing the NSWL Client on a Windows System
- Installing the NSWL Client on a AIX System

Downloading the NSWL Client

Updated: 2014-06-25

You can obtain the NSWL client package from either the NetScaler product CD or the Citrix downloads site. Within the package there are separate installation packages for each supported platforms.

To download the NSWL client package from the Citrix site

1. Open the URL: https://www.citrix.com/downloads.html.
2. Log in to the site using your credentials.
3. Open the page for the required release number and build.
4. In the page, under Weblog Clients, click Download. The package has the name format as follows: Weblog-<release number>-<build number>.zip.

Installing the NSWL Client on a Solaris System

Updated: 2014-06-25

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the nswl_solaris-<release number>-<build number>.tar file from the package.
2. Copy the extracted file to a Solaris system on which you want to install the NSWL client.
3. Extract the files from the tar file with the following command:
   tar xvf nswl_solaris-9.3-51.5.tar
   A directory NSweblog is created in the temporary directory, and the files are extracted to the NSweblog directory.

4. Install the package with the following command:
   pkgadd -d
   The list of available packages appears. In the following example, one NSweblog package is shown:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1 NSweblog NetScaler Weblogging (SunOS,sparc) 7.0

5. You are prompted to select the packages. Select the package number of the NSweblog to be installed.
   After you select the package number and press Enter, the files are extracted and installed in the following directories:
   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples
6. To check whether the NSWL package is installed, execute the following command:
   pkginfo | grep NSweblog
   Note: To uninstall the NSWL package, execute the following command:
   pkgrm NSweblog

Installing the NSWL Client on a Linux System

Updated: 2014-06-25

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the nswl_linux-<release number>-<build number>.rpm file from the package.
2. Copy the extracted file to a system, running Linux OS, on which you want to install the NSWL client.
3. To install the NSWL package, execute the following command:
   rpm -i nswl_linux-9.3-51.5.rpm
   This command extracts the files and installs them in the following directories.
   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples
   Note: To uninstall the NSWL package, execute the following command:
   rpm -e NSweblog
   Note: To get more information about the NSweblog RPM file, execute the following command:
   rpm -qpi *.rpm
   Note: To view the installed Web server logging files, execute the following command:
   rpm -qpl *.rpm

Installing the NSWL Client on a FreeBSD System

Updated: 2014-06-25

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the nswl_bsd-<release number>-<build number>.tgz file from the package.
2. Copy the extracted file to a system, running FreeBSD OS, on which you want to install the NSWL client.
3. To install the NSWL package, execute the following command:
   pkg_add nswl_bsd-9.3-51.5.tgz
   This command extracts the files and installs them in the following directories.
   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples
   Note: To uninstall the NSWL package, execute the following command:
   pkg_delete NSweblog
4. To verify that the package is installed, execute the following command:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pkg_info | grep NSweblog

Installing the NSWL Client on a Mac System

Updated: 2014-06-25

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the nswl_macos-<release number>-<build number>.tgz file from the package.
2. Copy the extracted file to a system, running Mac OS, on which you want to install the NSWL client.
3. To install the NSWL package, execute the following command:
   pkg_add nswl_macos-9.3-51.5.tgz
   This command extracts the files and installs them in the following directories:
   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples
   Note: To uninstall the NSWL package, execute the following command:
   pkg_delete NSweblog
4. To verify that the package is installed, execute the following command:
   pkg_info | grep NSweblog

Installing the NSWL Client on a Windows System

Updated: 2014-09-18

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the nswl_win-<release number>-<build number>.zip file from the package.
2. Copy the extracted file to a Windows system on which you want to install the NSWL client.
3. On the Windows system, unzip the file in a directory (referred as <NSWL-HOME>). The following directories are extracted: bin, etc, and samples.
4. At the command prompt, run the following command from the <NSWL-HOME>\bin directory:
   nswl -install -f <directorypath>\log.conf
   where,

   <directorypath> refers to the path of the configuration file (log.conf). By default, the file is in the <NSWL-HOME>\etc directory. However, you can copy the configuration file to any other directory.

   Note: To uninstall the NSWL client, at the command prompt, run the following command from the <NSWL-HOME>\bin directory:
   > nswl -remove

Installing the NSWL Client on a AIX System

Updated: 2014-06-25

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the nswl_aix-<release number>-<build number>.rpm file from the package.
2. Copy the extracted file to a system, running AIX OS, on which you want to install the NSWL client.
3. To install the NSWL package, execute the following command:
   rpm -i nswl_aix-9.3-51.5.rpm

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

This command extracts the files and installs them in the following directories.

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

Note: To uninstall the NSWL package, execute the following command:

```
rpm -e NSweblog
```

Note: To get more information about the NSweblog RPM file, execute the following command:

```
rpm -qpi *.rpm
```

Note: To view the installed Web server logging files, execute the following command:

```
rpm -qpl *.rpm
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NSWL Client

Jun 01, 2015

After installing the NSWL client, you can configure the NSWL client using the nswl executable. These configurations are then stored in the NSWL client configuration file (log.conf).

Note: You can further customize logging on the NSWL client system by making additional modifications to the NSWL configuration file (log.conf). For details, see Customizing Logging on the NSWL Client System.
The following table describes the commands that you can use to configure the NSWL client.

NSWL command	Specifies
nswl -help	The available NSWL help options.
nswl -addns -f <path-to-configuration-file>	The system that gathers the log transaction data. You are prompted to enter the IP address of the NetScaler appliance. Enter a valid user name and password.
nswl -verify -f <path-to-configuration-file>	Check for syntax or semantic errors in the configuration file.
nswl -start -f <path-to-configuration-file>	Start the NSWL client based on the settings in the configuration file. Note: For Solaris and Linux: To start Web server logging as a background process, type the ampersand sign (&) at the end of the command.
nswl -stop (Solaris and Linux only)	Stop the NSWL client if it was started as a background process; otherwise, use CTRL+C to stop Web server logging.
nswl -install -f <path-to-configuration-file> (Windows only)	Install the NSWL client as a service in Windows.
nswl -startservice (Windows only)	Start the NSWL client by using the settings in the configuration file specified in the nswl install option. You can also start NSWL client from Start > Control Panel > Services. **Note:** The NSWL log files will be created in **C:\Windows\SysWOW64\.**
nswl -stopservice (Windows only)	Stops the NSWL client.
nswl -remove	Remove the NSWL client service from the registry.

Run the following commands from the directory in which the NSWL executable is located:

- Windows: \ns\bin
- Solaris and Linux: \usr\local\netscaler\bin

The Web server logging configuration files are located in the following directory path:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Windows: \ns\etc
- Solaris and Linux: \usr\local\netscaler\etc

The NSWL executable is started as .\nswl in Linux and Solaris.

This document includes the following details:
- Adding the IP Addresses of the NetScaler Appliance
- Verifying the NSWL Configuration File
- Running the NSWL Client

Adding the IP Addresses of the NetScaler Appliance

Updated: 2013-07-17

In the NSWL client configuration file (log.conf), add the NetScaler IP address (NSIP) from which the NSWL client will start collecting logs.

To add the NSIP address of the NetScaler appliance

1. At the client system command prompt, type:

   nswl -addns -f < directorypath > \log.conf

   < directorypath >: Specifies the path to the configuration file (log.conf).

2. At the next prompt, enter the following information:

   - **NSIP**: Specify the IP address of the NetScaler appliance.
   - Username and Password: Specify the nsroot user credentials of the NetScaler appliance.

Note: If you add multiple NetScaler IP addresses (NSIP), and later you do not want to log all of NetScaler system log details, you can delete the NSIPs manually by removing the NSIP statement at the end of the log.conf file. During a failover setup, you must add both primary and secondary NetScaler IP addresses to the log.conf by using the command. Before adding the IP address, make sure the user name and password exist on the NetScaler appliances.

Verifying the NSWL Configuration File

To make sure that logging works correctly, check the NSWL configuration file (log.conf) on the client system for syntax errors.

To verify the configuration in the NSWL configuration file

At the client system command prompt, type:

nswl -verify -f <directorypath>\log.conf

< directorypath >: Specifies the path to the configuration file (log.conf).

Running the NSWL Client

To start Web server logging

At the client system command prompt, type:

nswl -start -f <directorypath>\log.conf
<directorypath>: Specifies the path to the configuration file (log.conf).

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To stop Web server logging started as a background process on the Solaris or Linux operating systems

At the command prompt, type:

```
nswl -stop
```

To stop Web server logging started as a service on the Windows operating system

At the command prompt, type:

```
nswl -stopservice
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Customizing Logging on the NSWL Client System

Jun 01, 2015

You can customize logging on the NSWL client system by making additional modifications to the NSWL client configuration file (log.conf). Use a text editor to modify the log.conf configuration file on the client system.

To customize logging, use the configuration file to define filters and log properties.

- **Log filters.** Filter log information based on the host IP address, domain name, and host name of the Web servers.
- **Log properties.** Each filter has an associated set of log properties. Log properties define how to store the filtered log information.

This document includes the following details:

- Sample Configuration File
- Creating Filters
- Specifying Log Properties
- Understanding the NCSA and W3C Log Formats
- Creating a Custom Log Format
- Arguments for Defining a Custom Log Format
- Time Format Definition

Sample Configuration File

Following is a sample configuration file:

```
##########
# This is the NSWL configuration file
# Only the default filter is active
# Remove leading # to activate other filters
##########
##########
# Default filter (default on)
# W3C Format logging, new file is created every hour or on reaching 10MB file size,
# and the file name is Exyymmdd.log
##########
Filter default
begin default
	logFormat		W3C
	logInterval		Hourly
	logFileSizeLimit	10
	logFilenameFormat	Ex%{%y%m%d}t.log
end default
##########
# netscaler caches example
# CACHE_F filter covers all the transaction with HOST name www.netscaler.com and the listed server ip's
##########
#Filter CACHE_F HOST www.netscaler.com IP 192.168.100.89 192.168.100.95 192.168.100.52 192.168.100.53 ON
##########
# netscaler origin server example
# Not interested in Origin server to Cache traffic transaction logging
##########
#Filter ORIGIN_SERVERS IP 192.168.100.64 192.168.100.65 192.168.100.66 192.168.100.67 192.168.100.225 192.168.100.226 192.168.
100.227 192.168.100.228 OFF
##########
# netscaler image server example
# all the image server logging.
##########
#Filter IMAGE_SERVER HOST www.netscaler.images.com IP 192.168.100.71 192.168.100.72 192.168.100.169 192.168.100.170 192.168.10
0.171 ON
##########
# NCSA Format logging, new file is created every day midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log.
# Exclude objects that ends with .gif .jpg .jar.
##########
#begin ORIGIN_SERVERS
#	logFormat		NCSA
#	logInterval		Daily
#	logFileSizeLimit	40
#	logFilenameFormat	/datadisk5/ORGIN/log/%v/NS%{%m%d%y}t.log
#	logExclude		.gif .jpg .jar
#end ORIGIN_SERVERS

##########
# NCSA Format logging, new file is created every day midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log with log record timestamp as GMT.
##########
#begin CACHE_F
#	logFormat		NCSA
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
#       logInterval          Daily
#       logFileSizeLimit       20
#       logFilenameFormat /datadisk5/netscaler/log/%v/NS%{%m%d%y}t.log
#       logtime              GMT
#end CACHE_F

##########
# W3C Format logging, new file on reaching 20MB and the log file path name is
# atadisk6/netscaler/log/server's ip/Exmmyydd.log with log record timestamp as LOCAL.
##########
#begin IMAGE_SERVER
#       logFormat             W3C
#       logInterval          Size
#       logFileSizeLimit       20
#       logFilenameFormat /datadisk6/netscaler/log/%AEx%{%m%d%y}t
#       logtime              LOCAL
#end IMAGE_SERVER

##########
# Virtual Host by Name firm, can filter out the logging based on the host name by,
##########

#Filter VHOST_F IP 10.101.2.151 NETMASK 255.255.255.0
#begin VHOST_F
#       logFormat             W3C
#       logInterval          Daily
#       logFileSizeLimit       10
logFilenameFormat /ns/prod/vhost/%v/Ex%{%m%d%y}t
#end VHOST_F

########## END FILTER CONFIGURATION ##########
```

Creating Filters

Updated: 2014-01-06

You can use the default filter definition located in the configuration file (log.conf), or you can modify the filter or create a new filter. You can create more than one log filter.

Note: Consolidated logging, which logs transactions for which no filter is defined, uses the default filter if it is enabled. Consolidated logging of all servers can be done by defining only the default filter.
If the server hosts multiple Web sites and each Web site has its own domain name, and each domain is associated with a virtual server, you can configure Web server logging to create a separate log directory for each Web site. The following table displays the parameters for creating a filter.

Table 1. Parameters for Creating a Filter

Parameter	Specifies
filterName	Name of the filter. The filter name can include alphanumeric characters and cannot be longer than 59 characters. Filter names longer than 59 characters are truncated to 59 characters.
HOST name	Host name of the server for which the transactions are being logged.
IP ip	IP address of the server for which transactions are to be logged (for example, if the server has multiple domains that have one IP address).
IP ip 2...ip n:	Multiple IP addresses (for example, if the server domain has multiple IP addresses).
ip6 ip	IPv6 address of the server for which transactions are to be logged.
IP ip NETMASK mask	IP addresses and netmask combination to be used on a subnet.
ON | OFF	Enable or disable the filter to log transactions. If no argument is selected, the filter is enabled (ON).

To create a filter

To create a filter, enter the following command in the log.conf file:

- filter <filterName> <HOST name> | [IP<ip>] | [IP<ip 2...ip n>] | <IP ip NETMASK mask> [ON | OFF]
- filter <filterName> <HOST name> | [IP6 ip/<prefix length>] [ON | OFF]

To create a filter for a virtual server

To create a filter for a virtual server, enter the following command in the log.conf file:

filter <filterName> <VirtualServer IP address>
Example

In the following example, you specify an IP address of 192.168.100.0 and netmask of 255.255.255.0. The filter applies to IP addresses 192.168.100.1 through 192.168.100.254.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Filter F1 HOST www.netscaler.com ON
Filter F2 HOST www.netscaler.com IP 192.168.100.151 ON
Filter F3 HOST www.netscaler.com IP 192.168.100.151 192.165.100.152 ON
Filter F4 IP 192.168.100.151
Filter F5 IP 192.168.100.151 HOST www.netscaler.com OFF
Filter F6 HOST www.netscaler.com HOST www.xyz.com HOST www.abcxyz.com IP 192.168.100.200 ON
Filter F7 IP 192.250.100.0 NETMASK 255.255.255.0
Filter F8 HOST www.xyz.com IP 192.250.100.0 NETMASK 255.255.255.0 OFF
For creating filters for servers having IPv6 addresses.
Filter F9 2002::8/112 ON
Filter F10 HOST www.abcd.com IP6 2002::8 ON
```

Specifying Log Properties

Log properties are applied to all log entries associated with the filter. The log property definition begins with the keyword BEGIN and ends with END as illustrated in the following example:

```
BEGIN <filtername>
 logFormat ...
 logFilenameFormat ...
 logInterval ...
 logFileSize ....
 logExclude ....
 logTime ….
 END
```

Entries in the definition can include the following:

- **LogFormat** specifies the Web server logging feature that supports NCSA, W3C Extended, and custom log file formats.

  By default, the logformat property is w3c. To override, enter custom or NCSA in the configuration file, for example:

  ```
  LogFormat NCSA
  ```

  Note: For the NCSA and custom log formats, local time is used to time stamp transactions and for file rotation.
- **LogInterval** specifies the intervals at which new log files are created. Use one of the following values:

  - Hourly: A file is created every hour.
  - Daily: A file is created every day at midnight. Default value.
  - Weekly: A file is created every Sunday at midnight.
  - Monthly: A file is created on the first day of the month at midnight.
  - None: A file is created only once, when Web server logging starts.

  Example

  ```
  LogInterval Daily
  ```

- **LogFileSizeLimit** specifies the maximum size of the log file in MB. It can be used with any log interval (weekly, monthly, and so on.) A file is created when the maximum file size limit is reached or when the defined log interval time elapses.

  To override this behavior, specify the size as the loginterval property so that a file is created only when the log file size limit is reached.

  The default LogFileSizeLimit is 10 MB.

  Example

  ```
  LogFileSizeLimit 35
  ```

- **LogFilenameFormat** specifies the file name format of the log file. The name of the file can be of the following types:

  - Static: Specifies a constant string that contains the absolute path and file name.
  - Dynamic: Specifies an expression containing the following format:

    - Server IP address (%A)
    - Date (%{format}t)
    - URL suffix (%x)
    - Host name (%v)

  Example

  ```
  LogFileNameFormat Ex%{%m%d%y}t.log
  ```

  This command creates the first file name as Exmmddyy.log, then every hour creates a file with file name: Exmmddyy.log.0, Exmmddyy.log.1,..., Exmmddyy.log.n.

  Example

  ```
  LogInterval size
  LogFileSize 100
  LogFileNameFormat Ex%{%m%d%y}t
  ```

  Caution: The date format %t specified in the LogFilenameFormat command overrides the log interval property for that filter. To prevent a new file being created every day instead of when the specified log file size is reached, do not use %t in the LogFilenameFormat.
- **LogExclude** prevents logging of transactions with the specified file extensions.

  Example

  ```
  LogExclude .html
  ```

  This command creates a log file that excludes log transactions for *.html files.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **LogTime** specifies log time as either GMT or LOCAL.

  The defaults are:
  - NCSA log file format: LOCAL
  - W3C log file format: GMT.

Understanding the NCSA and W3C Log Formats

The NetScaler supports the following standard log file formats:

- NCSA Common Log Format
- W3C Extended Log Format

NCSA Common Log Format

If the log file format is NCSA, the log file displays log information in the following format:

Client_IP_address -User_Name [Date:Time -TimeZone] "Method Object HTTP_version" HTTP_StatusCode BytesSent
To use the NCSA Common log format, enter NCSA in the LogFormat argument in the log.conf file.

The following table describes the NCSA Common log format.

Table 2. NCSA Common Log Format

Argument	Specifies
Client _IP_address	The IP address of the client computer.
User Name	The user name.
Date	The date of the transaction.
Time	The time when the transaction was completed.
Time Zone	The time zone (Greenwich Mean Time or local time).
Method	The request method (for example; GET, POST).
Object	The URL.
HTTP_version	The version of HTTP used by the client.
HTTP_StatusCode	The status code in the response.
Bytes Sent	The number of bytes sent from the server.

W3C Extended Log Format

An extended log file contains a sequence of lines containing ASCII characters terminated by either a Line Feed (LF) or the sequence Carriage Return Line Feed (CRLF.) Log file generators must follow the line termination convention for the platform on which they are run.

Log analyzers must accept either LF or CRLF form. Each line may contain either a directive or an entry. If you want to use the W3C Extended log format, enter W3C as the Log-Format argument in the log.conf file.

By default, the standard W3C log format is defined internally as the custom log format, shown as follows:

%{%Y-%m-%d%H:%M:%S}t %a %u %S %A %p %m %U %q %s %j %J %T %H %+{user-agent}i %+{cookie} i%+{referer}i
You can also change the order or remove some fields in this W3C log format. For example:

logFormat W3C %{%Y-%m-%d%H:%M:%S}t %m %U
W3C log entries are created with the following format:

#Version: 1.0 #Fields: date time cs-method cs-uri #Date: 12-Jun-2001 12:34 2001-06-12 12:34:23 GET /sports/football.html 2001-06-12 12:34:30 GET /sports/football.html

Entries

Entries consist of a sequence of fields relating to a single HTTP transaction. Fields are separated by white space; Citrix recommends the use of tab characters. If a field in a particular entry is not used, a dash (-) marks the omitted field.

Directives

Directives record information about the logging process. Lines beginning with the pound sign (#) contain directives.

The following table describes the directives.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 3. Directive Descriptions

Directive	Description
Version: <integer>.<integer>	Displays the version of the extended log file format used. This document defines version 1.0.
Fields: [<specifier>...]	Identifies the fields recorded in the log.
Software: <string>	Identifies the software that generated the log.
Start-Date: <date> <time>	Displays the date and time at which the log was started.
End-Date: <date> <time>	Displays the date and time at which logging finished.
Date: <date> <time>	Displays the date and time when the entry was added.
Remark: <text>	Displays comments. Analysis tools ignore data recorded in this field.

Note: The Version and Fields directives are required. They precede all other entries in the log file.

Example

The following sample log file shows the log entries in W3C Extended log format:

#Version: 1.0 #Fields: time cs-method cs-uri #Date: 12-Jan-1996 00:00:00 00:34:23 GET /sports/football.html 12:21:16 GET /sports/football.html 12:45:52 GET /sports/football.html 12:57:34 GET /sports/foo

Fields

The Fields directive lists a sequence of field identifiers that specify the information recorded in each entry. Field identifiers may have one of the following forms:

- **identifier:** Relates to the transaction as a whole.
- **prefix-identifier:** Relates to information transfer between parties defined by the value prefix.
- **prefix (header):** Specifies the value of the HTTP header field header for transfer between parties defined by the value prefix. Fields specified in this manner always have the type <string>.

The following table describes defined prefixes.

Table 4. Prefix Descriptions

Prefix	Specifies
c	Client
s	Server
r	Remote
cs	Client to server
sc	Server to client
sr	Server to remote server (prefix used by proxies)
rs	Remote server to server (prefix used by proxies)
x	Application-specific identifier

Examples

The following examples are defined identifiers that use prefixes:

cs-method: The method in the request sent by the client to the server.

sc(Referer): The Referer field in the reply.

c-ip: The IP address of the client.

Identifiers

The following table describes the W3C Extended log format identifiers that do not require a prefix.

Table 5. W3C Extended Log Format Identifiers (No Prefix Required)

Identifier	Description

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Identifier	Description
date	The date on which the transaction was done.
time	The time when the transaction is done.
time-taken	The time taken (in seconds) for the transaction to complete.
bytes	The number of bytes transferred.
cached	Records whether a cache hit has occurred. A zero indicates a cache miss.

The following table describes the W3C Extended log format identifiers that require a prefix.

Table 6. W3C Extended Log Format Identifiers (Requires a Prefix)

Identifier	Description
IP	The IP address and the port number.
dns	The DNS name.
status	The status code.
comment	The comment returned with status code.
method	The method.
url	The URL.
url-stem	The stem portion of the URL.
url-query	The query portion of the URL.

The W3C Extended Log file format allows you to choose log fields. These fields are shown in the following table.

Table 7. W3C Extended Log File Format (Allows Log Fields)

Field	Description
Date	The date on which the transaction is done.
Time	The time when the transaction is done.
Client IP	The IP address of the client.
User Name	The user name.
Service Name	The service name, which is always HTTP.
Server IP	The server IP address.
Server Port	The server port number
Method	The request method (for example; GET, POST).
Url Stem	The URL stem.
Url Query	The query portion of the URL.
Http Status	The status code in the response.
Bytes Sent	The number of bytes sent to the server (request size, including HTTP headers).
Bytes Received	The number of bytes received from the server (response size, including HTTP headers).
Time Taken	The time taken for transaction to complete, in seconds.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Field	Description
Protocol Version	The version number of HTTP being used by the client.
User Agent	The User-Agent field in the HTTP protocol.
Cookie	The Cookie field of the HTTP protocol.
Referer	The Referer field of the HTTP protocol.

Creating a Custom Log Format

Updated: 2013-09-30

You can customize the display format of the log file data manually or by using the NSWL library. By using the custom log format, you can derive most of the log formats that Apache currently supports.

Creating a Custom Log Format by Using the NSWL Library

Use one of the following NSWL libraries depending on whether the NSWL executable has been installed on a Windows or Solaris host computer:

- **Windows:** The nswl.lib library located in \ns\bin directory on the system manager host computer.
- **Solaris:** The libnswl.a library located in /usr/local/netscaler/bin.

To create the custom log format by using the NSWL Library

1. Add the following two C functions defined by the system in a C source file:
   ns_userDefFieldName() : This function returns the string that must be added as a custom field name in the log record.

   ns_userDefFieldVal() : This function implements the custom field value, then returns it as a string that must be added at the end of the log record.

2. Compile the file into an object file.
3. Link the object file with the NSWL library (and optionally, with third party libraries) to form a new NSWL executable.
4. Add a %d string at the end of the logFormat string in the configuration file (log.conf).

Example

########## # A new file is created every midnight or on reaching 20MB file size, # and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log and create digital #signature field for each

Creating a Custom Log Format Manually

To customize the format in which log file data should appear, specify a character string as the argument of the LogFormat log property definition. For more information, see Arguments for Defining a Custom Log Format. The following is an example where character strings are used to create a log format:

LogFormat Custom ""%a - "%{user-agent}i" %[%d/%m/%Y]t %U %s %b %T"

- The string can contain the “c” type control characters \n and \t to represent new lines and tabs.
- Use the <Esc> key with literal quotes and backslashes.

The characteristics of the request are logged by placing % directives in the format string, which are replaced in the log file by the values.

If the %v (Host name) or %x (URL suffix) format specifier is present in a log file name format string, the following characters in the file name are replaced by an underscore symbol in the log configuration file name:

" * . / : < > ? \ |

Characters whose ASCII values lie in the range of 0-31 are replaced by the following:

%<ASCII value of character in hexadecimal>.

For example, the character with ASCII value 22 is replaced by %16.

Caution: If the %v format specifier is present in a log file name format string, a separate file is opened for each virtual host. To ensure continuous logging, the maximum number of files that a process can have open should be sufficiently large. See your operating system documentation for a procedure to change the number of files that can be opened.

Creating Apache Log Formats

You can derive from the custom logs most of the log formats that Apache currently supports. The custom log formats that match Apache log formats are:

NCSA/combined: LogFormat custom %h %l %u [%t] "%r" %s %B "%{referer}i" "%{user-agent}i"

NCSA/Common: LogFormat custom %h %l %u [%t] "%r" %s %B

Referer Log: LogFormat custom "%{referer}i" -> %U

Useragent: LogFormat custom %{user-agent}i

Similarly, you can derive the other server log formats from the custom formats.

Arguments for Defining a Custom Log Format

Updated: 2015-04-02

The following table describes the data that you can use as the Log Format argument string:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 8. Custom Log Format

Argument	Specifies
%a	Remote IPv4 address.
%A	Local IPv4 address.
%a6	Remote IPv6 address.
%A6	Local IPv6 address.
%B	Bytes sent, excluding the HTTP headers (response size).
%b	Bytes received, excluding the HTTP headers (request size).
%d	User-defined field.
%e1	Value of the first custom HTTP request header.
%e2	Value of the second custom HTTP request header.
%E1	Value of the first custom HTTP response header.
%E2	Value of the second custom HTTP response header.
Note: For instructions on how to export custom HTTP headers, see Configuring the NetScaler for Web Server Logging.	
%g	Greenwich Mean Time offset (for example, -0800 for Pacific Standard Time).
%h	Remote host.
%H	Request protocol.
%{Foobar}i	Contents of the Foobar: header line(s) in the request sent to the server. The system supports the User-Agent, Referer and cookie headers. The + after the % in this format informs the logging client to use the + as a word separator.
%j	Bytes received, including headers (request size)
%J	Bytes sent, including headers (response size)
%l	Remote log name (from identd, if supplied).
%m	Request method.
%M	Time taken to serve the request (in microseconds)
%{Foobar}o	Contents of Foobar: header line(s) in the reply. USER-AGENT, Referer, and cookie headers (including set cookie headers) are supported.
%p	Canonical port of the server serving the request.
%P	The admin partition.
%q	Query string (prefixed with a question mark (?) if a query string exists).
%r	First line of the request.
%s	Requests that were redirected internally, this is the status of the original request.
%t	Time, in common log format (standard English time format).
%	Time, in the form given by format, must be in the strftime(3) format. For format descriptions, see Time Format Definition.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Argument	Specifies
%T	Time taken to serve the request, in seconds.
%u	Remote user (from auth; may be bogus if return status (%s) is 401).
%U	URL path requested.
%v	Canonical name of the server serving the request.
%V	Virtual server IPv4 address in the system, if load balancing, content switching, and/or cache redirection is used.
%V6	Virtual server IPv6 address in the system, if load balancing, content switching, and/or cache redirection is used.

For example, if you define the log format as %+{user-agent}i, and if the user agent value is Citrix NetScaler system Web Client, then the information is logged as NetScaler system+Web+Client. An alternative is to use double quotation marks. For example, "%{user-agent}i" logs it as "Citrix NetScaler system Web Client." Do not use the <Esc> key on strings from %.. .r, %. . .i and, %. . .o. This complies with the requirements of the Common Log Format. Note that clients can insert control characters into the log. Therefore, you should take care when working with raw log files.

Time Format Definition

The following table lists the characters that you can enter as the format part of the %{format}t string described in the Custom Log Format table of Arguments for Defining a Custom Log Format. Values within brackets ([ ]) show the range of values that appear. For example, [1,31] in the %d description in the following table shows %d ranges from 1 to 31.

Table 9. Time Format Definition

Argument	Specifies
%%	The same as %.
%a	The abbreviated name of the week day for the locale.
%A	The full name of the week day for the locale.
%b	The abbreviated name of the month for the locale.
%B	The full name of the month for the locale.
%C	The century number (the year divided by 100 and truncated to an integer as a decimal number [1,99]); single digits are preceded by a 0.
%d	The day of month [1,31]; single digits are preceded by 0.
%e	The day of month [1,31]; single digits are preceded by a blank.
%h	The abbreviated name of the month for the locale.
%H	The hour (24-hour clock) [0,23]; single digits are preceded by a 0.
%I	The hour (12-hour clock) [1,12]; single digits are preceded by a 0.
%j	The number of the day in the year [1,366]; single digits are preceded by 0.
%k	The hour (24-hour clock) [0,23]; single digits are preceded by a blank.
%l	The hour (12-hour clock) [1,12]; single digits are preceded by a blank.
%m	The number of the month in the year [1,12]; single digits are preceded by a 0.
%M	The minute [00,59]; leading 0 is permitted but not required.
%n	Inserts a new line.
%p	The equivalent of either a.m. or p.m. for the locale.
%r	The appropriate time representation in 12-hour clock format with %p.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Argument	Specifies
%S	The seconds [00,61]; the range of values is [00,61] rather than [00,59] to allow for the occasional leap second and for the double leap second.
%3	The milliseconds [000,999]; the range of values is [000,999].
%6	The microseconds [000000,999999]; the range of values is [000000,999999].
%9	The nanoseconds [000000000,999999999]; the range of values is [000000000,999999999].
%t	Inserts a tab.
%u	The day of the week as a decimal number [1,7]. 1 represents Sunday, 2 represents Tuesday and so on.
%U	The number of the week in the year as a decimal number [00,53], with Sunday as the first day of week 1.
%w	The day of the week as a decimal number [0,6]. 0 represents Sunday.
%W	Specifies the number of the week in the year as a decimal number [00,53]. Monday is the first day of week 1.
%y	The number of the year within the century [00,99]. For example, 5 would be the fifth year of that century.
%Y	The year, including the century (for example, 1993).

Note: If you specify a conversion that does not correspond to any of the ones described in the preceding table, or to any of the modified conversion specifications listed in the next paragraph, the behavior is undefined and returns 0.
The difference between %U and %W (and also between modified conversions %OU and %OW) is the day considered to be the first day of the week. Week number 1 is the first week in January (starting with a Sunday for %U, or a Monday for %W). Week number 0 contains the days before the first Sunday or Monday in January for %U and %W.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Call Home

Aug 16, 2016

Note: The current NetScaler 1000V release does not support this feature.

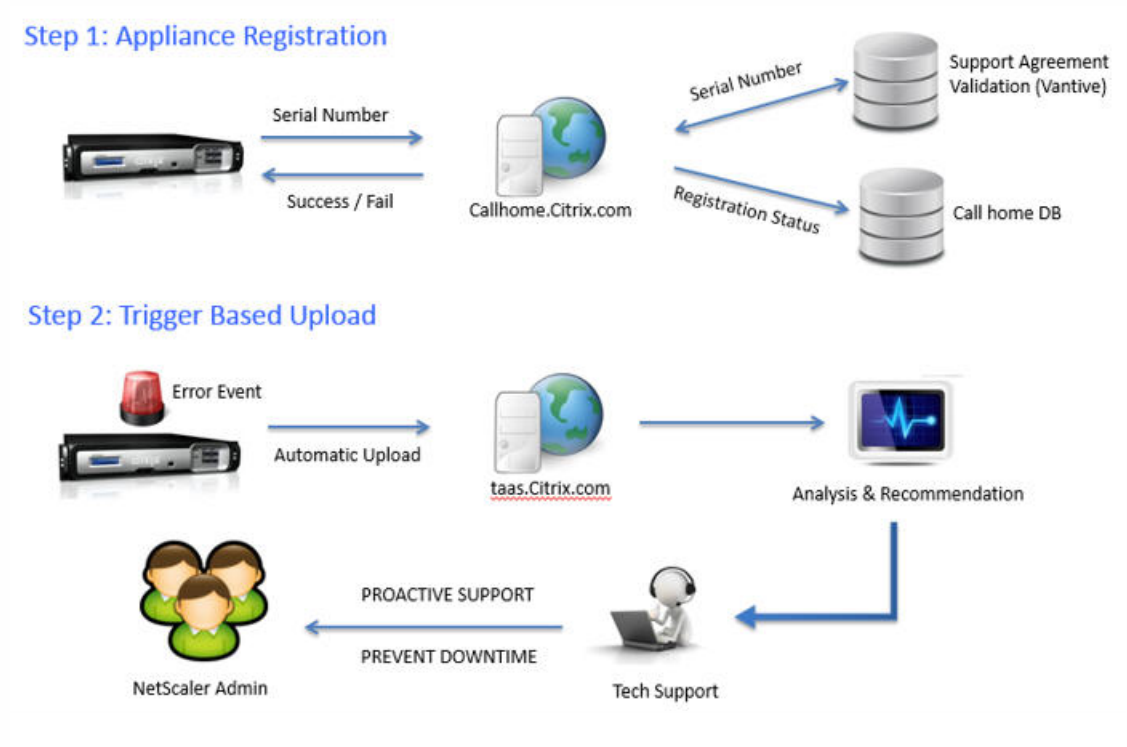
The Call Home feature monitors your NetScaler appliance for critical error conditions. Call Home registers your appliance with the Citrix Technical Support server. If your appliance is successfully registered with the Support server, Call Home automatically uploads system data to that server in the event that one of the conditions occurs. The NetScaler Appliance keeps a full log of all upload events. If you are unable to correct the problem after reviewing the appliance's log, you can contact the Citrix Technical Support team and open a service request. The team can analyze the uploaded system data and recommend possible solutions.

The Call Home feature is supported on all three platforms of NetScaler ADC.

- In NetScaler MPX, Call Home feature is supported on all MPX models.
- In NetScaler VPX, Call Home feature is supported only for VPX-1000 and VPX 3000 models.
  Note: CallHome support has been extended for VPX appliances that obtain their licenses from external or central licensing pools. However, the feature remains same as for a standard VPX appliance.

- In NetScaler SDX, Call Home feature is supported on VPX instances running on a SDX platform. It monitors the Hard Disk and assigned SSL chips for any errors or failures. The VPX instance however does not have access to the Power Supply Unit (PSU) and therefore CallHome does not monitor its status. In a SDX platform, you can configure CallHome either directly on an individual instance or through the SVM. **Note**: Monitoring of other SDX components is not supported.E.

Following is a typical setup for Call Home.



To implement Call Home, you must do the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Authenticating the appliance
- Enabling Call Home
- Registering the appliance
- Uploading the system data
- Generating a service request

Authenticating the appliance. If your NetScaler appliance does not have direct Internet connectivity, you can configure a proxy server through which system logs generated by Call Home feature are uploaded to the Citrix server. When using a proxy server, Call Home provides three types of security authentication.

1. No authentication. If both the appliance and the proxy server are located at the customer end, no authentication is needed.

2. Proxy Server authentication. One-way authentication, in which the proxy server is authenticated by the NetScaler appliance. To configure one-way proxy server authentication, do the following:

1. Add an SSL service for the proxy server IP address, and enable server authentication for the service.
2. Bind a CA certificate to the service.
3. Bind an HTTPS Monitor to the service.
   When the service is up and running, it indicates that the server is authenticated, and you can proceed to the next step.
4. Set CallHome to use the newly added service

3. NetScaler and Proxy Server authentication. Two-way authentication, in which the NetScaler appliance is also authenticated. To configure a two-way authentication, do the following:

1. Add a SSL service for proxy server IP
2. Bind a CA certificate to the service.
3. Bind a Client Certificate.
4. Bind HTTPS Monitor to the service.
5. Set CallHome to use the new added service

Enabling Call Home. By default, the Citrix Call Home feature is disabled on the appliance. You must first enable the service to register the appliance for critical error conditions.

Registering a NetScaler appliance. The appliance has to be registered to the Citrix Technical Support server before Call Home can upload the system data to the server when predefined error conditions occur on the appliance.to the Citrix technical support server.

1. The Call Home process sends the following details to the Citrix Technical Support server:
   - Hardware serial number is shared for NetScaler MPX and SDX models
   - License serial number is shared for NetScaler VPX models.
2. The server checks its database for an active technical support service contract for the appliance.
3. If there is an active technical support service contract, the support server registers the NetScaler appliance for Call Home and sends a successful-registration response to the appliance stating that the feature is successfully enabled. If there is no active technical support service contract, the server sends a registration-failure response to the NetScaler appliance.

The following table lists the error conditions that Call Home currently monitors on a NetScaler appliance:

Table 1. List of error conditions monitored by Call Home

Error Condition	Indicates	Call Home Monitoring Interval	Corresponding SNMP Alarm Name
Compact flash drive errors	The compact flash drive on the appliance that encountered read or write errors.	24 hours	COMPACT-FLASH-ERRORS
Hard disk drive errors	The hard drives on the appliance that encountered read or write errors.	24 hours	HARD-DISK-DRIVE-ERRORS
Power supply unit failure	One of the power supply units on the NetScaler appliance has failed.	7 seconds	POWER-SUPPLY-FAILURE
SSL card failure	One of the SSL cards on the NetScaler appliance has failed.	7 seconds	SSL-CARD-FAILED

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Warm restart	The appliance has warm restarted due to a failure of a system process.	After every restart of the NetScaler appliance.	WARM-RESTART-EVENT

Note: The Call Home feature do not monitor the power supply unit (PSU) status for VPX models and VPX instances.

Uploading System Data. An error condition triggers the following sequence of events:

1. The Call Home process checks the registration status. If the status indicates successful registration, the process advances to the next step.

2. The Call Home process runs a showtech support script that collects all of the system related data in a tar file. The data in the tar file includes configurations, logs, and statistics. Call Home locally saves the tar file at /var/tmp/support/callhome.
3. Call Home uploads a copy of the tar file to the Citrix Technical Support server. The Appliance logs the uploading of the tar file in a log file named callhome.log located at /var/log. You can also configure the CALLHOME-UPLOAD-EVENT SNMP alarm to generate an SNMP alert whenever Call Home uploads happen.
4. If the SNMP alarm related to the error condition is enabled, the SNMP agent on the appliance generates an SNMP trap message and sends it to all of the configured SNMP trap destinations. For more information about configuring SNMP alarms and trap destinations, see "Configuring the NetScaler to Generate SNMPv1 and SNMPv2 Traps."

Note:

- Call Home creates the Call Home tar file and uploads it to the CITRIX tech support server for only the first occurrence of a particular error condition since the appliance was last restarted. If you want the NetScaler appliance to send you alerts each time a particular error condition occurs, configure the corresponding SNMP alarm for the error condition.
- For the warm restart error condition, the Call Home tar file is uploaded to the server again only if the result of the failure is different from the result of the previous failure.

The Call Home tar file has the following name format:

collector_callhome_<NSIP of the appliance>_<P for Primary or standalone, or S for Secondary>_<date>_<hours, in 24 hr format, according to the local time zone>_<minutes>.tar.gz. For example, collector_callhome_10.105.13.100_P_2Feb2012_20_30.tar.gz.

Generating a Service Request. After you review the logs and SNMP trap messages for Call Home upload events, you have the option of contacting the Citrix Technical Support team and opening a service request. For more information about contacting the team and opening a service request, seehttp://support.citrix.com/article/CTX132307.

The Support team can then analyze the system data in the uploaded Call Home tar files and sends recommendations for possible solutions to the administrator's email address.

Before you begin configuring Call Home, do the following:

- Make sure that the NetScaler appliance is connected to the Internet or to a proxy server that has internet connectivity.
- Make sure that you have an active Citrix Technical Support service contract for the appliance.

Configuring Call Home on the NetScaler appliance consists of the following tasks:

1. Enable the Call Home feature. When you enable the Call Home feature, the Call Home process registers the appliance with the Citrix Technical Support server. The registration takes some time to complete. During that time, the appliance displays the status as "IN PROGRESS". When the registration is complete, the appliance displays the status as "SUCCESSFUL."
   Note: While upgrading the NetScaler appliance from an older release to release 10.1 or later, the NetScaler appliance prompts you to enable the Call Home feature, if:

   - The Call Home feature is not supported in the older release.
   - The Call Home feature is disabled in the older release.
2. (Optional) Specify the administrator's email address. The Call Home process sends the email address to the Support server, where it is stored for future correspondence regarding Call Home uploads.
3. (Optional) Specify Proxy server settings. Netscaler appliance needs internet connectivity to upload the collector archive to the Citrix Technical Support server. If the appliance does not have internet connectivity, then a proxy server (having internet connectivity) can be configured to upload the data.
4. (Optional) Enable the CALLHOME-UPLOAD-EVENT SNMP alarm. The SNMP agent on the NetScaler appliance generates a trap message and sends to all the configured SNMP trap destinations. The message includes the status of uploading of the Call Home tar file by the Call Home process. For more information

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

about configuring SNMP alarms and trap destinations, see "Configuring the NetScaler to Generate SNMPv1 and SNMPv2 Traps."

5. (Optional) Enable all of the corresponding SNMP alarms. Call Home creates and uploads a Call Home tar file for the first occurrence of a monitored error condition since the appliance was last restarted. If you want to be alerted of these error conditions, you can configure the corresponding SNMP alarm. Table 1 lists all the corresponding SNMP alarms. For more information about configuring SNMP alarms and trap destinations, see "Configuring the NetScaler to Generate SNMPv1 and SNMPv2 Traps."

To enable Call Home by using the command line interface

At the command prompt, type any of the following:

- enable ns feature ch
- enable ns feature callhome

To check the status of the appliance's registration to the Support server by using the command line interface

At the command prompt, type:

show callhome

Example

```
> enable ns feature ch
Done

> show callhome
 Callhome feature: ENABLED
 Registration with Citrix upload server IN PROGRESS

 E-mail address configured:
 Proxy mode:NO   Ipaddress:     Port:0

 Trigger event                State    First occurrence   Latest occurrence
 -------------                -----    ----------------   -----------------
 1) Compact flash errors       Enabled   ..                ..
 2) Hard disk drive errors     Enabled   ..                ..
 3) Power supply unit failure  Enabled   ..                ..
 4) SSL card failure           Enabled   ..                ..
 5) Warm restart               Enabled   N/A               ..
 Done

 > show callhome
 Callhome feature: ENABLED
 Registration with Citrix upload server SUCCESSFUL

 E-mail address configured:
 Proxy mode:NO   Ipaddress:     Port:0

 Trigger event                State    First occurrence   Latest occurrence
 -------------                -----    ----------------   -----------------
 1) Compact flash errors       Enabled   ..                ..
 2) Hard disk drive errors     Enabled   ..                ..
 3) Power supply unit failure  Enabled   ..                ..
 4) SSL card failure           Enabled   ..                ..
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 5) Warm restart              Enabled    N/A              ..
 Done
```

To specify the administrator's email address and proxy server settings by using the command line interface

At the command prompt, type:

- set callhome -emailAddress <string>
- set callhome -proxyMode (YES | NO) [-IPAddress <ip_addr|ipv6_addr|*>] [-port <port|*>]
- show callhome

Example

```
> set callhome -emailAddress exampleadmin@example.com
Done

> set callhome –proxyMode Yes –IPAddress 10.102.167.33 –port 80
Done

> show callhome
 E-mail address configured: exampleadmin@example.com
 Proxy mode:YES   Ipaddress: 10.102.167.33      Port:80

 Trigger event                State     First occurrence   Latest occurrence
 -------------                -----     ----------------   -----------------
 1) Compact flash errors      Enabled   ..                 ..
 2) Hard disk drive errors    Enabled   ..                 ..
 3) Power supply unit failure Enabled   ..                 ..
 4) SSL card failure          Enabled   ..                 ..
 5) Warm restart              Enabled   N/A                ..
 Done
```

To enable Call Home proxy mode by using the command line interface

At the command prompt, type:

- set callhome -ipAddress <ipaddress> -port <port> -proxyMode [yes | no]
- show callhome
  Note: Proxy mode is enabled only when the -proxymode parameter is set to YES. If it is set to NO, the proxy functionality does not work, even if the IP address and port are configured. The port number should be for an HTTP service on the proxy server, not for an HTTPS service.

Example

```
> set callhome ipAddress 10.0.0.1 -port 80 -proxyMode yes
Done
```

To enable Call Home by using the configuration utility

Navigate to System > Settings, click Configure Advanced Features and select the Call Home option.

To check the status of the appliance's registration with the Support server by using the configuration utility

Navigation to System > Diagnostics and, in the Technical Support Tools group, select Call Home option to view the status

© 1999-2017 Citrix Systems, Inc. All rights reserved.

of registration.

To specify the administrator's email address by using the configuration utility

Navigation to System > Diagnostics and, in the Technical Support Tools group, select Call Home option to specify the administrator's email address.

To enable Call Home proxy mode by using the configuration utility

Navigation to System > Diagnostics and, in the Technical Support Tools group, select Call Home option to specify the proxy server's IP address and the port number.

Navigate to **NetScaler** > **Call Home,** select a NetScaler instance and click **Enable** option.

To enable Proxy Server Authentication by using the configuration utility

Navigate to **System** > **Diagnostics** and under **Technical Support Tools**, select **Call Home** option to select the proxy authentication service type.

To upload the technical support collector archive by using the NetScaler GUI

1. Navigate to **System** > **Diagnostics** page and click **Generate support file.**
2. In the **Tech Support** page, select the scope of archive from the drop-down list and select the **Upload the Collector Archive** check box to set the following options:
   1. Proxy Server (Optional) — Name of the server used for uploading the collective archive. You can use this parameter if the appliance does not have a direct internet connection or DNS configuration on this system. The basic proxy format is proxy IP:<proxy_port>. If the proxy requires authentication, the format isusername:**password@proxy_IP:<proxy_port>**.
   2. Case or Service Request Number—Case serial number if the case is already opened by Citrix Technical Support.
   3. Collector Archive File—The name of the collector archive file (for example,**/var/tmp/supported/collector_P_10.102.29.200_1Jun2016_15_11.tar.gz)** to be uploaded. If the file name is specified, no new collector archive is generated.
   4. Description—A brief description about the archive upload.
   5. User Name—My Citrix user name to log on to the Citrix Upload server.
   6. Password—My Citrix account password to gain access into Citrix Upload server.
3. Click **Run** to automatically upload the collected archive on to the Citrix Upload server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Reporting Tool

Jun 01, 2015

Use the Citrix® NetScaler® Reporting tool to view NetScaler performance statistics data as reports. Statistics data are collected by the nscollect utility and are stored in a database. When you want to view certain performance data over a period of time, the Reporting tool pulls out specified data from the database and displays them in charts.

Reports are a collection of charts. The Reporting tool provides built-in reports as well as the option to create custom reports. In a report, you can modify the charts and add new charts. You can also modify the operation of the data collection utility, nscollect, and stop or start its operation.

This document includes the following details:

- Using the Reporting Tool
- Working with Reports
- Working with Charts
- Examples
- Stopping and Starting the Data Collection Utility

Using the Reporting Tool

The Reporting tool is a Web-based interface accessed from the Citrix® NetScaler® appliance. Use the Reporting tool to display the performance statistics data as reports containing graphs. In addition to using the built-in reports, you can create custom reports, which you can modify at any time. Reports can have between one and four charts. You can create up to 256 custom reports.

To invoke the Reporting tool

1. Use the Web browser of your choice to connect to the IP address of the NetScaler (for example, http://10.102.29.170/). The Web Logon screen appears.
2. In the User Name text box, type the user name assigned to the NetScaler.
3. In the Password text box, type the password.
4. In the Start in drop-down box, select Reporting.
5. Click Login.

The following screen shots show the report toolbar and the chart toolbar, which are frequently referenced in this documentation.

Figure 1. *Report Toolbar*

Figure 2. *Chart Toolbar*

Working with Reports

Updated: 2013-09-27

You can plot and monitor statistics for the various functional groups configured on the NetScaler over a specified time interval. Reports enable you to troubleshoot or analyze the behavior of your appliance. There are two types of reports: built-in reports and custom reports. Report content for built-in or custom reports can be viewed in a graphical format or a

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tabular format. The graphical view consists of line, area, and bar charts that can display up to 32 sets of data (counters). The tabular view displays the data in columns and rows. This view is useful for debugging error counters.

The default report that is displayed in the Reporting tool is CPU vs. Memory Usage and HTTP Requests Rate. You can change the default report view by displaying the report you want as your default view, and then clicking Default Report.

Reports can be generated for the last hour, last day, last week, last month, last year, or you can customize the duration.

You can do the following with reports:

- Toggle between a tabular view of data and a graphical view of data.
- Change the graphical display type, such as bar chart or line chart.
- Customize charts in a report.
- Export the chart as an Excel comma-separated value (CSV) file.
- View the charts in detail by zooming in, zooming out, or using a drag-and-drop operation (scrolling).
- Set a report as the default report for viewing whenever you log on.
- Add or remove counters.
- Print reports.
- Refresh reports to view the latest performance data.

Using Built-in Reports

The Reporting tool provides built-in reports for frequently viewed data. Built-in reports are available for the following functional groups: System, Network, SSL, Compression, Integrated Cache, NetScaler Gateway, and Citrix NetScaler Application Firewall. By default, the built-in reports are displayed for the last day. However, you can view the reports for the last hour, last week, last month, or last year.

Note: You cannot save changes to built-in reports, but you can save a modified built-in report as a custom report.
To display a built-in report

1. In the left pane of the Reporting tool, under Built-in Reports, expand a group (for example, SSL).
2. Click a report (for example, SSL > All Backend Ciphers).

Creating and Deleting Reports

You can create your own custom reports and save them with user-defined names for reuse. You can plot different counters for different groups based on your requirements. You can create up to 256 custom reports.

You can either create a new report or save a built-in report as a custom report. By default, a newly created custom report contains one chart named System Overview, which displays the CPU Usage counter plotted for the last day. You can customize the interval and set the data source and time zone from the report toolbar. Within a report, you can use the chart toolbars to add, modify, or delete charts, as described in "[Working with Charts]."

By default, newly created custom reports contain one chart named System Overview that displays a CPU Usage counter plotted for the last day.

To create a custom report

1. In the Reporting tool, on the report toolbar, click Create, or if you want to create a new custom report based on an existing report, open the existing report, and then click Save As.
2. In Report Name box, type a name for the custom report.
3. Do one of the following:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- To add the report to an existing folder, in Create in or Save in, click the down arrow to choose an existing folder, and then click OK.
- To create a new folder to store the report, click the Click to add folder icon, in Folder Name, type the name of the folder, and in Create in, specify where you want the new folder to reside in the hierarchy, and then click OK.

Note: You can create up to 128 folders.

To delete a custom report

1. In the left pane of the Reporting tool, next to Custom Reports, click the Click to manage custom reports icon.
2. Select the check box that corresponds with the report you want to delete, and then click Delete.

   Note: When you delete a folder, all the contents of that folder are deleted.

Modifying the Time Interval

By default, built-in reports display data for the last day. However, if you want to change the time interval for a built-in report, you can save the report as a custom report. The new interval applies to all charts in the report. The following table describes the time-interval options.

Table 1. Time Intervals

Time interval	Displays
Last Hour	Statistics data collected for the last hour.
Last Day	Statistics data collected for the last day (24 hours).
Last Week	Statistics data collected for the last week (7 days).
Last Month	Statistics data collected for the last month (31 days).
Last Year	Statistics data collected for the last year (365 days).
Custom	Statistics data collected for a time period that you are prompted to specify.

To modify the time interval

1. In the left pane of the Reporting tool, click a report.
2. On the report toolbar, click Duration, and then click a time interval.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting the Data Source and Time Zone

You can retrieve data from different data sources to display them in the reports. You can also define the time zone for the reports and apply the currently displayed report's time selection to all the reports, including the built-in reports.

To set the data source and time zone

1. In the Reporting tool, on the report toolbar, click Settings.
2. In the Settings dialog box, in Data Source, select the data source from which you want to retrieve the counter information.
3. Do one or both of the following:
   - If you want the tool to remember the time period for which a chart is plotted, select the Remember time selection for charts check box.
   - If you want the reports to use the time settings of your NetScaler appliance, select the Use Appliance's time zone check box.

Exporting and Importing Custom Reports

You can share reports with other NetScaler administrators by exporting reports. You can also import reports.

To export or import custom reports

1. In the left pane of the Reporting tool, next to Custom Reports, click the Click to manage custom reports icon.
2. Select the check box that corresponds with the report you want to export or import, and then click Export or Import.
   Note: When you export the file, it is exported in a .gz file format.

Working with Charts

Updated: 2013-09-06

Use charts to plot and monitor counters or groups of counters. You can include up to four charts in one report. In each chart, you can plot up to 32 counters. The charts can use different graphical formats (for example, area and bar). You can move the charts up or down within the report, customize the colors and visual display for each counter in a chart, and delete a chart when you do not want to monitor it.

In all report charts, the horizontal axis represents time and the vertical axis represents the value of the counter.

Adding a Chart

When you add a chart to a report, the System Overview chart appears with the CPU Usage counter plotted for the last one day. To plot a different group of statistics or select a different counter, see "Modifying a Chart."

Note: If you add charts to a built-in report, and you want to retain the report, you must save the report as a custom report. Use the following procedure to add a chart to a report.

To add a chart to a report

1. In the left pane of the Reporting tool, click a report.
2. Under the chart where you want to add the new chart, click the Add icon.

Modifying a Chart

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You can modify a chart by changing the functional group for which the statistics are displayed and by selecting different counters.

To modify a chart

1. In the left pane of the Reporting tool, click a report.
2. Under the chart that you want to modify, click Counters.
3. In the dialog box that appears, in the Title box, type a name for the chart.
4. Next to Plot chart for, do one of the following:
   - To plot counters for global counters, such as Integrated Cache and Compression, click System global statistics.
   - To plot entity counters for entity types, such as Load Balancing and GSLB, click System entities statistics.
5. In Select group, click the desired entity.
6. Under Counters, in Available, click the counter name(s) that you want to plot, and then click the > button.
7. If you selected System entities statistics in step 4, on the Entities tab, under Available, click the entity instance name(s) you want to plot, and then click the > button.
8. Click OK.

Viewing a Chart

You can specify the graphical formats of the plotted counters in a chart. Charts can be viewed as line charts, spline charts, step-line charts, scatter charts, area charts, bar charts, stacked area charts, and stacked bar charts. You can also zoom in, zoom out, or scroll inside the plot area of a chart. You can zoom in or out for all data sources for 1 hour, 1 day, 1 week, 1 month, 1 year, and 3 years.

Other options for customizing the view of a chart include customizing the axes of the charts, changing the background and edge color of the plot area, customizing the color and size of the grids, and customizing the display of each data set (counter) in a chart.

Data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if CPU usage and Memory usage are displayed in first and second order at the bottom of the chart, CPU usage is equal to Data Set 1 and Memory usage is equal to Data Set 2.

Whenever you modify a built-in report, you need to save the report as a custom report to retain your changes.

To change the graph type of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart you want to view, on the chart toolbar, click Customize.
3. On the Chart tab, under Category, click Plot type, and then click the graph type you want to display for the chart. If you want to display the graph is 3D, select the Use 3D check box.

To refocus a chart with detailed data

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click Zoom In, and do one or both of the following:
   - To refocus the chart to display data for a specific time window, drag and drop the cursor from the start time to the end time. For example, you can view data for a one-hour period on a certain day.
   - To refocus the chart to display data for a data point, simply click once on chart where you want to zoom in and get more detailed information.
3. Once you have the desired range of time for which you want to view detailed data, on the report toolbar, click Tabular

© 1999-2017 Citrix Systems, Inc. All rights reserved.

View. Tabular view displays the data in numeric form in rows and columns.

To view numeric data for a graph

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click Tabular View. To return to the graphical view, click Graphical View.
   Note: You can also view the numeric data in the graphical view by hovering your cursor over the notches in the gridlines.

To scroll through time in a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click Scroll, and then click inside the chart and drag the cursor in the direction for which you want to see data for a new time period. For example, if you want to view data in the past, click and drag to the left.

To change the background color and text color of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the axes, click Customize.
3. On the Chart tab, under Category, click one or more of the following:
   - To change the background color, click Background Color, and then select the options for color, transparency, and effects.
   - To change the text color, click Text Color, and then select the options for color, transparency, and effects.

To customize the axes of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the axes, click Customize.
3. On the Chart tab, under Category, click one or more of the following:
   - To change the scale of the left y-axis, click Left Y-Axis, and then select the scale you want.
   - To change the scale of the right y-axis, click Right Y-Axis, in Data set to plot, select the date set, and then select the scale you want.
     Note: The data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if CPU usage and Memory usage are displayed in first and second order at the bottom of the chart, CPU usage is equal to Data Set 1 and Memory usage is equal to Data Set 2.
   - To plot each data set in its own hidden y-axis, click Multiple Axes, and then click Enable.

To change the background color, edge color, and gridlines for a plot area of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the plot area, click Customize.
3. On the Plot Area tab, under Category, click one or more of the following:
   - To change the background color and edge color of the chart, click Background Color and Edge Color, and then select the options for color, transparency, and effects.
   - To change the horizontal or vertical grids of the chart, click Horizontal Grids or Vertical Grids, and then select the options for displaying the grids, grid width, grid color, transparency, and effects.

To change the color and graph type of a data set

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the display of the data set (counters), click Customize.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

3. On the Data Set tab, in Select Data Set, select the data set (counter) for which you want to customize the graphical display.
   Note: The data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if CPU usage and Memory usage are displayed in first and second order at the bottom of the chart, CPU usage is equal to Data Set 1 and Memory usage is equal to Data Set 2.
4. Under Category, do one of more of the following:
   - To change the background color, click Color, and then select the options for color, transparency, and effects.
   - To change the graph type, click Plot type, and then select the graph type you want to display for the data set. If you want to display the graph as 3D, select the Use 3D check box.

Exporting Chart Data to Excel

For further data analysis, you can export charts to Excel in a comma-separated value (CSV) format.

To export chart data to Excel

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart with the data you want to export to Excel, click Export.

Deleting a Chart

If you do not want to use a chart, you can remove it from the report. You can permanently remove charts from custom reports only. If you delete a chart from a built-in report and want to retain the changes, you need to save the report as a custom report.

To delete a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart that you want to delete, click the Delete icon.

Examples

To display the trend report for CPU usage and memory usage for the last week

1. In the left pane of the Reporting tool, under Built-in Reports, expand System.
2. Click the report CPU vs. Memory Usage and HTTP Requests Rate.
3. In the right pane, on the report toolbar, click Duration, and then click Last Week.

To compare the bytes received rate and the bytes transmitted rate between two interfaces for the last week

1. In the right pane, on the report toolbar, click Create.
2. In the Report Name box, type a name for the custom report (for example, Custom_Interfaces), and then click OK. The report is created with the default System Overview chart, which displays the CPU Usage counter plotted for the last hour.
3. Under System Overview, on the chart toolbar, click Counters.
4. In the counter selection pane, in Title, type a name for the chart (for example, Interfaces bytes data).
5. In Plot chart for, click System entities statistics, and then in Select Group, select Interface.
6. On the Entities tab, click the interface name(s) you want to plot (for example, 1/1 and 1/2), and then click the > button.
7. On the Counters tab, click Bytes received (Rate) and Bytes transmitted (Rate) and then click the > button.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

8. Click OK.
9. On the report toolbar, click Duration, and then click Last Week.

Stopping and Starting the Data Collection Utility

Updated: 2014-09-24

The data collection utility, nscollect, runs automatically when you start the NetScaler ADC. This utility retrieves the application performance data and stores it in the form of data sources on the ADC. You can create up to 32 data sources. The default data source is /var/log/db/default.

The data collection utility creates databases for global counters and entity-specific counters, and uses this data to generate reports. Global-counter databases are created at /var/log/db/<DataSourceName> . The entity-specific databases are created based on the entities configured on the NetScaler, and a separate folder is created for each entity type in /var/log/db/<DataSourceName/EntityNameDB>.

Nscollect retrieves data once every 5 minutes. It retains data in 5-minute granularity for one day, hourly for the last 30 days, and daily for three years.

You might have to stop and restart the data collection utility if data is not updated accurately or the reports display corrupted data.

To stop nscollect

At the command prompt, type:
/netscaler/nscollect stop

To start nscollect on the current SSH session to the NetScaler:

At the command prompt, type:

/netscaler/nscollect start

To start nscollect on the local system:

At the command prompt, type:

/netscaler/nscollect start &

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

AppFlow

May 29, 2015

The Citrix NetScaler appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. It also collects web page performance data and database information. AppFlow transmits the information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information, web page performance data, and database information.

Using UDP as the transport protocol, AppFlow transmits the collected data, called *flow records*, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor.

AppFlow use actions and policies to send records for a selected flow to specific set of collectors. An AppFlow action specifies which set of collectors will receive the AppFlow records. Policies, which are based on Advanced expressions can be configured to select flows for which flow records will be sent to the collectors specified by the associated AppFlow action.

To limit the types of flows, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

Note: This feature is supported only on NetScaler nCore builds.
This document includes the following details:

- How AppFlow Works
- Configuring the AppFlow Feature
- Exporting Performance Data of Web Pages to AppFlow Collector

How AppFlow Works

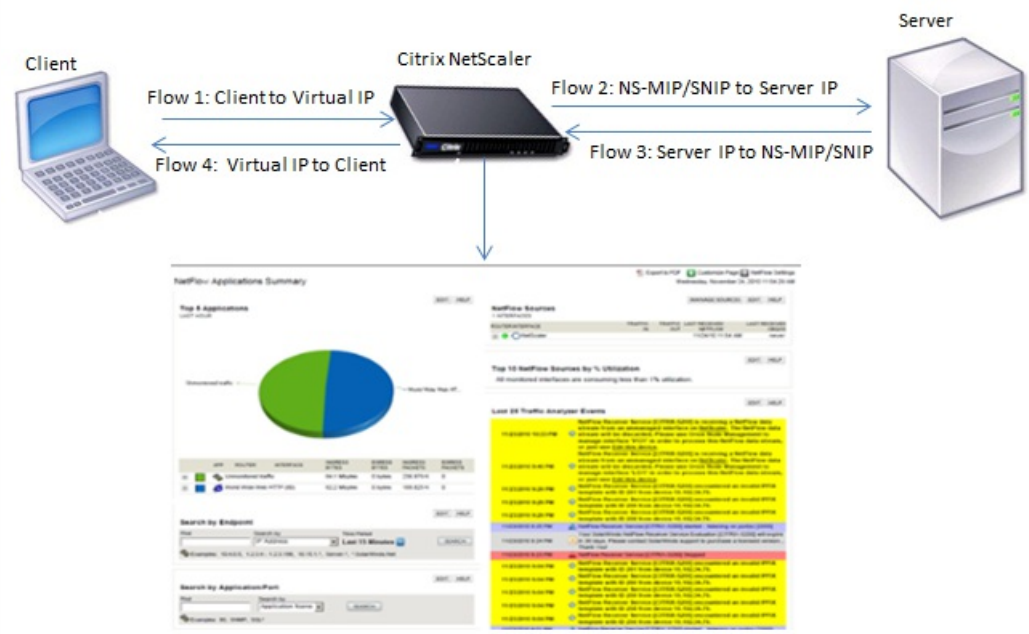
Updated: 2015-05-28

In the most common deployment scenario, inbound traffic flows to a Virtual IP address (VIP) on the NetScaler appliance and is load balanced to a server. Outbound traffic flows from the server to a mapped or subnet IP address on the NetScaler and from the VIP to the client. A flow is a unidirectional collection of IP packets identified by the following five tuples: sourceIP, sourcePort, destIP, destPort, and protocol.

The following figure describes how the AppFlow feature works.

Figure 1. NetScaler Flow Sequence

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Application-specific flow information captured by AppFlow

As shown in the figure, the network flow identifiers for each leg of a transaction depend on the direction of the traffic.

The different flows that form a flow record are:

Flow1: <Client-IP, Client-Port, VIP-IP, VIP-port, Protocol>

Flow2: <NS-MIP/SNIP, NS-port, Server-IP, Server-Port, Protocol>

Flow3: <Server-IP, Server-Port, NS-MIP/SNIP, NS-Port, Protocol>

Flow4: <VIP-IP, VIP-port, Client-IP, Client-Port, Protocol>

To help the collector link all four flows in a transaction, AppFlow adds a custom transactionID element to each flow. For application-level content switching, such as HTTP, it is possible for a single client TCP connection to be load balanced to different backend TCP connections for each request. AppFlow provides a set of records for each transaction.

This topic includes the following details:

- Flow Records
- Templates

Flow Records

Updated: 2013-08-20

AppFlow records contain standard NetFlow or IPFIX information, such as time stamps for the beginning and end of a flow, packet count, and byte count. AppFlow records also contain application-level information (such as HTTP URLs, HTTP request methods and response status codes, server response time, and latency), web page performance data (such as page load time, page render time, and time spent on the page), and database information (such as database protocol, database response status and database response size). IPFIX flow records are based on templates that need to be sent before sending flow records.

Templates

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppFlow defines a set of templates, one for each type of flow. Each template contains a set of standard Information Elements (IEs) and Enterprise-specific Information Elements (EIEs). IPFIX templates define the order and sizes of the Information Elements (IE) in the flow record. The templates are sent to the collectors at regular intervals, as described in RFC 5101.

A template can include the following EIEs:

transactionID

An unsigned 32-bit number identifying an application-level transaction. For HTTP, this corresponds to a request and response pair. All flow records that correspond to this request and response pair have the same transaction ID. In the most common case, there are four uniflow records that correspond to this transaction. If the NetScaler generates the response by itself (served from the integrated cache or by a security policy), there may be only two flow records for this transaction.

connectionID

An unsigned 32-bit number identifying a layer-4 connection (TCP or UDP). The NetScaler flows are usually bidirectional, with two separate flow records for each direction of the flow. This information element can be used to link the two flows. For the NetScaler, connectionID is an identifier for the connection data structure to track the progress of a connection. In an HTTP transaction, for instance, a given connectionID may have multiple transactionID elements corresponding to multiple requests that were made on that connection.

tcpRTT

The round trip time, in milliseconds, as measured on the TCP connection. This can be used as a metric to determine the client or server latency on the network.

httpRequestMethod

An 8-bit number indicating the HTTP method used in the transaction. An options template with the number-to-method mapping is sent along with the template.

httpRequestSize

An unsigned 32-bit number indicating the request payload size.

httpRequestURL

The HTTP URL requested by the client.

httpUserAgent

The source of incoming requests to the Web server.

httpResponseStatus

An unsigned 32-bit number indicating the response status code.

httpResponseSize

An unsigned 32-bit number indicating the response size.

httpResponseTimeToFirstByte

An unsigned 32-bit number indicating the time taken to receive the first byte of the response.

httpResponseTimeToLastByte

An unsigned 32-bit number indicating the time taken to receive the last byte of the response.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

flowFlags

An unsigned 64-bit flag used to indicate different flow conditions.

EIEs for web page performance data

clientInteractionStartTime

Time at which the browser receives the first byte of the response to load any objects of the page such as images, scripts, and stylesheets.

clientInteractionEndTime

Time at which the browser received the last byte of response to load all the objects of the page such as images, scripts, and stylesheets.

clientRenderStartTime

Time at which the browser starts to render the page.

clientRenderEndTime

Time at which browser finished rendering the entire page, including the embedded objects.

EIEs for database information

dbProtocolName

An unsigned 8-bit number indicating the database protocol. Valid values are 1 for MS SQL and 2 for MySQL.

dbReqType

An unsigned 8-bit number indicating the database request method used in the transaction. For MS SQL, valid values are 1 is for QUERY, 2 is for TRANSACTION, and 3 is for RPC. For valid values for MySQL, see the MySQL documentation.

dbReqString

Indicates the database request string without the header.

dbRespStatus

An unsigned 64-bit number indicating the status of the database response received from the web server.

dbRespLength

An unsigned 64-bit number indicating the response size.

dbRespStatString

The response status string received from the web server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the AppFlow Feature

Jul 02, 2015

You configure AppFlow in the same manner as most other policy-based features. First, you enable the AppFlow feature. Then you specify the collectors to which the flow records are sent. After that, you define actions, which are sets of configured collectors. Then you configure one or more policies and associate an action to each policy. The policy tells the NetScaler appliance to select requests the flow records of which are sent to the associated action. Finally, you bind each policy either globally or to specific vservers to put it into effect.

You can further set AppFlow parameters to specify the template refresh interval and to enable the exporting of httpURL, httpCookie, and httpReferer information. On each collector, you must specify the NetScaler IP address as the address of the exporter.
Note: For information about configuring the NetScaler as an exporter on the collector, see the documentation for the specific collector.
The configuration utility provides tools that help users define the policies and actions that determine exactly how the NetScaler appliance export records for a particular flow to a set of collectors(action.) The command line interface provides a corresponding set of CLI-based commands for experienced users who prefer a command line.

This topic includes the following details:

- Enabling AppFlow
- Specifying a Collector
- Configuring an AppFlow Action
- Configuring an AppFlow Policy
- Binding an AppFlow Policy
- Enabling AppFlow for Virtual Servers
- Enabling AppFlow for a Service
- Setting the AppFlow Parameters
- Example: Configuring AppFlow for DataStream

Enabling AppFlow

Updated: 2014-08-07

To be able to use the AppFlow feature, you must first enable it.

Note: AppFlow can be enabled only on nCore NetScaler appliances.

To enable the AppFlow feature by using the command line interface

At the command prompt, type one of the following commands:

enable ns feature AppFlow

To enable the AppFlow feature by using the configuration utility

Navigate to System > Settings, click Configure Advanced Features and select the AppFlow option.

Specifying a Collector

Updated: 2014-08-07

A collector receives flow records generated by the NetScaler appliance. To be able to send flow records, you must specify

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

at least one collector. You can specify up to four. However, you cannot export the same data to multiple collectors. You can remove unused collectors. By default, the collector listens to IPFIX messages on UDP port 4739. You can change the default port, when configuring the collector. Similarly, by default, NSIP is used as the source IP for appflow traffic. You can change this default source IP to a SNIP or MIP address when configuring a collector.

To specify a collector by using the command line interface

At the command prompt, type the following commands to add a collector and verify the configuration:

- add appflow collector <name> -IPAddress <ipaddress> -port <port_number> -netprofile <netprofile_name>
- show appflow collector <name>

Example

```
> add appflow collector col1 -IPaddress 10.102.29.251 -port 8000 -netprofile n2
```

To specify a collector by using the configuration utility

Navigate to System > AppFlow > Collectors, and create the AppFlow collector.

Configuring an AppFlow Action

Updated: 2014-08-07

An AppFlow action is a set collectors, to which the flow records are sent if the associated AppFlow policy matches.

To configure an AppFlow action by using the command line interface

At the command prompt, type the following commands to configure an Appflow action and verify the configuration:

- add appflow action <name> --collectors <string> ... [-clientSideMeasurements (Enabled | Disabled)] [-comment <string>]
- show appflow action

Example

```
> add appflow action apfl-act-collector-1-and-3 -collectors collector-1 collecter-3
```

To configure an AppFlow action by using the configuration utility

Navigate to System > AppFlow > Actions, and create the AppFlow action.

Configuring an AppFlow Policy

Updated: 2014-08-07

After you configure an AppFlow action, you must next configure an AppFlow policy. An AppFlow policy is based on a rule, which consists of one or more expressions.

Note: For creating and managing AppFlow policies, the configuration utility provides assistance that is not available at the command line interface.

To configure an AppFlow policy by using the command line interface

At the command prompt, type the following command to add an AppFlow policy and verify the configuration:

- add appflow policy <name> <rule> <action>
- show appflow policy <name>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

```
> add appflow policy apfl-pol-tcp-dsprt client.TCP.DSTPORT.EQ(22) apfl-act-collector-1-and-3
```

To configure an AppFlow policy by using the configuration utility

Navigate to System > AppFlow > Policies, and create the AppFlow policy.

To add an expression by using the Add Expression dialog box

1. In the Add Expression dialog box, in the first list box choose the first term for your expression.

   HTTP

   The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.

   SSL

   The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

   CLIENT

   The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

   When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.
2. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
3. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

Binding an AppFlow Policy

Updated: 2014-08-07

To put a policy into effect, you must bind it either globally, so that it applies to all traffic that flows through the NetScaler, or to a specific virtual server, so that the policy applies only to the traffic related to that virtual server.

When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the NetScaler operating system, policy priorities work in reverse order—the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you globally bind it. You can then add additional policies at any time without having to change the priority of an existing policy.

To globally bind an AppFlow policy by using the command line interface

At the command prompt, type the following command to globally bind an AppFlow policy and verify the configuration:

- bind appflow global <policyName> <priority> [<gotoPriorityExpression [-type <type>] [-invoke (<labelType> <labelName>)]
- show appflow global

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

```
bind appflow global af_policy_lb1_10.102.71.190 1 NEXT -type REQ_OVERRIDE -invoke vserver google
```

To bind an AppFlow policy to a specific virtual server by using the command line interface

At the command prompt, type the following command to bind an appflow policy to a specific virtual server and verify the configuration:
bind lb vserver <name> -policyname <policy_name> -priority <priority>
Example

```
bind lb vserver google -policyname af_policy_google_10.102.19.179 -priority 251
```

To globally bind an AppFlow policy by using the configuration utility

Navigate to System > AppFlow, click AppFlow policy Manager and select the relevant Bind Point (Default Global) and Connection Type, and then bind the AppFlow policy.

To bind an AppFlow policy to a specific virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers, select the virtual server, and click Policies, and bind the AppFlow policy.

Enabling AppFlow for Virtual Servers

Updated: 2014-08-12

If you want to monitor only the traffic through certain virtual servers, enable AppFlow specifically for those virtual servers. You can enable AppFlow for load balancing, content switching, cache redirection, SSL VPN, GSLB, and authentication virtual servers.

To enable AppFlow for a virtual server by using the command line interface

At the command prompt, type:

```
set cs vserver <name> <protocol> <IPAddress> <port> -appflowLog ENABLED
```

Example

```
> set cs vserver Vserver-CS-1 HTTP 10.102.29.161 80 -appflowLog ENABLED
```

To enable AppFlow for a virtual server by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, select the virtual server, and enable AppFlow Logging option.

Enabling AppFlow for a Service

Updated: 2014-08-07

You can enable AppFlow for services that are to be bound to the load balancing virtual servers.

To enable AppFlow for a service by using the command line interface

At the command prompt, type:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

set service <name> -appflowLog ENABLED
Example

```
set service ser -appflowLog ENABLED
```

To enable AppFlow for a service by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, select the service, and enable AppFlow Logging option.

Setting the AppFlow Parameters

Updated: 2014-08-08

You can set AppFlow parameters to customize the exporting of data to the collectors.

To set the AppFlow Parameters by using the command line interface

At the command prompt, type the following commands to set the AppFlow parameters and verify the settings:

- set appflow param [-templateRefresh <secs>] [-appnameRefresh <secs>] [-flowRecordInterval <secs>] [-udpPmtu <positive_integer>] [-httpUrl (**ENABLED** | **DISABLED**)] [-httpCookie (**ENABLED** | **DISABLED**)] [-httpReferer (**ENABLED** | **DISABLED**)] [-httpMethod (**ENABLED** | **DISABLED**)] [-httpHost (**ENABLED** | **DISABLED**)] [-httpUserAgent (**ENABLED** | **DISABLED**)] [-httpXForwardedFor (**ENABLED** | **DISABLED**)][-clientTrafficOnly (**YES** | **NO**)]
- show appflow Param

Example

```
> set appflow Param -templateRefresh 240 -udpPmtu 128 -httpUrl enabled
```

To set the AppFlow parameters by using the configuration utility

Navigate to System > AppFlow, click Change AppFlow Settings, and specify relevant AppFlow parameters.

Example: Configuring AppFlow for DataStream

Updated: 2013-08-20

The following example illustrates the procedure for configuring AppFlow for DataStream using the command line interface.

```
> enable feature appflow
> add db user sa password freebsd
> add lbvserver lb0 MSSQL 10.102.147.97 1433 -appflowLog ENABLED
> add service sv0 10.103.24.132 MSSQL 1433 -appflowLog ENABLED
> bind lbvserver lb0 sv0
> add appflow collector col0 -IPAddress 10.102.147.90
> add appflow action act0 -collectors col0
> add appflow policy pol0 "mssql.req.query.text.contains(\"select\")" act0
> bind lbvserver lb0 -policyName pol0 -priority 10
```

When the Netscaler appliance receives a database request, the appliance evaluates the request against a configured policy. If a match is found, the details are sent to the AppFlow collector configured in the policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exporting Performance Data of Web Pages to AppFlow Collector

May 28, 2015

The EdgeSight Monitoring application provides web page monitoring data with which you can monitor the performance of various Web applications served in a Netscaler environment. You can now export this data to AppFlow collectors to get an in-depth analysis of the web page applications. AppFlow, which is based on IPFIX standard, provides more specific information about web application performance than does EdgeSight monitoring alone.

You can configure both load balancing and content switching virtual servers to export EdgeSight Monitoring data to AppFlow collectors. Before configuring a virtual server for AppFlow export, associate an Appflow action with the EdgeSight Monitoring responder policy.

The following web page performance data is exported to AppFlow:

- **Page Load Time**. Elapsed time, in milliseconds, from when the browser starts to receive the first byte of a response until the user starts to interact with the page. At this stage, all the page content might not be loaded.
- **Page Render Time**.Elapsed time, in milliseconds, from when the browser receives the first byte of response until either all page content has been rendered or the page load action has timed out.
- **Time Spent on the Page**. Time spent by users on a page. Represents the period of time from one page request to the next one.

AppFlow transmits the performance data by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. The AppFlow templates use the following enterprise-specific Information Elements (EIEs) to export the information:

- **Client Load End Time**. Time at which the browser received the last byte of a response to load all the objects of the page such as images, scripts, and stylesheets.
- **Client Load Start Time**.Time at which the browser receives the first byte of the response to load any objects of the page such as images, scripts, and stylesheets.
- **Client Render End Time**. Time at which browser finished rendering the entire page, including the embedded objects.
- **Client Render Start Time**. Time at which the browser started rendering the page.

This topic includes the following details:

- Prerequisites for Exporting Performance Data of Web Pages to AppFlow Collectors
- Associating an AppFlow Action with the EdgeSight Monitoring Responder Policy

Prerequisites for Exporting Performance Data of Web Pages to AppFlow Collectors

Updated: 2013-09-13

Before associating the AppFlow action with the AppFlow policy, verify that the following prerequisites have been met:

- The AppFlow feature has been enabled and configured. For instructions, see "Configuring the AppFlow feature".
- The Responder feature has been enabled. For instructions, see "Enabling a Responder Feature".
- The EdgeSight Monitoring feature has been enabled. For instructions, see "Enabling an Application for EdgeSight Monitoring."
- EdgeSight Monitoring has been enabled on the load balancing or content switching virtual servers bound to the services of applications for which you want to collect the performance data. For instructions, see "Enabling an Application for

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

EdgeSight Monitoring."

Associating an AppFlow Action with the EdgeSight Monitoring Responder Policy

Updated: 2013-10-31

To export the web page performance data to the AppFlow collector, you must associate an AppFlow action with the EdgeSight Monitoring responder policy. An AppFlow action specifies which set of collectors receive the traffic.

To associate an AppFlow action with the EdgeSight Monitoring Responder policy by using the command line interface

At the command prompt, type:

```
set responder policy <name> -appflowAction <action_Name>
```

Example

```
set responder policy pol -appflowAction actn
```

To associate an AppFlow action with the EdgeSight Monitoring Responder policy by using the configuration utility

1. Navigate to AppExpert > Responder > Policies.
2. In the details pane, select an EdgeSight Monitoring responder policy, and then click **Open**.
3. In the **Configure Responder Policy** dialog box, in the **AppFlow Action** drop-down list, select the AppFlow action associated with the collectors to which you want to send the web-page performance data.
4. Click **OK**.

Configuring a Virtual Server to Export EdgeSight Statistics to Appflow Collectors

To export EdgeSight statistics information from a virtual server to the AppFlow collector, you must associate an AppFlow action with the virtual server.

To associate an AppFlow action with a Load Balancing or Content Switching virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers or Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, select a virtual server, or multiple virtual servers, and then click Enable EdgeSight Monitoring.
3. In the Enable EdgeSight Monitoring dialog box, select the Export EdgeSight statistics to Appflow check box.
4. From the Appflow Action drop-down list, select the AppFlow action. The AppFlow action defines the list of AppFlow collectors to which it exports EdgeSight Monitoring statistics. If you have selected multiple load balancing virtual servers, the same AppFlow Action will be associated with the responder policies bound to them. You can later change the AppFlow Action configured for each of the selected Load Balancing virtual server individually, if required.
5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AutoScale

Aug 26, 2013

Efficient hosting of applications in a cloud requires continuous optimization of application availability. To meet increasing demand, you have to scale network resources upward. When demand subsides, you need to scale down to avoid the unnecessary cost of idle resources. To minimize the cost of running the application by deploying only as many instances as are necessary during any given period of time, you have to constantly monitor traffic. However, monitoring traffic manually is not a feasible option. For the application environment to be able to scale up or down rapidly, you need to automate the processes of monitoring traffic and of scaling resources up and down whenever necessary.

If your organization uses Citrix CloudPlatform to deploy and manage the cloud environment, a Citrix NetScaler appliance can automatically scale users' applications as needed. The CloudPlatform elastic load balancing feature includes a feature called *AutoScale*. A CloudPlatform user can use the AutoScale feature to specify thresholds for various conditions for automatically scaling the application fleet upward and downward. The scale-up and scale-down conditions can vary from simple use cases, such as a server's CPU usage, to complex use cases, such as a combination of a server's CPU usage and responsiveness. CloudPlatform, in turn, configures the NetScaler appliance to load balance traffic to the application virtual machines (VMs), monitor application thresholds and performance, and trigger scale-up and scale-down actions to add or remove VMs to or from the application fleet.

The CloudPlatform user performs all AutoScale configuration tasks by using the CloudPlatform user interface or APIs. The CloudPlatform user:

1. Creates a load balancing rule, with the necessary load balancing algorithm and stickiness.
2. Configures AutoScale parameters by specifying the application instance template, the minimum number of instances to maintain, the maximum number of instances permitted, scale-up and scale-down policies, and other information necessary for the functioning of the feature.
3. Submits the configuration.

For information about configuring a load balancing rule and AutoScale, see *Citrix CloudPlatform 3.0.5 (powered by Apache CloudStack) Administrator's Guide*, at http://support.citrix.com/article/CTX134823.

When the CloudPlatform user completes the AutoScale configuration, CloudPlatform uses the NetScaler NITRO API to push all the necessary configuration commands to the NetScaler appliance. As the NetScaler administrator, you do not have to perform any tasks for configuring AutoScale on the NetScaler appliance. However, you might have to be aware of certain prerequisites, and you might have to troubleshoot the configuration if issues arise in the AutoScale configuration. To troubleshoot the configuration, you have to be aware of how CloudPlatform works and what configuration CloudPlatform pushes to the NetScaler appliance. You also need a working knowledge of how to troubleshoot issues on a NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

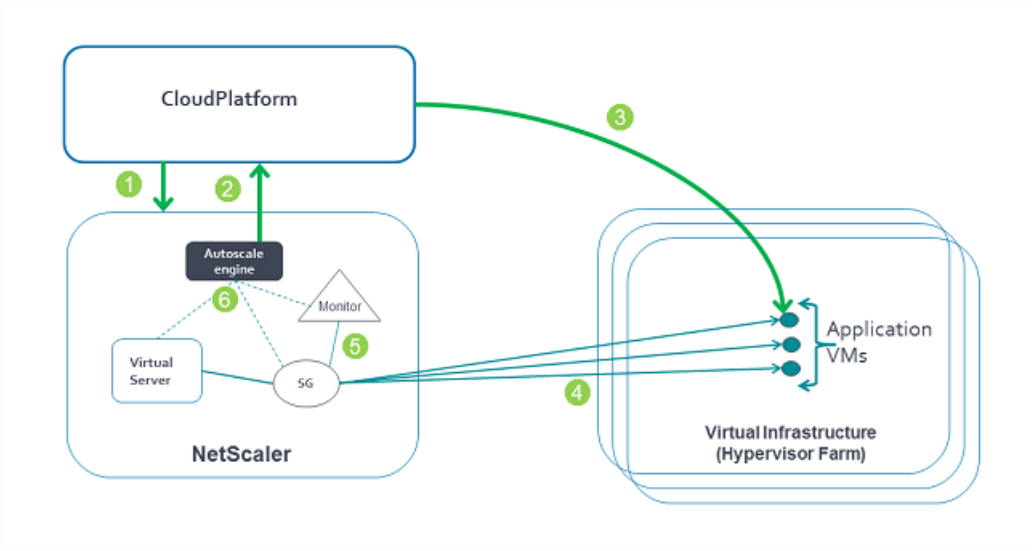
How AutoScale Works

Sep 06, 2013

When the CloudPlatform user completes the AutoScale configuration, CloudPlatform uses the NetScaler NITRO API to create an AutoScale-related configuration on the NetScaler appliance. For information about the configuration commands that CloudPlatform uses to configure the NetScaler appliance, see "NetScaler Configuration Details."

The following diagram shows the sequence of operations, beginning with CloudPlatform pushing the AutoScale configuration to the NetScaler appliance. The events are numbered in the order in which they occur, and are described below.

Figure 1. AutoScale Architecture



When the CloudPlatform user submits the AutoScale configuration, the following events occur:

1. CloudPlatform uses the NetScaler NITRO API to push the AutoScale configuration to the NetScaler appliance, creating AutoScale-related entities on the appliance. The entities include a load balancing virtual server, a service group, and monitors.
2. The AutoScale engine on the NetScaler appliance sends API requests to CloudPlatform to initially deploy the minimum number of virtual machines required.
3. CloudPlatform provisions the minimum number of instances (VMs) on the hypervisors (virtualization hosts) that it manages.
4. The NetScaler appliance discovers the IP addresses assigned by CloudPlatform to the newly created VMs and binds them, as services, to the service group representing them. The NetScaler appliance can then load balance traffic to the VMs.
5. NetScaler monitors bound to the service group start monitoring the load by collecting SNMP metrics from the instances.
6. The AutoScale engine on the NetScaler appliance monitors the metrics collected from the VMs and triggers scale-up and scale-down events whenever the metrics breach the configured threshold for the specified period. As part of the scale-up trigger, the NetScaler AutoScale engine sends an API request to CloudPlatform to deploy a new VM. After the virtual machine is deployed, the AutoScale engine binds the service representing the VM (IP address and port) to the service group and, after the configured quiet time, starts forwarding load balanced traffic to the new virtual machine. Likewise, as part of the scale-down trigger, the NetScaler AutoScale engine selects a VM, stops forwarding new requests to that

© 1999-2017 Citrix Systems, Inc. All rights reserved.

instance, and waits for the configured quiet time (to allow for the processing of current requests to complete) before it sends an API request to CloudPlatform to destroy the chosen instance.

In this way, the NetScaler appliance monitors the application and triggers scale-up and scale-down events on the basis of application load and/or performance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Supported Environment

Sep 30, 2013

AutoScale is supported in the following environment:

- Citrix CloudPlatform 3.0.5.
- Citrix NetScaler MPX/SDX/virtual appliance running Citrix NetScaler release 10.e and later.
- SNMP v1/v2.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Prerequisites

Sep 06, 2013

Before you set up AutoScale, do the following:

- Make sure that CloudPlatform is reachable from the NetScaler appliance. You can do so by logging on to the NetScaler appliance and sending ping requests to the CloudPlatform server's IP address.
- Make sure that the network service offering used in CloudPlatform includes the NetScaler appliance as an external load balancing device.
- Use a CloudPlatform and NetScaler release that supports AutoScale. For information about NetScaler releases that support AutoScale, see "Supported Environment."

If you have to troubleshoot an AutoScale setup, you also have to know the prerequisites for setting up AutoScale in CloudPlatform. See the "Prerequisites" section of "Configuring AutoScale" in the *Citrix CloudPlatform 3.0.5 (powered by Apache CloudStack) Administrator's Guide*, at http://support.citrix.com/article/CTX134823.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler Configuration Details

Mar 19, 2012

The following table describes the AutoScale configuration commands that are used by Citrix CloudPlatform to configure a NetScaler appliance.

Table 1. NetScaler Configuration for AutoScale

AutoScale configuration command(s)	Description
add lb vserver Cloud-VirtualServer-192.0.2.116-22 TCP 192.0.2.116 22 -persistenceType NONE -lbMethod ROUNDROBIN -cltTimeout 9000 -minAutoscaleMembers 2 -maxAutoscaleMembers 5	Creates a load balancing virtual server to evenly distribute the load on the application instances (ROUND-ROBIN method). The virtual server also specifies the limits for the number of instances to which the application can scale up or down (maxAutoscaleMembers and minAutoscaleMembers, respectively).
add serviceGroup Clouda35a6b6b76614006b97476e841b80f79 TCP -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -cltTimeout 9000 -svrTimeout 9000 -CKA NO -TCPB NO -CMP NO -autoScale POLICY -memberPort 22 bind lb vserver Cloud-VirtualServer-192.0.2.116-22 Clouda35a6b6b76614006b97476e841b80f79	Creates an AutoScale service group for the application instances, with the service group's autoScale parameter set to POLICY. Also specifies the port on which the service group members must receive traffic. The second command binds the service group to the load balancing virtual server.
add server autoscale-internal_server_Clouda35a6b6b76614006b97476e841b80f79 autoscale-internal_server_Clouda35a6b6b76614006b97476e841b80f79 bind serviceGroup Clouda35a6b6b76614006b97476e841b80f79 autoscale-internal_server_Clouda35a6b6b76614006b97476e841b80f79 22	Creates a server entry to represent the application instances. Binds the server entry to the service group.
add lb metricTable Cloud-MTbl-192.0.2.116-22 bind lb metricTable Cloud-MTbl-192.0.2.116-22 Linux_User_CPU_-_percentage 1.3.6.1.4.1.2021.11.9.0	Configures a new SNMP monitor to retrieve the specified

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AutoScale configuration command(s)	Description
add lb monitor Cloud-Mon-192.0.2.116-22 LOAD -interval 24 -destPort 161 -snmpCommunity public -metricTable Cloud-MTbl-192.0.2.116-22 bind lb monitor Cloud-Mon-192.0.2.116-22 -metric Linux_User_CPU_-_percentage -metricThreshold 2147483647 bind serviceGroup Clouda35a6b6b76614006b97476e841b80f79 -monitorName Cloud-Mon-192.0.2.116-22 -passive	metrics.
add autoscale profile Cloud-AutoScale-Profile-192.0.2.116-22 -type CLOUDSTACK -url "http://10.102.31.107:8080/client/api" -apiKey t0fEWPtk_ncQYbofjAm1jjlgGTR7UNZrkZ3sdEpLREBNzBPLSNpNz8qNSbc439xNtYnEYdWn_MsUC_CUazaIKg -sharedSecret -PrE5h3DP7swHAN12TGBlX-xSTRLHzob91l6O0VO1FMxvE1UOl7uoD6_Z0bkkLaVtK5Y10oBkTzgbTwp3u5lCQ	Creates an AutoScale profile to specify the details required by NetScaler for making API requests to CloudPlatform (URL, API key, and shared secret).
add autoscale action Cloud-AutoScale-ScaleUpAction-192.0.2.116-22 -type SCALE_UP -profileName Cloud-AutoScale-Profile-192.0.2.116-22 -parameters "command=deployVirtualMachine&zoneid=2ab23590-78cb-4106-8d85-4412a2f2435f&serviceofferingid=b9503e47-0d8f-4c89-a88d-04d8b17fe8e9&templateid=1a4a5084-208c-47a8-9c16-d582550cf759&displayname=AutoScale-LB-lb&networkids=a3c97129-b729-4c72-994f-7b918f20ce4d&lbruleid=f96b7f3b-19ec-4123-891c-604f05b032b3" -quietTime 90 -vServer Cloud-VirtualServer-192.0.2.116-22	Creates a scale-up action, which enables the NetScaler appliance to add virtual machines (instances) to the application fleet.
add autoscale action Cloud-AutoScale-ScaleDownAction-192.0.2.116-22 -type SCALE_DOWN -profileName Cloud-AutoScale-Profile-192.0.2.116-22 -parameters "command=destroyVirtualMachine&lbruleid=f96b7f3b-19ec-4123-891c-604f05b032b3" -vmDestroyGracePeriod 30 -quietTime 90 -vServer Cloud-VirtualServer-192.0.2.116-22	Creates a scale-down action, which enables the NetScaler appliance to remove virtual machines (instances) from the application fleet.
add autoscale policy Cloud-AutoScale-Policy-Min-192.0.2.116-22 -rule "SYS.VSERVER(\"Cloud-VirtualServer-192.0.2.116-22\").ACTIVESERVICES.LT(SYS.VSERVER(\"Cloud-VirtualServer-192.0.2.116-22\").MINAUTOSCALEMEMBERS)" -action Cloud-AutoScale-ScaleUpAction-192.0.2.116-22	Creates an AutoScale policy to initially create the specified minimum number of VMs and, later, to ensure that the number of VMs in the fleet does not fall below the required minimum.
add autoscale policy Cloud-AutoScale-Policy-Max-192.0.2.116-22 -rule "SYS.VSERVER(\"Cloud-VirtualServer-192.0.2.116-22\").ACTIVESERVICES.GT(SYS.VSERVER(\"Cloud-VirtualServer-192.0.2.116-22\").MAXAUTOSCALEMEMBERS)" -action Cloud-AutoScale-ScaleDownAction-192.0.2.116-22	Creates an AutoScale policy to prevent the number of VMs in the fleet from exceeding the specified maximum.
add autoscale policy Cloud-AutoScale-Policy-192.0.2.116-22-35 -rule "SYS.VSERVER(\"Cloud-VirtualServer-192.0.2.116-22\").ACTIVESERVICES.LT(SYS.VSERVER(\"Cloud-VirtualServer-192.0.2.116-22\").MAXAUTOSCALEMEMBERS) && (SYS.VSERVER(\"Cloud-VirtualServer-192.0.2.116-22\").SNMP_TABLE(0).AVERAGE_VALUE.GT(90))" -action Cloud-AutoScale-ScaleUpAction-192.0.2.116-22	Creates an AutoScale policy to evaluate the metrics that are collected and trigger a

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AutoScale configuration command(s)	Description
	scale-up action when the metric value breaches the threshold specified for the scale-up policy.
add autoscale policy Cloud-AutoScale-Policy-192.0.2.116-22-36 -rule "SYS.VSERVER(\"Cloud-VirtualServer-192.0.2.116-22\").ACTIVESERVICES.GT(SYS.VSERVER(\"Cloud-VirtualServer-192.0.2.116-22\").MINAUTOSCALEMEMBERS) && (SYS.VSERVER(\"Cloud-VirtualServer-192.0.2.116-22\").SNMP_TABLE(0).AVERAGE_VALUE.LT(30))" -action Cloud-AutoScale-ScaleDownAction-192.0.2.116-22	Creates an AutoScale policy to evaluate the collected metrics and trigger a scale-down action when the metric value breaches the threshold specified by the scale-down policy.
add ns timer Cloud-AutoScale-Timer-192.0.2.116-22 -interval 30 bind ns timer Cloud-AutoScale-Timer-192.0.2.116-22 -policyName Cloud-AutoScale-Policy-Min-192.0.2.116-22 -priority 1 -gotoPriorityExpression END -sampleSize 1 -threshold 1 bind ns timer Cloud-AutoScale-Timer-192.0.2.116-22 -policyName Cloud-AutoScale-Policy-Max-192.0.2.116-22 -priority 2 -gotoPriorityExpression END -sampleSize 1 -threshold 1 bind ns timer Cloud-AutoScale-Timer-192.0.2.116-22 -policyName Cloud-AutoScale-Policy-192.0.2.116-22-35 -priority 3 -gotoPriorityExpression END -sampleSize 2 -threshold 2 bind ns timer Cloud-AutoScale-Timer-192.0.2.116-22 -policyName Cloud-AutoScale-Policy-192.0.2.116-22-36 -priority 4 -gotoPriorityExpression END -sampleSize 2 -threshold 2	Creates a timer that enables evaluation of the AutoScale policies at the configured sampling intervals.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Sep 06, 2013

Before you attempt to resolve an AutoScale issue, make sure that the prerequisites have been adhered to, on both the CloudPlatform server and the NetScaler appliance, as described in "Prerequisites." If that does not resolve the issue, your problem could be one of the following.

The AutoScale configuration was successfully configured in CloudPlatform. Yet, the minimum number of VMs has not been created.

- Recommend that the CloudPlatform user deploy one VM manually in the network before configuring AutoScale. Ask the user to remove the AutoScale configuration from the NetScaler appliance or the load balancer from the network, manually deploy one VM (preferably using the template created for the AutoScale configuration), and then create the AutoScale configuration.
- Verify that the CloudPlatform user has configured the VM template in such a way that the VMs that are created from the template can accept traffic without manual intervention. If a provisioned VM cannot accept traffic automatically, the metric remains above the threshold, and the AutoScale configuration continues to provision additional VMs, as designed. To remedy the issue, disable AutoScale from CloudPlatform, fix the template, and then enable AutoScale.
- Verify that the CloudPlatform user has not exceeded the limit for the number of VMs imposed by the user's account.
- Verify that the CloudPlatform server is up and is reachable from the NetScaler appliance.
- Verify that the CloudPlatform log file, management-server.log, has reported the successful creation of the AutoScale configuration in CloudPlatform.
- Verify that the scale-up policy that is responsible for initial scale up (the policy name is prefixed with Cloud-AutoScale-Policy-Min) is receiving hits.

The AutoScale configuration is rapidly spawning a large number of VMs

- Verify that the CloudPlatform user has configured the VM template in such a way that the VMs that are created from the template can accept traffic without manual intervention. If a provisioned VM cannot accept traffic automatically, the metric remains above the threshold, and the AutoScale configuration continues to provision additional VMs, as designed. To remedy the issue, disable AutoScale from CloudPlatform, fix the template, and then enable AutoScale.
- Verify that the quiet time that the CloudPlatform user has configured in the AutoScale configuration is sufficient to ensure even traffic distribution to all the VMs, including the new VM. If the quiet time is too low, and traffic distribution has not stabilized, the metrics might remain above the threshold, and additional VMs might be spawned.

When I ran the top command on my VM, I noticed that the CPU usage on my VM had breached the threshold that was configured for the scale-up action in AutoScale. Yet, the application is not scaling up.

- Verify that the CloudPlatform user has installed an SNMP agent in the VM template, and that the SNMP agent is up and running on every VM.
- Verify that the CloudPlatform user has not exceeded the limit for the number of VMs imposed by the user's account.
- Verify that the CloudPlatform user has correctly configured the SNMP parameters to collect metrics from the VM (for example, the community string and the port).
- Verify that the scale-up or scale-down policy is receiving hits.
- Verify that the CloudPlatform server is up, and that the CloudPlatform server is reachable from the NetScaler appliance.

One or more additional VMs have been created, but they are not accepting traffic (that is, VMs have been created, but the average value of the metrics is still above the threshold)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Verify that the user has configured the templates in such a way that the VMs created from the templates can start serving traffic without any manual intervention.
- Verify that the service is running on the VMs, on the configured member port.
- Send a ping request to the gateway (virtual router), from the VM that is not accepting traffic.

The AutoScale configuration has been deleted, but the VMs continue to exist

- The VMs might not be deleted immediately after the AutoScale configuration is deleted. Wait for about 5 minutes after you have deleted the AutoScale configuration, and then check again.
- If the destruction of VMs has not commenced after 5 minutes, you might have to delete the VMs manually.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Clustering

Dec 07, 2016

A NetScaler cluster is a group of nCore appliances working together as a single system image. Each appliance of the cluster is called a node. The cluster can have one appliance or as many as 32 NetScaler nCore hardware or virtual appliances as nodes.

The client traffic is distributed between the nodes to provide high availability, high throughput, and scalability.

To create a cluster, you must add the appliances as cluster nodes, set up communication between the nodes, set up links to the client and server networks, configure the appliances, and configure the distribution of client and server traffic.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler Configuration Support in a Cluster

Jan 05, 2017

Clustering of NetScaler appliances is supported from NetScaler 10 onwards.

The following table lists the NetScaler configurations that are **not** supported in NetScaler 10 and also gives the status of the features in subsequent releases. For the features that are supported since the first cluster release, click here.

> **Important**
>
> - Unless specified otherwise in this table, all NetScaler features are supported in a cluster.
>
> - The "Node-level" entry in the table indicates that the feature is supported only on individual cluster nodes.

Configuration	**10**	**10.1**	**10.5**	**11.0**
SSL (classic policies) Note: SSL - advanced policies are supported from NetScaler 10 onwards.	No	No	No	No
SSL FIPS	No	No	No	No
SSL Certificate Bundle	No	No	No	No
Content switching actions Note: The content switching feature is supported from NetScaler 10.1 onwards.	No	Yes	Yes	Yes
Policy-based logging for content switching policies	No	Yes	Yes	Yes
Rate limiting	No	Yes	Yes	Yes
Action analytics	No	Yes	Yes	Yes
Branch Repeater load balancing	No	Yes	Yes	Yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GSLB	No	No	Yes	Yes
RTSP	No	No	No	No
DNSSEC	No	No	No	No
DNS64	No	No	No	No
FTP	No	No	No	Yes Note: Supported from NetScaler 11.0 Build 62.x onwards.
TFTP	No	No	No	No
Connection mirroring	No	No	No	No
Integrated caching	Node-Level	Node-level	Node-level	Node-level
Large shared cache	No	Node-level	Node-level	Node-level
Application firewall	No	No	Node-level	Yes
HTTP Denial-of-Service Protection (HDOSP)	Node-level	Node-level	Node-level	Node-level
Priority queuing (PQ)	Node-level	Node-level	Node-level	Node-level
Sure connect (SC)	Node-level	Node-level	Node-level	Node-level
AppQoE	NA	Node-level	Yes	Yes
Surge protection	Node-level	Node-level	Node-level	Node-level

© 1999-2017 Citrix Systems, Inc. All rights reserved.

MPTCP	No	No	Yes	Yes
Striped SNIPs	Yes	Yes	Yes	Yes Note: Supported in L2 clusters. Not supported in L3 clusters.
MSR	Yes	Yes	Yes	Yes Note: Supported in L2 clusters. Not supported in L3 clusters.
IS-IS (IPv4 and IPv6)	No	Yes	Yes	Yes
Jumbo Frames	No	No	Yes	Yes Note: Supported in L2 clusters. Not supported in L3 clusters.
IP-IP tunneling	No	Yes	Yes	Yes
Link load balancing	No	No	Yes	Yes
FIS (Failover Interface Set)	No	No	Yes	Yes
Link redundancy (LR)	No	No	Yes	Yes
NAT46	No	No	No	No
NAT64	No	No	No	No
LSN/CGNAT	No	No	No	No
v6 ReadyLogo	No	No	No	No
Traffic domains	No	No	Yes	Yes Note: Supported in L2

© 1999-2017 Citrix Systems, Inc. All rights reserved.

				clusters. Not supported in L3 clusters.
Route monitor	No	No	No	No
GRE tunneling (CB)	No	No	No	No
Layer 2 mode	No	No	Yes	Yes
Net profiles	No	No	Yes	Yes
HTTPS callout	No	Yes	Yes	Yes
AAA-TM	No	Node-level	Node-level	Node-level
AppFlow	No	Node-level	Node-level	Node-level
Insight	No	No	No	No
HDX Insight	No	No	No	No
VMAC/VRRP	No	No	Yes	Yes
NetScaler Push	No	No	No	No
Stateful Connection Failover	No	No	No	No
Graceful Shutdown	No	No	No	No
DBS AutoScale	No	No	No	No
DSR using TOS	No	No	No	No
Finer Startup-RR Control	Node-level	Node-level	Node-level	Node-level
XML XSM	No	No	No	No

© 1999-2017 Citrix Systems, Inc. All rights reserved.

DHCP RA	No	No	No	No
Bridge Group	No	No	Yes Note: Supported from NetScaler 10.5 Build 52.1115.e onwards.	Yes
Network Bridge	No	No	No	No
Web Interface on NetScaler (WIonNS)	No	No	No	Yes (check FAQ) Note: Supported from NetScaler 11.0 Build 62.x onwards.
EdgeSight Monitoring	No	No	No	No
Metrics tables - Local	No	No	No	No
DNS Caching	Node-level	Node-level	Node-level	Node-level
Call Home	Node-level	Node-level	Node-level	Node-level
NetScaler Gateway ICA Proxy mode	No	No	Node-level	Yes
NetScaler Gateway (SSL VPN / full VPN and CVPN)	No	No	Node-level	Node-level
CloudBridge Connector	No	No	No	No

NetScaler configurations supported from NetScaler 10 onwards

Load balancing, load balancing persistency, DNS load balancing, SIP, maxClient, Spillover (connection and dynamic), Spillover based on bandwidth, DataStream, Compression control, Content filtering, TCP buffering, Cache redirection, Distributed Denial-of-Service (DDoS), Client Keep-alive, Basic networking (IPv4 and IPv6), OSPF (IPv4 and IPv6), RIP (IPv4 and IPv6), RIP (IPv4 and IPv6), VLAN, ICMP, Fragmentation, MBF, ACL, Simple ACL, PBR, MSR, Path MTU discovery, IP-IP, SNMP, Policies (classic and advanced), Rewrite, Responder, HTTP callout, Web server logging, Audit logging (nslog and syslog), USIP, Location

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

commands, HTML injection, NITRO API, AppExpert, KRPC.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Prerequisites for Cluster Nodes

Feb 02, 2016

NetScaler appliances that are to be added to a cluster must satisfy the following criteria:

- All appliances must have the same software version and build.
- All appliances must be of the same platform type. This means that a cluster must have either all hardware appliances (MPX) or virtual appliances (VPX) or SDX NetScaler instances.

  Note:
  - For a cluster of hardware appliances (MPX), the appliances must be of the same model type.
  - For a cluster of virtual appliances (VPX), the appliances must be deployed on the following hypervisors: XenServer, Hyper-V, VMware ESX, and KVM.
  - Clusters of SDX NetScaler instances are supported in NetScaler 10.1 and later releases. To create a cluster of SDX NetScaler instances, see "Setting up a Cluster of NetScaler Instances".
  - Jumbo frames are not supported on a NetScaler cluster that is made up of NetScaler SDX instances.
  - You cannot create L3 clusters of SDX instances.
- [For releases prior to NetScaler 11.0] All appliances must be on the same network. In NetScaler 11.0 and later releases, appliances can belong to different networks.
- All appliances must have the same licenses. Also, depending on the NetScaler version, there are some additional aspects to address:
  - For releases prior to NetScaler 10.5 Build 52.x:
    - A separate cluster license file is required. This file must be copied to the /nsconfig/license/ directory of the configuration coordinator.
    - Because of the separate cluster license file, the cluster feature is available irrespective of the NetScaler license.
  - For releases after NetScaler 10.5 Build 52.x:
    - No separate cluster license is required.
    - Cluster is licensed with the Enterprise and Platinum licenses. Cluster is not available for Standard license.
- Be initially configured and connected to a common client-side and server-side network.

Note: For a cluster of virtual appliances, that has large configurations, it is recommended to use 6 GB RAM for each node of the cluster.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

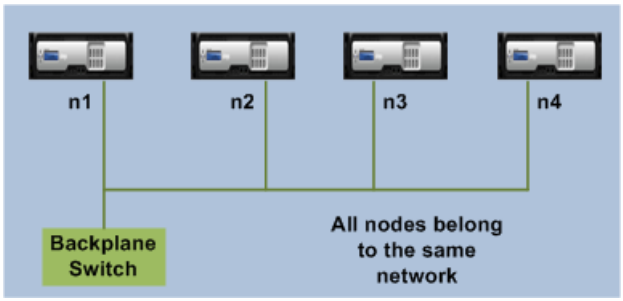
Cluster Overview

Jun 17, 2015

A NetScaler cluster is formed by grouping NetScaler appliances together. Based on the network location of the NetScaler appliances that you intend adding to the cluster, you must be aware of the following cluster setups:

Note: Unless specified otherwise, cluster features and configurations are the same for L2 and L3 clusters.

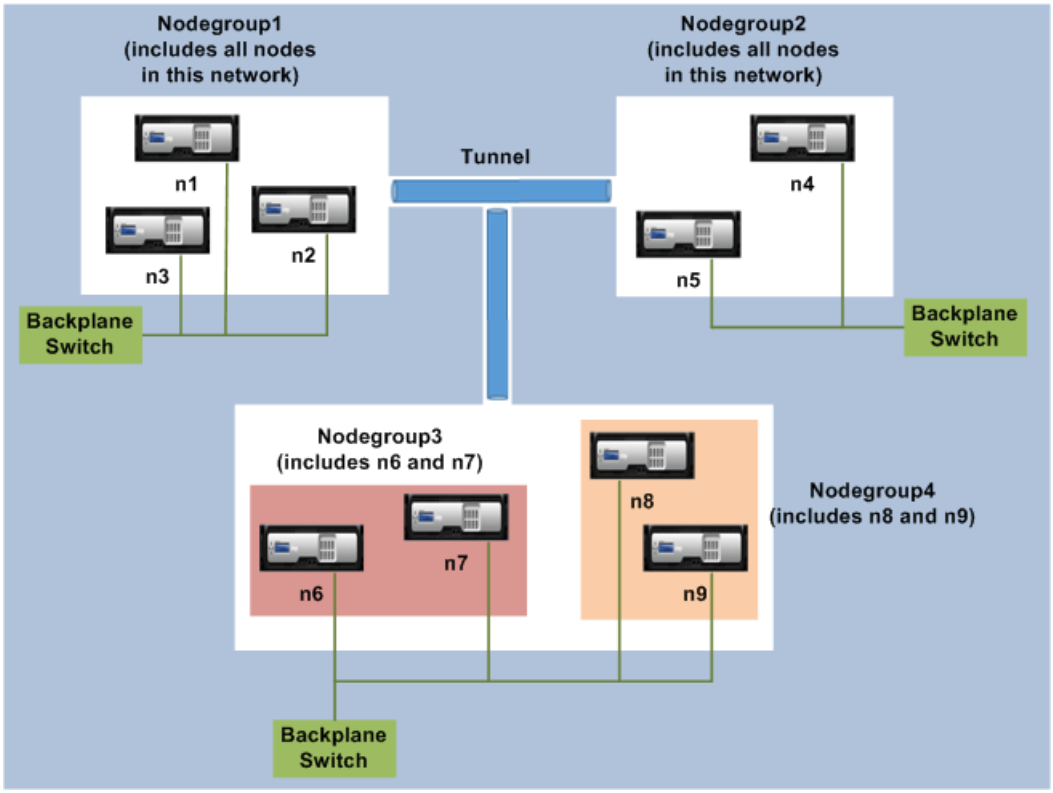
- **L2 cluster:** In this cluster deployment, all cluster nodes belong to the same network.



- **L3 cluster (also referred to as 'cluster in INC mode'):** In this cluster deployment, cluster nodes can belong to different networks. The cluster nodes from a specific network must be grouped into nodegroups that include only nodes from that network. From the following figure, we see that nodes n1, n2, n3 are in the same network and are grouped into Nodegroup1.

  Similarly, the case for nodes n4 and n5, that are grouped in Nodegroup2. In the third network, there are two nodegroups. Nodegroup3 includes n6 and n7 and Nodegroup4 includes n8 and n9.

  Note: Supported from NetScaler 11.0 onwards.

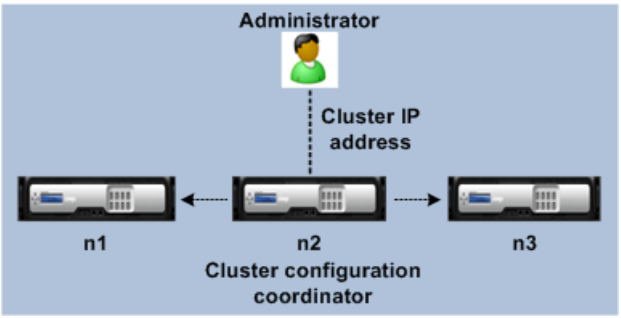


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Synchronization Across Cluster Nodes

May 25, 2015

All configurations on a NetScaler cluster are performed on the cluster IP address, which is the management address of the cluster. This cluster IP address is owned by a cluster node that is referred to as the cluster configuration coordinator as shown in the following figure:



The configurations that are available on the configuration coordinator are automatically propagated to the other cluster nodes and therefore all cluster nodes have the same configurations.

Note:

- NetScaler allows only a few configurations to be performed on individual cluster nodes through their NetScaler IP (NSIP) address. In these cases, you must ensure configuration consistency manually across all nodes in the cluster. These configurations are not propagated across the other cluster nodes. For more information on operations supported on each cluster nodes, see "Operations Supported on Individual Cluster Nodes".
- The following commands when executed on the cluster IP address are not propagated to other cluster nodes:
  - shutdown: Shuts down only the configuration coordinator.
  - reboot: Reboots only the configuration coordinator.
  - rm cluster instance: Removes the cluster instance from the node that you are executing the command on.
- For a command to propagate to other cluster nodes:
  - The quorum must be configured on cluster instance.
  - The majority of the cluster quorum with (n/2 + 1) of the cluster nodes must be active for the cluster to be operational.
  - A cluster can run with a minimum number of nodes when majority rule (n/2 + 1) is relaxed.

When a node is added to a cluster, the configurations and the files (SSL certificates, licenses, DNS, and so on) that are available on the cluster configuration coordinator are synchronized to the newly added cluster node. When an existing cluster node, that was intentionally disabled or that had failed, is once again added, the cluster compares the configurations available on the node with the configurations available on the configuration coordinator. If there is a mismatch in configurations, the node is synchronized by using one of the following:

- **Full synchronization.** If the difference between configurations exceeds 255 commands, all the configurations of the configuration coordinator are applied to the node that is rejoining the cluster. The node remains operationally unavailable for the duration of the synchronization.
- **Incremental Synchronization.** If the difference between configurations is less than or equal to 255 commands, only the configurations that are not available are applied to the node that is rejoining the cluster. The operational state of the node remains unaffected.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: You can also manually synchronize the configurations and files. For more information, see "Synchronizing Cluster Configurations" and "Synchronizing Cluster Files".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Striped, Partially Striped, and Spotted Configurations

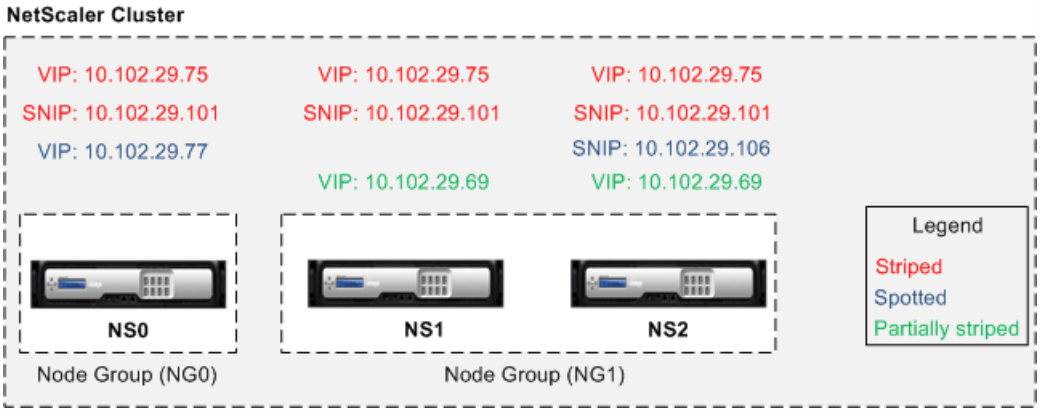
May 25, 2015

By virtue of command propagation, all nodes in a cluster have the same configurations. However, you may want some configurations to be available only on certain cluster nodes. While you cannot restrict the nodes on which the configurations are available, you can specify the nodes on which the configurations are active.

For example, you can define a SNIP address to be active on only one node, or define a SNIP address to be active on all nodes, or define a VIP address to be active on only one node, or define a VIP address to be active on all nodes, or define a VIP address to be active only on two nodes of a 3-node cluster.

Depending on the number of nodes the configurations are active on, cluster configurations are referred to as striped, partially striped, or spotted configurations.

Figure 1. Three-node cluster with striped, partially striped, and spotted configurations



The following table provides more details on the types of configurations:

Configuration Type	Active on...	Applicable to...	Configurations...
Striped configuration	All the cluster nodes	All entities	No specific configuration required to make an entity striped. By default, all entities defined on a cluster IP address are striped on all the cluster nodes.
Partially striped configuration	A subset of cluster nodes	Refer "Node Groups"	Bind the entities that you want to be partially striped, to a node group. The configuration will be active only on the cluster nodes that belong to the node group.
Spotted configuration	Single cluster node	• SNIP address • SNMP Engine ID • Hostname of cluster	A spotted configuration can be defined using one of two approaches. • **SNIP address.** When creating the SNIP address, specify the node on which you want the SNIP address to be active, as the owner node. **Example:**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuration Type	Active on...	Applicable to...	Configurations...
		nodes • Entities that can be bound to a node group	add ns ip 10.102.29.106 255.255.255.0 -type SNIP -ownerNode 2 (assuming node NS2 ID is 2) Note: You cannot change the ownership of a spotted SNIP address at run time. To change the ownership, you must first delete the SNIP address and add it again by specifying the new owner. • **Entities that can be bound to a node group.** By binding the entity to a single-member node group.

Note: Citrix recommends that you use spotted SNIP addresses. You can use striped SNIP addresses only if there is a shortage of IP addresses. The use of striped IP addresses can result in ARP flux issues.

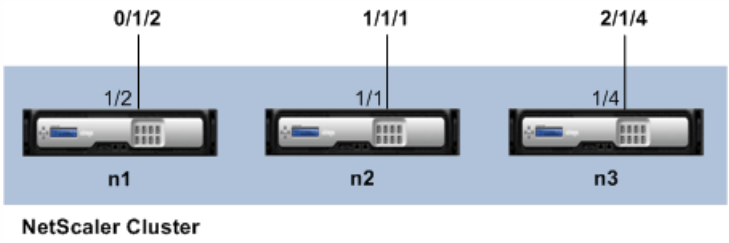
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Communication in a Cluster Setup

Jun 17, 2015

The interfaces of NetScaler appliances that are added to a cluster, are prefixed with a node ID. This helps identify the cluster node to which the interface belongs. Therefore, the interface identifier c/u, where c is the controller number and u is the unit number, now becomes n/c/u, where n is the node ID. For example, in the following figure, interface 1/2 of node n1 is represented as 0/1/2, interface 1/1 of node n2 is represented as 1/1/1, and interface 1/4 of node n3 is represented as 2/1/4.

Figure 1. Interface naming convention in a cluster



Server communication

The cluster communicates with the server through the physical connections between the cluster node and the server-side connecting device. The logical grouping of these physical connections is called the server data plane.

Client communication

The cluster communicates with the client through the physical connections between the cluster node and the client-side connecting device. The logical grouping of these physical connections is called the client data plane.

Inter-node communication

The cluster nodes can also communicate with each other. The manner in which they communicate depends on whether the node exists on the same network or across networks.

- Cluster nodes within the same network communicate with each other by using the cluster backplane. The backplane is a set of interfaces in which one interface of each node is connected to a common switch, which is called the cluster backplane switch. The different types of traffic that goes through backplane, which is used by internode communication are:
  - Node to Node Messaging (NNM)
  - Steered traffic
  - Configuration propagation and synchronization
- Each node of the cluster uses a special MAC cluster backplane switch address to communicate with other nodes through the backplane. The cluster special MAC is of the form: **0x02 0x00 0x6F <cluster_id> <node_id> <reserved>,** where <cluster_id> is the cluster instance ID, <node_id> is the node number of the NetScaler appliance that are added to a cluster.
- Across networks, steering of packets is done through a GRE tunnel and other node-to-node communication is routed across nodes as required.

The following figures shows the communication interfaces in L2 clusters and L3 clusters.

Figure 2. Cluster communication interfaces - L2 cluster

© 1999-2017 Citrix Systems, Inc. All rights reserved.

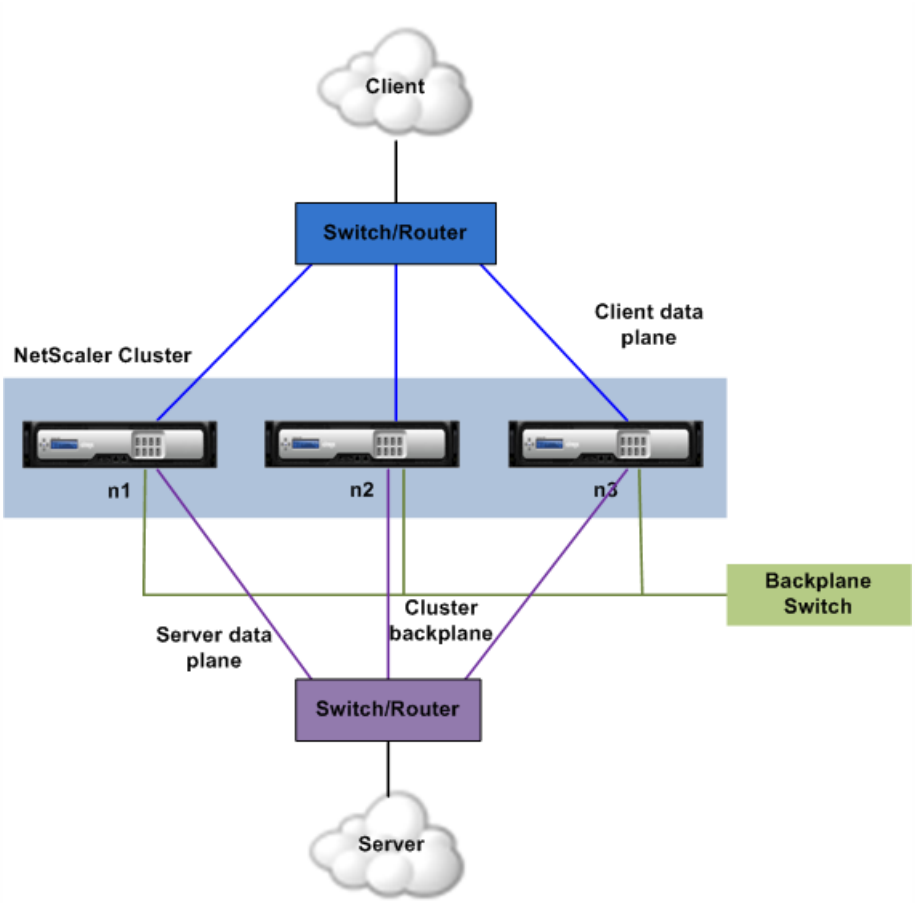
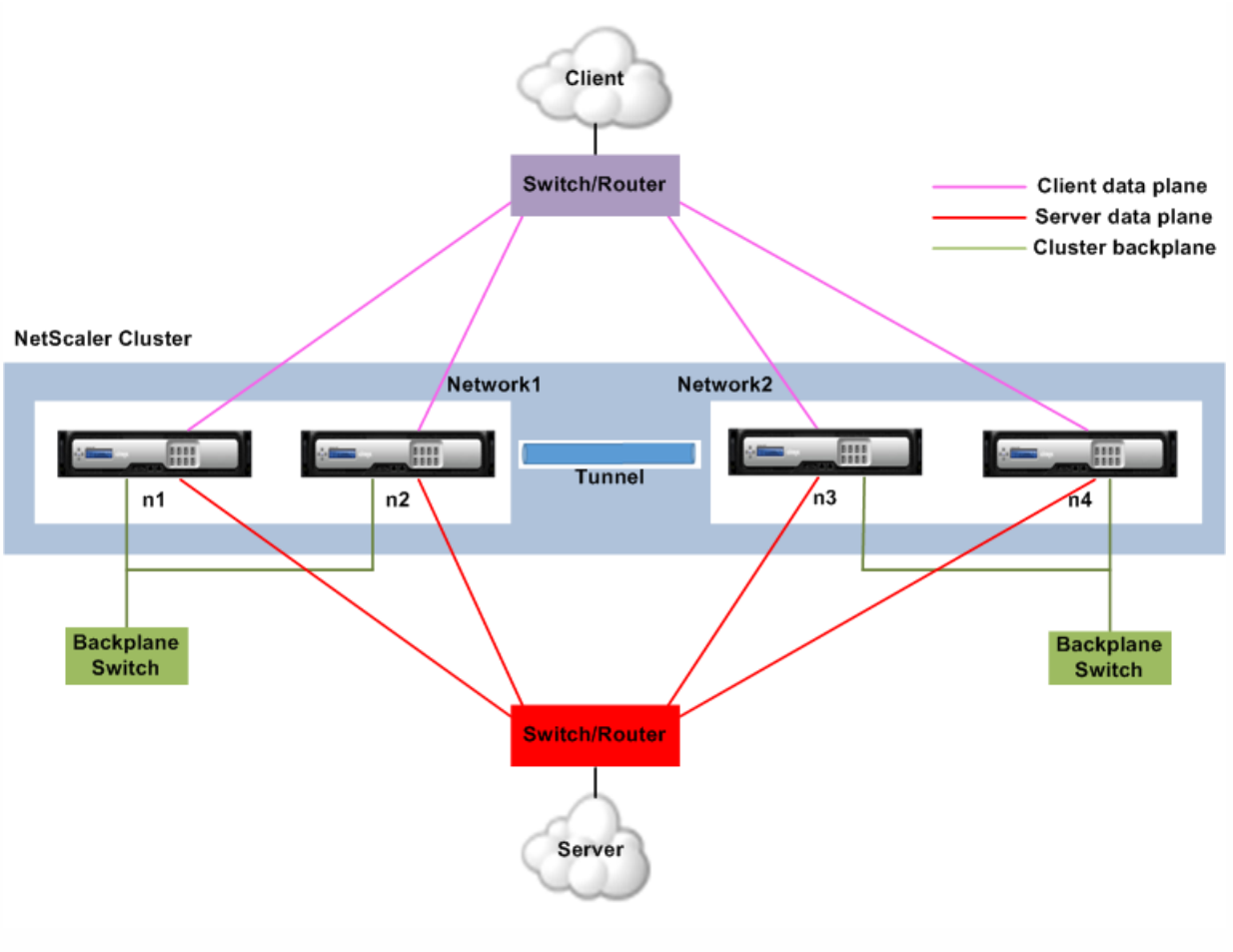


Figure 3. Cluster communication interfaces - L3 cluster

© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Traffic Distribution in a Cluster Setup

Mar 23, 2017

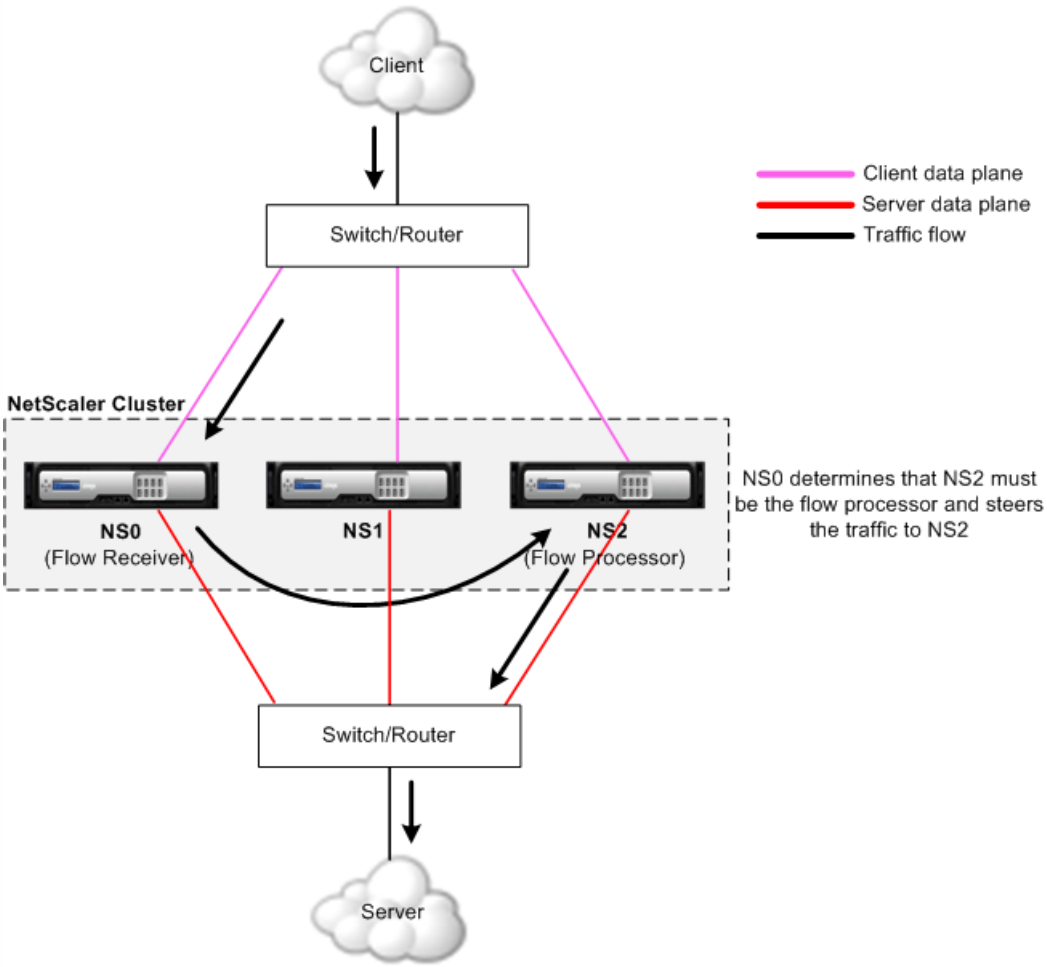
In a cluster setup, external networks view the collection of NetScaler appliances as a single entity. So, the cluster must select a single node that must receive the traffic. The cluster does this selection by using Equal Cost Multiple Path (ECMP) or cluster link aggregation traffic distribution mechanism. The selected node is called the flow receiver.

Note: For an L3 cluster (nodes across different networks), only ECMP traffic distribution can be used.
The flow receiver gets the traffic and then, using internal cluster logic determines the node that must process the traffic. This node is called the flow processor. The flow receiver steers the traffic to the flow processor over the backplane (if the flow receiver and the flow processor are on the same network) or through the tunnel (if the flow receiver and the flow processor are on different networks).

Note:

- The flow receiver and flow processor must be nodes capable of serving traffic.
- From NetScaler 11 onwards, you can disable steering on the cluster backplane. For more information, see "Disabling Steering on the Cluster Backplane".

Figure 1. Traffic distribution in a cluster



The above figure shows a client request flowing through the cluster. The client sends a request to a virtual IP (VIP) address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

A traffic distribution mechanism configured on the client data plane selects one of the cluster nodes as the flow receiver. The flow receiver receives the traffic, determines the node that must process the traffic, and steers the request to that node (unless the flow receiver selects itself as the flow processor).

The flow processor establishes a connection with the server. The server processes the request and sends the response to the subnet IP (SNIP) address that sent the request to the server.

- If the SNIP address is a striped or partially striped IP address, the traffic distribution mechanism configured on the server data plane selects one of the cluster nodes as the flow receiver. The flow receiver receives the traffic, determines the flow processor, and steers the request to the flow processor through the cluster backplane.
- If the SNIP address is a spotted IP address, the node that owns the SNIP address receives the response from the server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

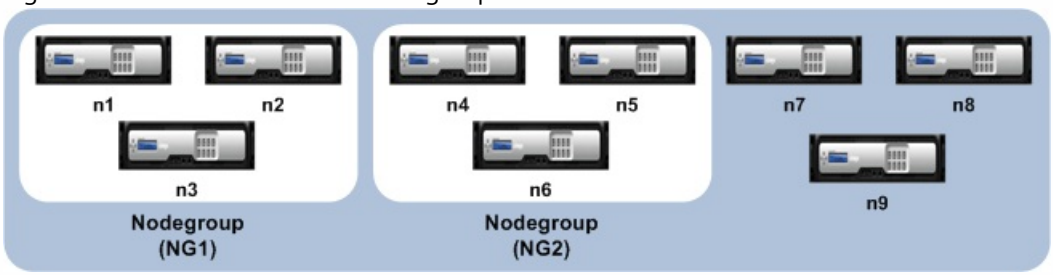
Cluster Nodegroups

May 25, 2015

Note: Nodegroups are supported from NetScaler 10.1 onwards.

As the name indicates, a cluster nodegroup is a group of cluster nodes.

Figure 1. NetScaler cluster with nodegroups



The above figure shows a cluster which has nodegroups NG1 and NG2 that include 3 cluster nodes each. The cluster also has 3 nodes that are not part of any nodegroup.

A nodegroup can be configured for the following:

- To define spotted and partially striped configurations. For more information, see "Nodegroups for Spotted and Partially-Striped Configurations".
- To configure redundancy of nodegroups. For more information, see "Configuring Redundancy for Nodegroups".
  Note: Supported from NetScaler 10.5 Build 52.1115.e onwards.
- To define an L3 cluster (also called a cluster in INC mode). In an L3 cluster, cluster nodes can be from different networks. You must group nodes that belong to a network in a single nodegroup. For example, if n1, n2, n3 are in network1 and n4, n5, n6 are in network2, then NG1 must include nodes of network1 and NG2 must include nodes of network2. For setting up an L3 cluster, see "Creating a NetScaler Cluster".
  Note: Supported from NetScaler 11 onwards.

Note: The above functions of a nodegroup are mutually exclusive. This means that a nodegroup can provide only one of the above mentioned functionality.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cluster and Node States

Sep 23, 2015

For a cluster to be functional, a majority of the nodes (n/2 + 1) must be operationally active (operational state is ACTIVE). Check table below.

> **Important**
>
> From NetScaler release 10.5, you can configure the cluster to be functional even when the majority criteria is not satisfied. This configuration must be performed when creating a cluster.

The following table describes the states of a cluster node:

Type	Possible values	Description
Admin state	PASSIVE, ACTIVE, SPARE	Specifies the responsibility of the node within the cluster setup. You can set the admin state to be one of the following: • **PASSIVE** (default state). These nodes are in sync with the cluster, but do not serve any traffic. You must explicitly change its state to one of the other admin states, to make it ready to play a more active role. • **ACTIVE.** These nodes are in sync with the cluster and serve traffic that reaches the cluster. Check operational state for information on when this node can serve traffic. • **SPARE.** These nodes act as backup nodes for the cluster. Spare nodes are always in sync with the cluster, but do not serve any traffic till one of the ACTIVE (admin state) nodes becomes unavailable. When this happens, the admin state of this node continues to be SPARE, but its operational state changes to ACTIVE. **Note:** The preemption parameter that is specified on a cluster instance, indicates whether the SPARE node remains operationally active even when a ACTIVE (admin state) node becomes available. • If preemption is disabled, the spare node continues to serve traffic even if a node in ACTIVE admin state comes back online. • If preemption is enabled, when a node in ACTIVE admin state comes back online, it preempts the spare node and starts serving traffic. The spare node goes back to inactive state.
Health	UP, NOT UP, UNKNOWN	Indicates whether the cluster node can successfully handle traffic by checking for the following criteria for the node: • The interfaces are up and enabled. • The SSL cards are available. • The cluster synchronization operation is enabled and completed. • The backplane interface is up and enabled. • The CLAG member(s) are up. Based on the above criteria, the health of a node can be: • **UP.** When all the criteria are satisfied.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Type	Possible values	• **UNKNOWN.** When the node cannot receive heartbeats from other nodes. Description

Operational state	ACTIVE, INACTIVE, UNKNOWN	Indicates that the node can serve traffic. The operational state of a node is determined by a combination of the admin state and the health of the node.
		• **ACTIVE.** When the health of the node is UP and one of the following is true:
		  • Node in ACTIVE admin state.
		  • Node in SPARE admin state and node is being used as backup.
		• **INACTIVE.** In the following cases:
		  • Node in PASSIVE admin state, regardless of the health.
		  • Node in ACTIVE admin state and health is NOT UP.
		  • Node in SPARE admin state and node is not being used as backup.
		• **UNKNOWN.** When the node cannot receive heartbeats from other nodes.
		Review the *ns.log* file, error counters, and the output of the "show cluster node" command, to help determine the exact reason for a node to be in INACTIVE and UNKNOWN state.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Routing in a Cluster

May 07, 2015

Routing in a cluster works in much the same way as routing in a standalone system. A few points to note:

- All routing configurations must be performed from the cluster IP address and the configurations are propagated to the other cluster nodes.
- Routing runs only on spotted SNIP addresses and NSIP addresses.
- Routes are limited to the maximum number of ECMP routes supported by the upstream router
- Node-specific routing configurations must be performed by using the owner-node argument as follows:

```
!
interface vlan97
!
router ospf
 owner-node 0
  ospf router-id 97.131.0.1
 exit-owner-node
 owner-node 1
  ospf router-id 97.131.0.2
 exit-owner-node
 owner-node 2
  ospf router-id 97.131.0.3
 exit-owner-node
 redistribute kernel
 network 97.0.0.0/8 area 0
!
```

- Retrieve node-specific routing configurations by specifying the node(s) in the owner-node argument as follows:

```
> vtysh
ns# owner-node 0 1
ns(node-0 1)# show cluster state
ns(node-0 1)# exit-owner-node
```

- Clear node-specific routing configurations by specifying the node(s) in the owner-node argument as follows:

```
> vtysh
ns# owner-node 0 1
ns(node-0 1)# clear config
ns(node-0 1)# exit-owner-node
```

- Routing protocol daemons can run and adjacencies can be formed on active and inactive nodes of a cluster.
- Only active nodes advertise host routes to striped VIP addresses. Spotted VIP addresses are advertised by active owner node.
- Active and inactive nodes can learn dynamic routes and install them into the routing table.
- Routes learnt on a node are propagated to other nodes in the cluster only if route propagation is configured. This is mostly needed in asymmetric topologies where the unconnected nodes may not be able to form adjacencies.

  ```
  ns(config)# ns route-install propagate
  ```

  Note: Make sure that route propagation is not configured in a symmetric cluster topology as it can result in making the node unavailable to the cluster.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting up a NetScaler Cluster

Jun 15, 2015

NetScaler appliances that you want to add to the cluster must satisfy the criteria specified in "Prerequisites for Cluster Nodes". Before actually setting up a cluster, you must be aware of cluster basics. For information, see "Cluster Overview".

Forming a cluster requires you to set up inter-node communication, create the cluster (by adding the first NetScaler appliance), and then add the other cluster nodes. Each of these steps is explained with relevant details in subsequent topics.

Note: While there are some differences in setting up an L2 and L3 cluster, there are many similarities too. The subsequent topics explain the setup for both cluster types while highlighting the configurations that are specific to L3 clusters.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting up Inter-Node Communication

Feb 04, 2016

The nodes in a cluster setup communicate with one another using the following inter-node communication mechanisms:

- Nodes that are within the network (same subnet) communicate with each other through the cluster backplane. The backplane must be explicitly set up. See the detailed steps listed below.
- Across networks, steering of packets is done through a GRE tunnel and other node-to-node communication is routed across nodes as required.
  Note:
  - A cluster can include nodes from different networks from NetScaler 11.0 onwards.
  - In an L3 cluster deployment, packets between NetScaler nodes are exchanged over an unencrypted GRE tunnel that uses the NSIP addresses of the source and destination nodes for routing. When this exchange occurs over the internet, in the absence of an IPSec tunnel, the NSIPs will be exposed on the internet and this could result in security issues. Citrix advises customers to establish their own IPSec solution when using a L3 cluster.

1. Identify the network interface that you want to use for the backplane.
2. Connect an Ethernet or optical cable from the selected network interface to the cluster backplane switch.

For example, to use interface 1/2 as the backplane interface for node 4, connect a cable from the 1/2 interface of node 4 to the backplane switch.

Important points to note when setting up the cluster backplane

- Do not use the appliance's management interface (0/x) as the backplane interface. In a cluster, the interface 0/1/x is read as:

  0 -> node ID 0
  1/x -> NetScaler interface

- Backlane interfaces must not be used for the client or server data planes.
- Configure a link aggregate (LA) channel to optimize the throughput of the cluster backplane.
- Citrix recommends that you dedicate a separate switch for the backplane, so that large amounts of traffic can be handled seamlessly.
- Backplane interfaces of all nodes of a cluster must be connected to the same switch and bound to the same L2 VLAN.
- If you have multiple clusters with the same cluster instance ID, make sure that the backplane interfaces of each cluster are bound to a different VLAN.
- The backplane interface is always monitored, regardless of the HA monitoring settings of that interface.
- The state of MAC spoofing on the different virtualization platforms can affect the steering mechanism on the cluster backplane. Therefore, make sure the appropriate state is configured:

  - XenServer - Disable MAC spoofing
  - Hyper-V - Enable MAC spoofing

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- VMware ESX - Enable MAC spoofing (also make sure "Forged Transmits" is enabled)

- The MTU for the cluster backplane is automatically updated. However, if jumbo frames are configured on the cluster, the MTU of the cluster backplane must be explicitly configured. The value must be set to 78 + X, where X is the maximum MTU of the client and server data planes. For example, if MTU of server data plane is 7500 and of the client data plane is 8922, then the MTU of cluster backplane must be set to 78 + 8922 = 9000. To set this MTU, use the following command:
  > set interface <backplane_interface> -mtu <value>
- The MTU for interfaces of the backplane switch must be specified to be greater than or equal to 1578 bytes, if the cluster has features like MBF, L2 policies, ACLs, routing in CLAG deployments, and vPath.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating a NetScaler Cluster

Apr 03, 2015

To create a cluster, start by taking one of the NetScaler appliances that you want to add to the cluster. On this node, you must create the cluster instance and define the cluster IP address. This node is the first cluster node and is called the cluster configuration coordinator. All configurations that are performed on the cluster IP address are stored on this node and then propagated to the other cluster nodes.

The responsibility of configuration coordination in a cluster is not fixed to a specific node. It can change over time depending on the following factors:

- The priority of the node. The node with the highest priority (lowest priority number) is made the configuration coordinator. Therefore, if a node with a priority number lower than that of the existing configuration coordinator is added, the new node takes over as the configuration coordinator.
  Note: Node priority can be configured from NetScaler 10.1 onwards.
- If the current configuration coordinator goes down. The node with the next lowest priority number takes over as the configuration coordinator. If the priority is not set or if there are multiple nodes with the lowest priority number, the configuration coordinator is selected from one of the available nodes.

Note: The configurations of the appliance (including SNIP addresses and VLANs) are cleared by implicitly executing the clear ns config extended command. However, the default VLAN and NSVLAN are not cleared from the appliance. Therefore, if you want the NSVLAN on the cluster, make sure it is created before the appliance is added to the cluster. For an L3 cluster (cluster nodes on different networks), networking configurations are not cleared from the appliance.

1. Log on to an appliance (for example, appliance with NSIP address 10.102.29.60) that you want to add to the cluster.
2. Add a cluster instance.
   add cluster instance <clId> -quorumType <NONE | MAJORITY> -inc <ENABLED | DISABLED>

   Note:
   - The cluster instance ID must be unique within a LAN.
   - For an L3 cluster, make sure the inc parameter is set to ENABLED. The "inc" parameter must be disabled for an L2 cluster.
3. [Only for an L3 cluster] Create a nodegroup. In the next step, the newly added cluster node must be associated with this nodegroup.
   Note: This nodegroup will include all or a subset of the NetScaler appliances that belong to the same network.
   add cluster nodegroup <name>

4. Add the NetScaler appliance to the cluster.
   add cluster node <nodeId> <IPAddress> -state <state> -backplane <interface_name> -nodegroup <name>

   Note: For an L3 cluster:
   - The nodegroup parameter must be set to the name of the nodegroup created above.
   - The backplane parameter is mandatory for nodes that are associated with a nodegroup that has more than one node, so that the nodes within the network can communicate with each other.

   Example

   Adding a node for an L2 cluster (all cluster nodes are in the same network).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1

Adding a node for an L3 cluster which includes a single node from each network. Here, you do not have to set the backplane.

> add cluster node 0 10.102.29.60 -state PASSIVE -nodegroup ng1

Adding a node for an L3 cluster which includes multiple nodes from each network. Here, you have to set the backplane so that nodes within a network can communciate with each other.

> add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1 -nodegroup ng1

5. Add the cluster IP address (for example, 10.102.29.61) on this node.
   add ns ip <IPAddress> <netmask> -type clip

   Example

   > add ns ip 10.102.29.61 255.255.255.255 -type clip
6. Enable the cluster instance.
   enable cluster instance <clId>

7. Save the configuration.
   save ns config

8. Warm reboot the appliance.
   reboot -warm

   Verify the cluster configurations by using the show cluster instance command. Verify that the output of the command displays the NSIP address of the appliance as a node of the cluster.

1. Log on to an appliance (for example, an appliance with NSIP address 10.102.29.60) that you intend to add to the cluster.
2. Navigate to System > Cluster.
3. In the details pane, click the Manage Cluster link.
4. In the Cluster Configuration dialog box, set the parameters required to create a cluster. For a description of a parameter, hover the mouse cursor over the corresponding text box.
5. Click Create.
6. In the Configure cluster instance dialog box, make sure that the Enable cluster instance check box is selected.
7. In the Cluster Nodes pane, select the node and click Open.
8. In the Configure Cluster Node dialog box, set the State.
9. Click OK, and then click Save.
10. Warm reboot the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding a Node to the Cluster

Jun 17, 2015

You can seamlessly scale the size of a cluster to include a maximum of 32 nodes. When a NetScaler appliance is added to the cluster, the configurations from that appliance are cleared (by internally executing the clear ns config -extended command). The SNIP addresses, MTU settings of the backplane interface, and all VLAN configurations (except the default VLAN and NSVLAN) are also cleared from the appliance.

The cluster configurations are then synchronized on this node. There can be an intermittent drop in traffic while the synchronization is in progress.

Important: Before you add a NetScaler appliance to a cluster:

- Set up the backplane interface for the node. Check preceding topic.
- Check if the licenses that are available on the appliance match those availabe on the configuration coordinator. The appliance is added only if the licenses match.
- If you want the NSVLAN on the cluster, make sure that the NSVLAN is created on the appliance before it is added to the cluster.
- Citrix recommends that you add the node as a passive node. Then, after joining the node to the cluster, complete the node specific configuration from the cluster IP address. Run the force cluster sync command if the cluster has only spotted IP addresses, has L3 VLAN binding, or has static routes.
- When an appliance with a preconfigured link aggregate (LA) channel is added to a cluster, the LA channel continues to exist in the cluster environment. The LA channel is renamed from LA/x to nodeId/LA/x, where LA/x is the LA channel identifier.

1. Log on to the cluster IP address and, at the command prompt, do the following:
   1. Add the appliance (for example, 10.102.29.70) to the cluster.
      add cluster node <nodeId> <IPAddress> -state <state> -backplane <interface_name> -nodegroup <name>

      Note: For an L3 cluster:
      - The nodegroup parameter must be set to a nodegroup that has nodes of the same network.
        - If this node belongs to the same network as the first node that was added, then configure the nodegroup that was used for that node.
        - If this node belongs to a different network, then create a nodegroup and bind this node to the nodegroup.
      - The backplane parameter is mandatory for nodes that are associated with a nodegroup that has more than one node, so that the nodes within the network can communicate with each other.

      Example

      > add cluster node 1 10.102.29.70 -state PASSIVE -backplane 1/1/1
   2. Save the configuration.
      save ns config
2. Log on to the newly added node (for example, 10.102.29.70) and do the following:
   1. Join the node to the cluster.
      join cluster -clip <ip_addr> -password <password>

      Example

      > join cluster -clip 10.102.29.61 -password nsroot

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Perform the following configurations:
   1. If the node is added to a cluster that has only spotted IPs, the configurations are synchronized before the spotted IP addresses are assigned to that node. In such cases, L3 VLAN bindings can be lost. To avoid this loss, either add a striped IP or add the L3 VLAN bindings.
   2. Define the required spotted configurations.
   3. Set the MTU for the backplane interface.
3. Save the configuration.
   save ns config
4. Warm reboot the appliance.
   reboot -warm

1. Log on to the cluster IP address.
2. Navigate to System > Cluster > Nodes.
3. In the details pane, click Add to add the new node (for example, 10.102.29.70).
4. In the Create Cluster Node dialog box, configure the new node. For a description of a parameter, hover the mouse cursor over the corresponding text box.
5. Click Create. When prompted to perform a warm reboot, click Yes.

If you have used the command line to add a node to the cluster, but have not joined the node to the cluster, you can use the following procedure.

Note: When a node joins the cluster, it takes over its share of traffic from the cluster and hence an existing connection can get terminated.

1. Log on to the node that you want to join to the cluster (for example, 10.102.29.70).
2. Navigate to System > Cluster.
3. In the details pane, under Get Started, click the Join Cluster link.
4. In the Join to existing cluster dialog box, set the cluster IP address and the nsroot password of the configuration coordinator. For a description of a parameter, hover the mouse cursor over the corresponding text box.
5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing the Details of a Cluster

Feb 13, 2015

You can view the details of the cluster instance and the cluster nodes by logging on to the cluster IP address.

Log on to the cluster IP address and, at the command prompt, type:

show cluster instance <clId>

Note: When executed from the NSIP address of a cluster node that is not the configuration coordinator, this command displays the status of the cluster on this node.

Log on to the cluster IP address and, at the command prompt, type:

show cluster node <nodeId>

1. Log on to the cluster IP address.
2. Navigate to System > Cluster.
3. In the details pane, under Get Started, click the Manage Cluster link to view the details of the cluster.

1. Log on to the cluster IP address.
2. Navigate to System > Cluster > Nodes.
3. In the details pane, click the node for which you want to view the details.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Distributing Traffic Across Cluster Nodes

Feb 13, 2015

After you have created the NetScaler cluster and performed the required configurations, you must deploy Equal Cost Multiple Path (ECMP) or cluster Link Aggregation (LA) on the client data plane (for client traffic) or server data plane (for server traffic). These mechanisms distribute external traffic across the cluster nodes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using Equal Cost Multiple Path (ECMP)

May 04, 2015

With the Equal Cost Multiple Path (ECMP) mechanism, virtual server IP addresses are advertised by all active cluster nodes. This means that traffic can be received by any cluster node, which then steers the traffic to the node that must process the traffic. There can be redundant steering in case of spotted and partially striped virtual servers. Therefore, from NetScaler 11 onwards, spotted and partially striped virtual server IP addresses are advertised only by the owner nodes. This reduces the redundant steering.

You must have detailed knowledge of routing protocols to use ECMP. For more information, see "Configuring Dynamic Routes. For more information on routing in a cluster, see "Routing in a Cluster".

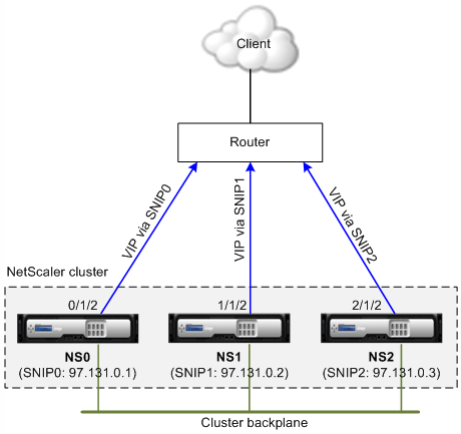
To use ECMP, you must first enable the required routing protocol (OSPF, RIP, BGP, or ISIS) on the cluster IP address. You must bind the interfaces and the spotted IP address (with dynamic routing enabled) to a VLAN. Configure the selected routing protocol and redistribute the kernel routes on the ZebOS by using the vtysh shell.

You must perform similar configurations on the cluster IP address and on the external connecting device.

Note:

- Make sure that the licenses on the cluster support dynamic routing, otherwise ECMP does not work.
- ECMP is not supported for wildcard virtual servers since RHI needs a VIP address to advertise to a router and wildcard virtual servers do not have associated VIP addresses.

Figure 1. ECMP topology



As seen in the above figure, the ECMP router can reach the VIP address via SNIP0, SNIP1, or SNIP2.

1. Log on to the cluster IP address.
2. Enable the routing protocol.
   enable ns feature <feature>

   Example: To enable the OSPF routing protocol.
   > enable ns feature ospf
3. Add a VLAN.
   add vlan <id>

   Example
   > add vlan 97
4. Bind the interfaces of the cluster nodes to the VLAN.
   bind vlan <id> -ifnum <interface_name>

   Example
   > bind vlan 97 -ifnum 0/1/2 1/1/2 2/1/2
5. Add a spotted SNIP address for each node and enable dynamic routing on it.
   add ns ip <SNIP> <netmask> -ownerNode <positive_integer> -dynamicRouting ENABLED

   Example
   > add ns ip 97.131.0.1 255.0.0.0 -ownerNode 0 -dynamicRouting ENABLED -type SNIP
   > add ns ip 97.131.0.2 255.0.0.0 -ownerNode 1 -dynamicRouting ENABLED -type SNIP
   > add ns ip 97.131.0.3 255.0.0.0 -ownerNode 2 -dynamicRouting ENABLED -type SNIP
6. Bind one of the spotted SNIP addresses to the VLAN. When you bind one spotted SNIP address to a VLAN, all other spotted SNIP addresses defined on the cluster in that subnet are automatically bound to the VLAN.
   bind vlan <id> -IPAddress <SNIP> <netmask>

   Example
   > bind vlan 97 -ipAddress 97.131.0.1 255.0.0.0
   Note: You can use NSIP addresses of the cluster nodes instead of adding SNIP addresses. If so, you do not have to perform steps 3 - 6.
7. Configure the routing protocol on ZebOS using vtysh shell.
   Example: To configure OSPF routing protocol on node IDs 0, 1, and 2.

   > vtysh ! interface vlan97 ! router ospf owner-node 0 ospf router-id 97.131.0.1 exit-owner-node owner-node 1 ospf router-id 97.131.0.2 exit-owner-node owner-node 2 ospf router-id 97.131
   Note: For VIP addresses to be advertised, RHI setting must done by using the vserverRHILevel parameter as follows:
   add ns ip <IPAddress> <netmask> -type VIP -vserverRHILevel <vserverRHILevel>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For OSPF specific RHI settings, there are additional settings that can be done as follows:

add ns ip <IPAddress> <netmask> -type VIP -ospfLSAType (TYPE1 | TYPE5) -ospfArea <positive_integer>

Use the add ns ip6 command to perform the above commands on IPv6 addresses.

8. Configure ECMP on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

   //For OSPF (IPv4 addresses) Global config: Configure terminal feature ospf Interface config: Configure terminal interface Vlan10 no shutdown ip address 97.131.0.5/8 Configure terminal ro

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case: ECMP with BGP Routing

Aug 06, 2013

To configure ECMP with BGP routing protocol, perform the following steps:

1. Log on to the cluster IP address.
2. Enable BGP routing protocol.
   > enable ns feature bgp
3. Add VLAN and bind the required interfaces.
   > add vlan 985
   > bind vlan 985 -ifnum 0/0/1 1/0/1
4. Add the spotted IP address and bind them to the VLAN.
   > add ns ip 10.100.26.14 255.255.255.0 -ownerNode 1 -dynamicRouting ENABLED
   > add ns ip 10.100.26.15 255.255.255.0 -ownerNode 2 -dynamicRouting ENABLED
   > bind vlan 985 -ipAddress 10.100.26.10 255.255.255.0
5. Configure BGP routing protocol on ZebOS using vtysh shell.
   > vtysh
   conf t
   router bgp 65535
   neighbor 10.100.26.1 remote-as 65535
6. Configure BGP on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.
   router bgp 65535
   no synchronization
   bgp log-neighbor-changes
   neighbor 10.100.26.14 remote-as 65535
   neighbor 10.100.26.15 remote-as 65535
   no auto-summary
   dont-capability-negotiate
   dont-capability-negotiate
   no dynamic-capability

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using Cluster Link Aggregation

Feb 08, 2016

Cluster link aggregation, as the name suggests, is a group of interfaces of cluster nodes. It is an extension of NetScaler link aggregation. The only difference is that, while link aggregation requires the interfaces to be from the same device, in cluster link aggregation, the interfaces are from different nodes of the cluster. For more information about link aggregation, see "Configuring Link Aggregation".

Important

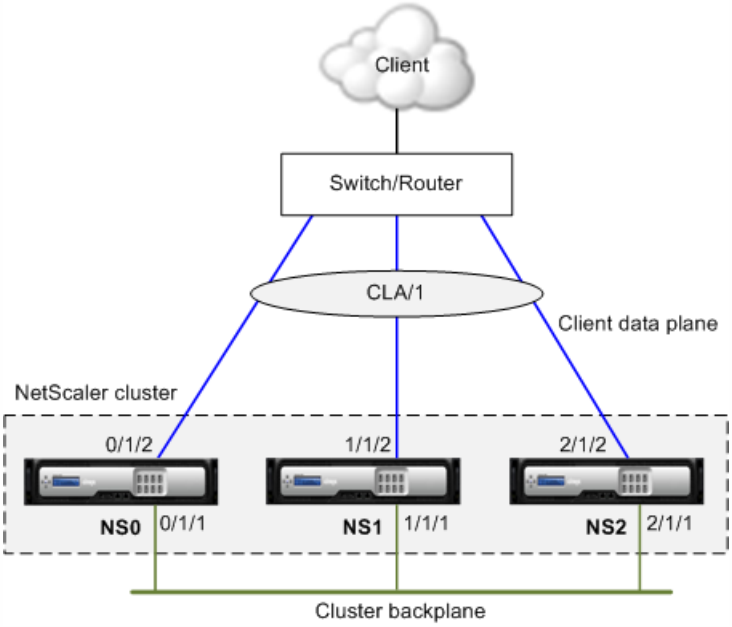
- Cluster link aggregation is supported for a cluster of hardware (MPX) appliances.
- Cluster link aggregation is supported for a cluster of virtual (VPX) appliances that are deployed on ESX and KVM hypervisors, with the following restrictions:
  - Dedicated interfaces need to be used. This means that the interfaces must not be shared with other virtual machines.
  - If the cluster link aggregation member interfaces are manually disabled or if cluster link aggregation itself is manually disabled, then interface power down capability is achieved only by LACP timeout mechanism.
  - Jumbo MTU is not supported on LACP cluster link aggregation.

    Note: Cluster link aggregation is not supported on VPX appliances that are deployed on XenServer, AWS, and Hyper-V.
- Cluster link aggregation is not supported on NetScaler SDX appliances.
- The number of interfaces that can be bound to cluster LA is 16 (from each node), and the maximum number of interfaces in cluster LA can be (16 * n), where n is the number of nodes in a cluster. The total number of interfaces in cluster LA depends on the number of interfaces for every port channel on upstream switch.

For example, consider a three-node cluster where all three nodes are connected to the upstream switch. A cluster LA channel (CLA/1) is formed by binding interfaces 0/1/2, 1/1/2, and 2/1/2.

Figure 1. Cluster Link Aggregation topology

© 1999-2017 Citrix Systems, Inc. All rights reserved.

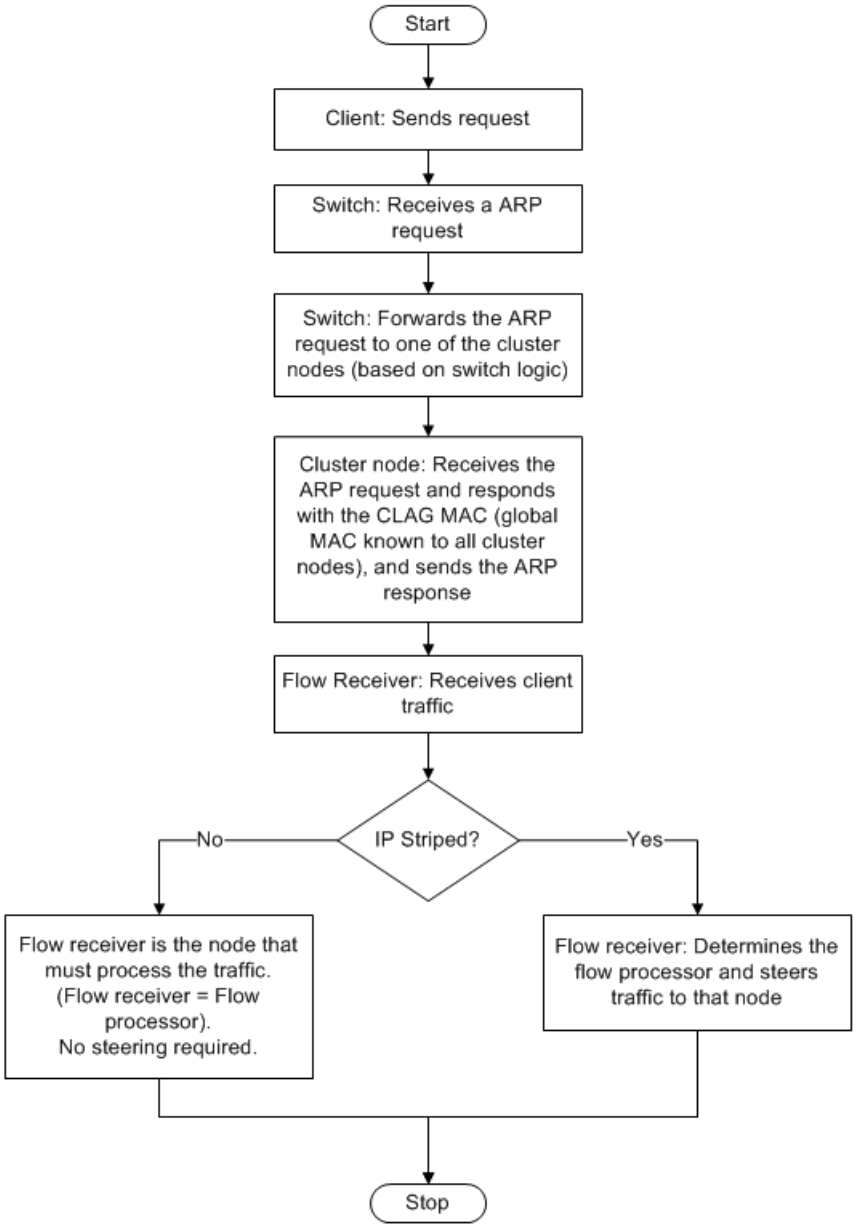


A cluster LA channel has the following attributes:

- Each channel has a unique MAC agreed upon by cluster nodes.
- The channel can bind both local and remote nodes' interfaces.
- A maximum of four cluster LA channels are supported in a cluster.
- Backplane interfaces cannot be part of a cluster LA channel.
- When an interface is bound to a cluster LA channel, the channel parameters have precedence over the network interface parameters. A network interface can be bound to one channel only.
- Management access to a cluster node, must not be configured on a cluster LA channel (for example, CLA/1) or its member interfaces. This is because when the node is INACTIVE, the corresponding cluster LA interface is marked as power down and therefore looses management access.

Figure 2. Traffic distribution flow using cluster LA

© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Static Cluster Link Aggregation

Jul 16, 2015

You must configure a static cluster LA channel on the cluster IP address and on the external connecting device. If possible, configure the upstream switch to distribute traffic based on IP address or port instead of MAC address.

1. Log on to the cluster IP address.
   Note: Make sure that you configure the cluster LA channel on the cluster IP address before configuring link aggregation on the external switch. Otherwise, the switch will forward traffic to the cluster even though the cluster LA channel is not configured. This can lead to loss of traffic.
2. Create a cluster LA channel.
   add channel <id> -speed <speed>

   Example
   > add channel CLA/1 -speed 1000
   Note: You must not specify the speed as AUTO. Rather, you must explicitly specify the speed as 10, 100, 1000, or 10000. Only interfaces that have the speed matching the <speed> attribute in the cluster LA channel are added to the active distribution list.
3. Bind the required interfaces to the cluster LA channel. Make sure that the interfaces are not used for the cluster backplane.
   bind channel <id> <ifnum>

   Example
   > bind channel CLA/1 0/1/2 1/1/2 2/1/2
4. Verify the configurations.
   show channel <id>

   Example
   > show channel CLA/1
   Note: You can bind the cluster LA channel to a VLAN by using the bind vlan command. The interfaces of the channel are automatically bound to the VLAN.
5. Configure static LA on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1). Similar configurations must be performed on other switches.
   Global config:
   Configure terminal

   Interface level config:

```
interface Ethernet2/47
  switchport
  switchport access vlan 10
  channel-group 7 mode on
  no shutdown

interface Ethernet2/48
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
switchport
switchport access vlan 10
channel-group 7 mode on
no shutdown
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Dynamic Cluster Link Aggregation

Apr 29, 2015

Dynamic cluster LA channel uses Link Aggregation Control Protocol (LACP).

You must perform similar configurations on the cluster IP address and on the external connecting device. If possible, configure the upstream switch to distribute traffic based on IP address or port instead of MAC address.

Points to remember:

- Enable LACP (by specifying the LACP mode as either ACTIVE or PASSIVE).

  Note: Make sure the LACP mode is not set as PASSIVE on both the NetScaler cluster and the external connecting device.
- Specify the same LACP key on each interface that you want to be the part of the channel. For creating a cluster LA channel, the LACP key can have a value from 5 through 8. For example, if you set the LACP key on interfaces 0/1/2, 1/1/2, and 2/1/2 to 5, CLA/1 is created. The interfaces 0/1/2, 1/1/2, and 2/1/2 are automatically bound to CLA/1. Similarly, if you set the LACP key to 6, CLA/2 channel is created.
- Specify the LAG type as Cluster.

On the cluster IP address, for each interface that you want to add to the cluster LA channel, type:

set interface <id> -lacpMode <lacpMode> -lacpKey <positive_integer> -lagType CLUSTER

Example: To configure a cluster LA channel CLA/1 of 3 interfaces.

```
> set interface 0/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 1/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 2/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
```

Note: Optionally, you can enable Link Redundancy in a Cluster with LACP.

Similarly, configure dynamic LA on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1). Similar configurations must be performed on other switches.

```
Global config:
Configure terminal
feature lacp
Interface level config:

interface Ethernet2/47
  switchport
  switchport access vlan 10
  channel-group 7 mode active
  no shutdown

interface Ethernet2/48
  switchport
  switchport access vlan 10
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
channel-group 7 mode active
no shutdown
```

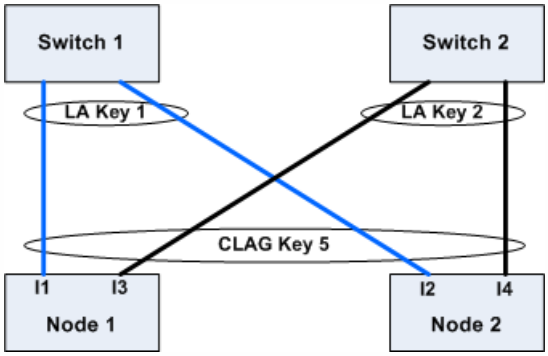

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Link Redundancy in a Cluster with LACP

May 05, 2015

A NetScaler cluster provides link redundancy for LACP to ensure that all nodes have the same partner key.

To understand the need for link redundancy, let us consider the example of the following cluster setup along with the accompanying cases (with attention to case 3):



In this setup, interfaces I1, I2, I3 and I4 are bound to LACP channel with KEY 5. On the partner side, I1 and I2 are connected to Switch 1 to form a single LA channel with KEY 1. Similarly, I3 and I4 are connected to Switch 2 to form a single LA channel with KEY 2.

Now let us consider the following cases to understand the need for link redundancy:

Case 1: Switch 1 is up and Switch 2 is down

In this case, cluster LA on both the nodes would stop receiving LACPDUs from Key2 and would start receiving LACPDUs from Key1. On both the nodes, cluster LA is connected to KEY 1 and I1 and I2 will be UP and channel on both the nodes would be UP.

Case 2: Switch1 goes down and Switch2 becomes UP

In this case, cluster LA on both the nodes would stop receiving LACPDUs from Key1 and would start receiving LACPDUs from Key2. On both the nodes, cluster LA is connected to Key2 and I3 and I4 will be UP and channel on both the nodes would be UP.

Case 3: Both Switch1 and Switch2 are UP

In this case, it is possible that cluster LA on node1 chooses Key1 as its partner and cluster LA on node2 chooses Key2 as its partner. This means that I1 on node1 and I4 on node2 are receiving traffic which is undesirable. This can happen because the LACP state machine is node-level and chooses its partners on first-come first-serve basis.

To solve these concerns, link redundancy of dynamic cluster LA is supported. To configure link redundancy on a channel or interface, you must enable it and optionally specify the threshold throughput as follows:

set channel CLA/1 -linkRedundancy ON -lrMinThroughput <positive_integer>

The throughput of the partner channels is checked against the configured threshold throughput. The partner channel that satisfies the threshold throughput is selected in first-in-first-out (FIFO) manner. If none of the partner channel meets the threshold, or if threshold throughput is not configured, the partner channel with the maximum number of links is selected.

Note: The threshold throughput can be configured from NetScaler 11 onwards.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing the NetScaler Cluster

Dec 16, 2015

After you have created a cluster and configured the required traffic distribution mechanism, the cluster is able to serve traffic. During the lifetime of the cluster, you can perform cluster tasks such as configuring nodegroups, disabling nodes of a cluster, discovering NetScaler appliances, viewing statistics, synchronizing cluster configurations, cluster files, and the time across the nodes, and upgrading or downgrading the software of cluster nodes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Linksets

Dec 16, 2015

Linksets must be used when some cluster nodes are not physically connected to the external network. In such a cluster topology, the unconnected cluster nodes use the interfaces specified in the linkset to communicate with the external network through the cluster backplane. Linksets are typically used in scenarios when the connecting devices have insufficient ports to connect the cluster nodes.

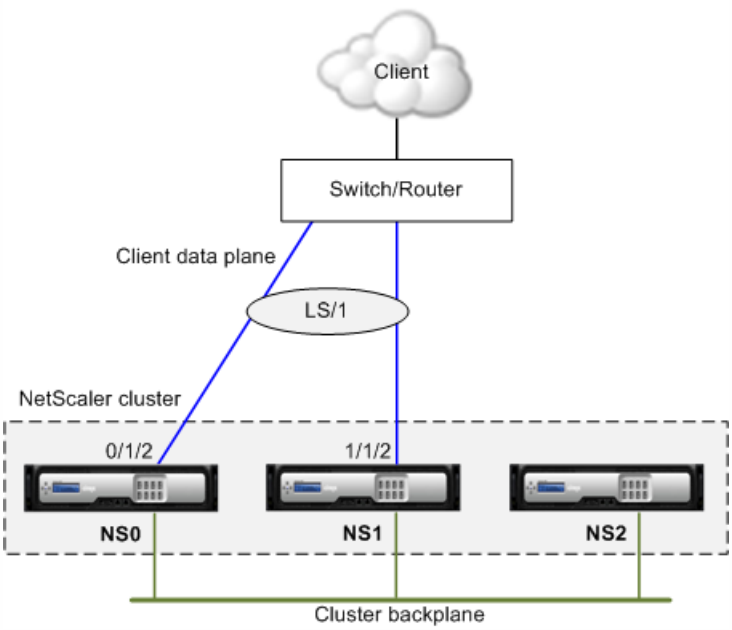
Note: Linksets are a mandatory configuration in the following scenarios:

- For deployments that require MAC-Based Forwarding (MBF).
- To improve manageability of ACL and L2 policies involving interfaces. You must define a linkset of the interfaces and add ACL and L2 policies based on linksets.

Linksets must be configured only through the cluster IP address.

For example, consider a three node cluster where the upstream switch has only two ports available. Using linksets, you can connect two nodes to the switch and leave the third node unconnected. In the following figure, a linkset (LS/1) is formed by binding the interfaces 0/1/2 and 1/1/2. NS2 is the unconnected node of the cluster.

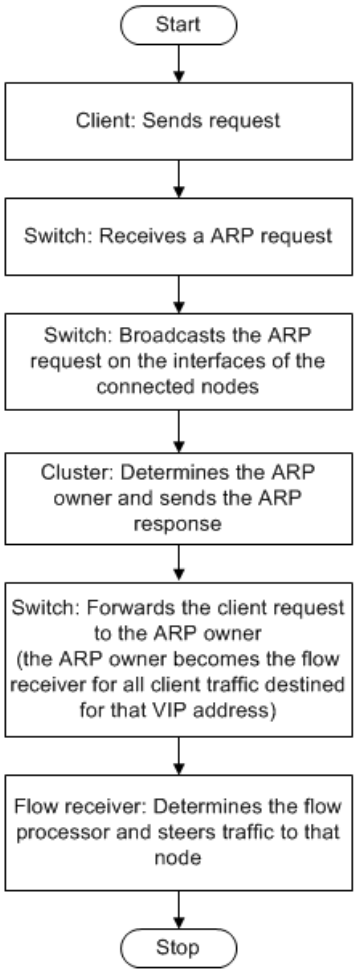
Figure 1. Linksets topology



The linkset informs NS2 that it can use interfaces 0/1/2 and 1/1/2 to communicate with the network devices. All traffic to and from NS2 is now routed through interfaces 0/1/2 or 1/1/2.

Figure 2. Traffic distribution flow using linksets

© 1999-2017 Citrix Systems, Inc. All rights reserved.



1. Log on to the cluster IP address.
2. Create a linkset.
   add linkset <id>

   Example
   > add linkset LS/1
3. Bind the required interfaces to the linkset. Make sure the interfaces are not used for the cluster backplane.
   bind linkset <id> -ifnum <interface_name> ...

   Example
   > bind linkset LS/1 -ifnum 0/1/2 1/1/2
4. Verify the linkset configurations.
   show linkset <id>

   Example
   > show linkset LS/1
   Note: You can bind the linkset to a VLAN by using the bind vlan command. The interfaces of the linkset are automatically bound to the VLAN.

1. Log on to the cluster IP address.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Navigate to System > Network > Linksets.
3. In the details pane, click Add.
4. In the Create Linkset dialog box:
   1. Specify the name of the linkset by setting the Linkset parameter.
   2. Specify the Interfaces to be added to the linkset and click Add. Repeat this step for each interface you want to add to the linkset.
5. Click Create, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Nodegroups for Spotted and Partially-Striped Configurations

May 25, 2015

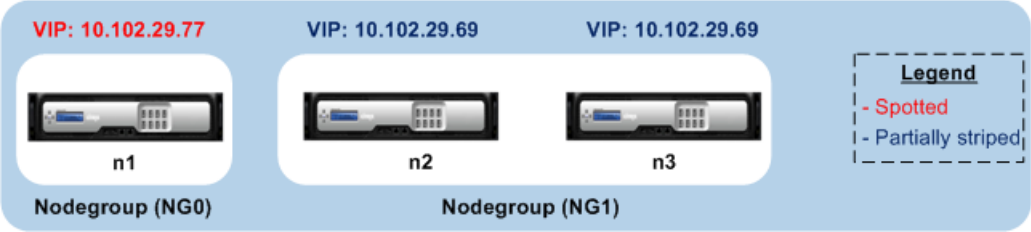
By virtue of the default cluster behavior, all configurations performed on the cluster IP address are available on all nodes of the cluster. However, there might be cases where you need some configurations to be available only on specific cluster nodes.

You can achieve this requirement by defining a nodegroup that includes the specific cluster nodes, and then binding the configuration to that nodegroup. This ensures that the configuration is active only on those cluster nodes. These configurations are called partially-striped or spotted (if active only one a single node). For more information, see Striped, Partially Striped, and Spotted Configurations.

For example, consider a cluster with three nodes. You create a nodegroup NG0 that includes node n1 and another nodegroup NG1 that includes n2 and n3. Bind load balancing virtual servers .77 to NG0 and load balancing virtual server .69 to NG1.

This means that virtual server .77 will be active only on n1 and consequently only n1 will receive traffic that is directed to .77. Similarly, virtual server .69 will be active only on nodes n2 and n3 and consequently only n2 and n3 will receive traffic that is directed to .69.

Figure 1. NetScaler cluster with nodegroups configured for spotted and partial-striped configurations



The entities or configurations that you can bind to a nodegroup are:

- Load balancing, content switching, cache redirection, authentication (AAA) virtual servers
  Note: FTP load balancing virtual servers cannot be bound to nodegroups.
- VPN virtual server (Supported from NetScaler 10.5 Build 50.10 onwards)
- Global Server Load Balancing (GSLB) sites and other GSLB entities (Supported from NetScaler 10.5 Build 52.11 onwards)
- Limit identifiers and stream identifiers

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Behavior of Nodegroups

Mar 18, 2015

Due to the interoperability of nodegroups with different NetScaler features and entities, there are some behavioral aspects to be noted. Nodes in a nodegroup can also be backed up. Read on for more information.

General behavior of a cluster nodegroup

- A nodegroup that has entities bound to it cannot be removed.
- A cluster node that belongs to a nodegroup with entities bound to it, cannot be removed.
- A cluster instance that has nodegroups with entities bound to it, cannot be removed.
- You cannot add an entity that has a dependency on another entity that is not part of the nodegroup. If you need to do so, first remove the dependency. Then, add both the entities to the nodegroup and reassociate the entities.
  Examples:
  - Assume you have a virtual server, VS1, whose backup is virtual server VS2. To add VS1 to a nodegroup, first make sure that VS2 is removed as the backup server of VS1. Then, bind each server individually to the nodegroup, and then configure VS2 as the backup for VS1.
  - Assume you have a content switching virtual server, CSVS1, whose target load balancing virtual server is LBVS1. To add CSVS1 to a nodegroup, first remove LBVS1 as the target. Then, bind each server individually to the nodegroup, and then configure LBVS1 as the target.
  - Assume you have a load balancing virtual server, LBVS1, that has a policy which invokes another load balancing virtual server, LBVS2. To add either one of the virtual servers, first remove the association. Then, bind each server individually to the nodegroup, and then reassociate the virtual servers.
- You cannot bind an entity to a nodegroup that has no nodes and that has the strict option enabled. Consequently, you cannot unbind the last node of a nodegroup that has entities bound to it and that has the strict option enabled
- The strict option cannot be modified for a nodegroup that has no nodes but has entities bound to it.

Backing up Nodes in a Nodegroup

By default, a nodegroup is designed to provide back up nodes for members of a nodegroup. If a nodegroup member goes down, a cluster node that is not a member of the nodegroup dynamically replaces the failed node. This node is called the replacement node.

Note: For a single-member nodegroup, a backup node is automatically preselected when an entity is bound to the nodegroup.
When the original member of the nodegroup comes up, the replacement node, by default, is replaced by the original member node.

From NetScaler 10.5 Build 50.10 onwards, however, the NetScaler allows you to change this replacement behavior. When you enable the sticky option, the replacement node is retained even after the original member node comes up. The original node takes over only when the replacement node goes down.

You can also disable the backup functionality. To do this, you must enable the strict option. In this scenario, when a nodegroup member goes down, no other cluster node is picked up as a backup node. The original node continues being part of the nodegroup when it comes up. This option ensures that entities bound to a nodegroup are active only on nodegroup members.

Note: The strict and sticky option can be set only when creating a nodegroup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Nodegroups for Spotted and Partially-Striped Configurations

Mar 31, 2015

To configure a nodegroup for spotted and partially-striped configurations you must first create a nodegroup and then bind the required nodes to the nodegroup. You must then associate the required entities to that nodegroup. The entities that are bound to the nodegroup will be:

- Spotted - If bound to a nodegroup that has a single node.
- Partially striped - If bound to a nodegroup that has more than one node.

Some points to remember:

- GSLB is supported on a cluster only when GSLB sites are bound to nodegroups that have a single cluster node. For more information, see Setting Up GSLB in a Cluster.
- NetScaler Gateway is supported on a cluster only when the VPN virtual servers are bound to nodegroups that have a single cluster node. The sticky option must be enabled on the nodegroup.
- For versions prior to NetScaler 11, application firewall is supported only on individual cluster nodes (spotted configuration). Application firewall profiles can be associated only with virtual servers that are bound to nodegroups that have a single cluster node. This means that application You are not allowed to do the following:
  - Bind application firewall profiles to striped or partially striped virtual servers.
  - Bind the policy to a global bind point or to user-defined policy labels.
  - Unbind, from a nodegroup, a virtual server that has application firewall profiles.
- NetScaler 11 introduced application firewall support for striped and partially-striped configurations. For more information, see Application Firewall Support for Cluster Configurations.

Check NetScaler Features Supported in a Cluster to see the NetScaler versions from which GSLB, NetScaler Gateway, and application firewall are supported in a cluster.

To configure a nodegroup by using the command line interface

1. Log on to the cluster IP address.
2. Create a nodegroup. Type:
   add cluster nodegroup <name> -strict (**YES** | **NO**)

   Example
   > add cluster nodegroup NG0 -strict YES
3. Bind the required nodes to the nodegroup. Type the following command for each member of the nodegroup:
   bind cluster nodegroup <name> -node <nodeId>

   Example: To bind nodes with IDs 1, 5, and 6.
   > bind cluster nodegroup NG0 -node 1
   > bind cluster nodegroup NG0 -node 5
   > bind cluster nodegroup NG0 -node 6
4. Bind the entity to the nodegroup. Type the following command once for every entity that you want to bind:
   bind cluster nodegroup <name> (-vServer <string> | -identifierName <string> | -gslbSite <string> -service <string>)

   Note: The gslbSite and service parameters are available from NetScaler 10.5 onwards.
   Example: To bind virtual servers VS1 and VS2 and rate limit identifier named identifier1.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> bind cluster nodegroup NG0 -vServer VS1
> bind cluster nodegroup NG0 -vServer VS2
> bind cluster nodegroup NG0 -identifierName identifier1
```

5. Verify the configurations by viewing the details of the nodegroup. Type:
   show cluster nodegroup <name>

   Example

   ```
   > show cluster nodegroup NG0
   ```

To configure a nodegroup by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to System > Cluster > Node Groups.
3. In the details pane, click Add.
4. In the Create Node Group dialog box, configure the nodegroup:
   1. Under Cluster Nodes, click the Add button.
      - The Available list displays the nodes that you can bind to the nodegroup and the Configured list displays the nodes that are bound to the nodegroup.
      - Click the + sign in the Available list to bind the node. Similarly, click the - sign in the Configured list to unbind the node.
   2. Under Virtual Servers, select the tab corresponding to the type of virtual server that you want to bind to the nodegroup. Click the Add button.
      - The Available list displays the virtual servers that you can bind to the nodegroup and the Configured list displays the virtual servers that are bound to the nodegroup.
      - Click the + sign in the Available list to bind the virtual server. Similarly, click the - sign in the Configured list to unbind the virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Redundancy for Nodegroups

Mar 18, 2015

Note: Supported from NetScaler 10.5 Build 52.1115.e onwards.

Nodegroups can be configured such that when one nodegroup goes down, another nodegroup can take over and process traffic. For example, when a nodegroup NG1 goes down, NG2 takes over.

Note: This functionality can be used to configure datacenter redundancy where each nodegroup is configured as a datacenter.

To achieve this use case, cluster nodes must be logically grouped into nodegroups, where some nodegroups must be configured as ACTIVE and others as SPARE. The active nodegroup with the highest priority (that is, the lowest priority number) is made operationally active and therefore serves traffic. When a node from this operationally active nodegroup goes down, the node count of this nodegroup is compared with the node count of the other active nodegroups in order of their priority. If a nodegroup has a higher or equal node count, that nodegroup is made operationally active. Else, the spare nodegroups are checked.

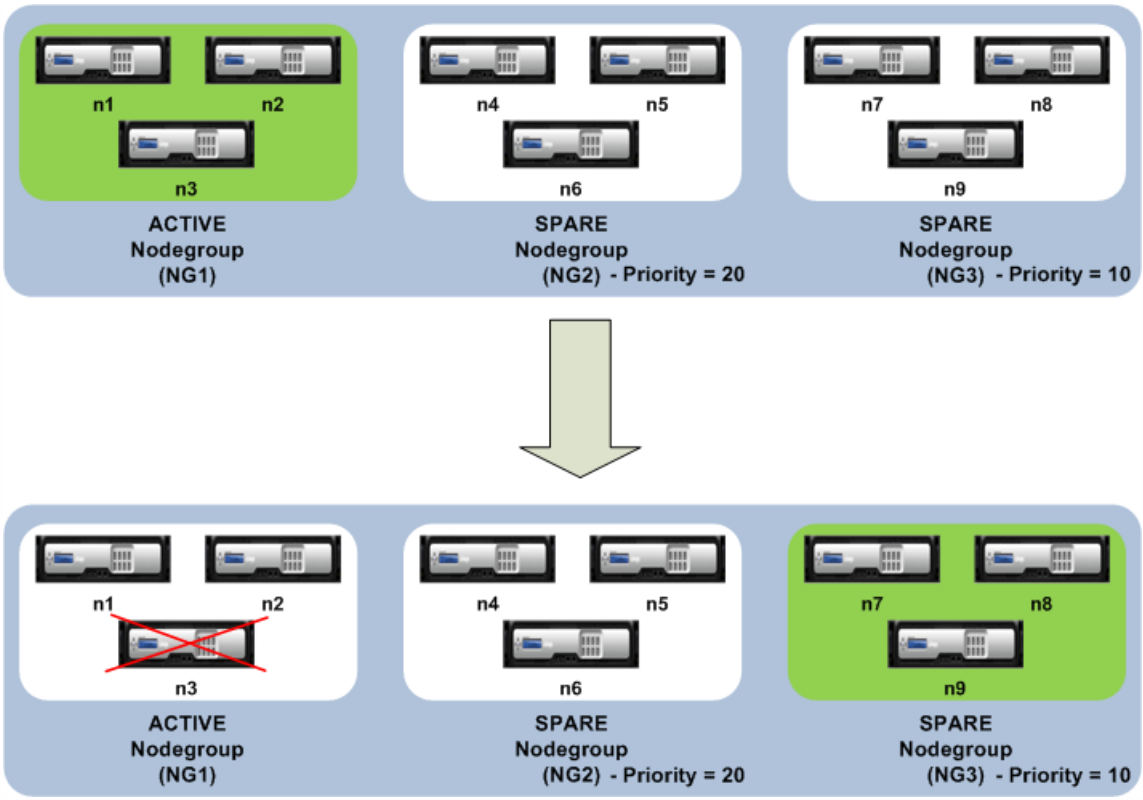
Note:

- Only one state-specific nodegroup can be active at a given point in time.
- A cluster node inherits the state of the nodegroup. So, if a node with "SPARE" state is added to nodegroup with state as "ACTIVE", the node automatically behaves as an active node.
- The preemption parameter that is defined for the cluster instance decides whether the initial active nodegroup will take control when the it comes up again.
- A spare node group can take up a node group and host active traffic when an active node group goes down.

The following figure shows a nodegroup setup that has nodegroup redundancy defined. NG1 is initially the active nodegroup. When it loses one of the nodes, the spare nodegroup (NG3) with the highest priority starts serving traffic.

Figure 1. NetScaler cluster with nodegroup redundancy configured

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Configuring redundancy for nodegroups

1. Log on to the cluster IP address.
2. Create the active nodegroup and bind the required cluster nodes.
   add cluster nodegroup NG1 -state ACTIVE

   bind cluster nodegroup NG1 -node n1

   bind cluster nodegroup NG1 -node n2

   bind cluster nodegroup NG1 -node n3

3. Create the spare nodegroup and bind the requisite nodes.
   add cluster nodegroup NG2 -state SPARE -priority 20

   bind cluster nodegroup NG2 -node n4

   bind cluster nodegroup NG2 -node n5

   bind cluster nodegroup NG2 -node n6

4. Create another spare nodegroup and bind the requisite nodes.
   add cluster nodegroup NG3 -state SPARE -priority 10

   bind cluster nodegroup NG3 -node n7

   bind cluster nodegroup NG3 -node n8

   bind cluster nodegroup NG3 -node n9

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Disabling Steering on the Cluster Backplane

Jun 17, 2015

Note: Supported from NetScaler 11 onwards.

The default behavior of a NetScaler cluster is to direct the traffic that it receives (flow receiver) to another node (flow processor) that must then process the traffic. This process of directing the traffic from flow receiver to flow processor occurs over the cluster backplane and is called steering.

If required, you can disable steering so that the process becomes local to the flow receiver and therefore makes the flow receiver as the flow processor. Such a configuration setup can come handy when you have a high latency link.

Note: This configuration is applicable only for striped virtual servers.

- For partially striped virtual servers, if the flow receiver is a non-owner node, the traffic is steered to a owner node. If however, the flow receiver is a owner node, then steering is disabled.
- For spotted virtual servers, flow receiver is the flow processor, and hence there is no need for steering.

Some points to remember when disabling the steering mechanism:

- Striped SNIPs are not supported as steering is disabled.
- MPTCP and FTP does not work.
- L2 mode must be disabled.
- If USIP is enabled, traffic may not reach back to the same node as the steering is disabled.
- Traffic that is directed to the cluster IP address is steered to the configuration coordinator.
- When a node joins or leaves a cluster, it is possible that more than 1/N connections will be affected. This is due to the fact that a change in the nodes available, may cause the routes to be rehashed. As a result, the traffic will be routed to another node and due to the non-availability of steering, the traffic will not be processed.

Steering can be disabled at the individual virtual server level or at the global level. The global configuration takes precedence over the virtual server setting.

Disabling backplane steering for all striped virtual servers

Configured at cluster instance level. Traffic meant for any striped virtual server will not be steered on cluster backplane.

```
add cluster instance <clId> -processLocal ENABLED
```

Disabling backplane steering for a specific striped virtual server

Configured on a striped virtual server. Traffic meant for the virtual server will not be steered on cluster backplane.

```
add lb vserver <name> <serviceType> -processLocal ENABLED
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Synchronizing Cluster Configurations

Feb 13, 2015

NetScaler configurations that are available on the configuration coordinator are synchronized to the other nodes of the cluster when:

- A node joins the cluster
- A node rejoins the cluster
- A new command is executed through the cluster IP address.

Additionally, you can forcefully synchronize the configurations that are available on the configuration coordinator (full synchronization) to a specific cluster node. Make sure you synchronize one cluster node at a time, otherwise the cluster can get affected.

To synchronize cluster configurations by using the command line interface

At the command prompt of the appliance on which you want to synchronize the configurations, type:

```
force cluster sync
```

To synchronize cluster configurations by using the configuration utility

1. Log on to the appliance on which you want to synchronize the configurations.
2. Navigate to System > Cluster.
3. In the details pane, under Utilities, click Force cluster sync.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Synchronizing Time Across Cluster Nodes

Mar 31, 2015

The cluster uses Precision Time Protocol (PTP) to synchronize the time across cluster nodes. PTP uses multicast packets to synchronize the time. If there are some issues in time synchronization, you must disable PTP and configure Network Time Protocol (NTP) on the cluster.

To enable/disable PTP by using the command line interface

At the command prompt of the cluster IP address, type:

```
set ptp -state disable
```

To enable/disable PTP by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to System > Cluster.
3. In the details pane, under Utilities, click Configure PTP Settings.
4. In the Enable/Disable PTP dialog box, select whether you want to enable or disable PTP.
5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Synchronizing Cluster Files

Feb 26, 2016

The files available on the configuration coordinator are called cluster files. These files are automatically synchronized on the other cluster nodes when the node is added to the cluster and periodically, during the lifetime of the cluster. Additionally, you can manually synchronize the cluster files.

The directories and files from the configuration coordinator that are synchronized are:

- /nsconfig/ssl/
- /var/netscaler/ssl/
- /var/vpn/bookmark/
- /nsconfig/dns/
- /nsconfig/htmlinjection/
- /netscaler/htmlinjection/ens/
- /nsconfig/monitors/
- /nsconfig/nstemplates/
- /nsconfig/ssh/
- /nsconfig/rc.netscaler
- /nsconfig/resolv.conf
- /nsconfig/inetd.conf
- /nsconfig/syslog.conf
- /nsconfig/snmpd.conf
- /nsconfig/ntp.conf
- /nsconfig/httpd.conf
- /nsconfig/sshd_config
- /nsconfig/hosts
- /nsconfig/enckey
- /var/nslw.bin/etc/krb5.conf
- /var/nslw.bin/etc/krb5.keytab
- /var/lib/likewise/db/
- /var/download/
- /var/wi/tomcat/webapps/
- /var/wi/tomcat/conf/Catalina/localhost/
- /var/wi/java_home/lib/security/cacerts
- /var/wi/java_home/jre/lib/security/cacerts
- /nsconfig/license/
- /nsconfig/rc.conf

Tip

Files (certificates and key files) that are copied to the cluster configuration coordinator manually (or through the shell) are not automatically available on the other cluster nodes. You must execute the "sync cluster files" command from the cluster IP address before executing a command that depends on these file(s).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To synchronize cluster files by using the command line interface

At the command prompt of the cluster IP address, type:

```
> sync cluster files <mode>
```

To synchronize cluster files by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to System > Cluster.
3. In the details pane, under Utilities, click Synchronize cluster files.
4. In the Synchronize cluster files dialog box, select the files to be synchronized in the Mode drop-down box.
5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing the Statistics of a Cluster

Feb 13, 2015

You can view the statistics of a cluster instance and cluster nodes to evaluate the performance or to troubleshoot the operation of the cluster.

To view the statistics of a cluster instance by using the command line interface

At the command prompt of the cluster IP address, type:

stat cluster instance <clId>

To view the statistics of a cluster node by using the command line interface

At the command prompt of the cluster IP address, type:

stat cluster node <nodeid>

Note: When executed from the cluster IP address, this command displays the cluster level statistics. However, when executed from the NSIP address of a cluster node, the command displays node level statistics.

To view the statistics of a cluster instance by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to System > Cluster.
3. In the details pane, in the center of the page, click Statistics.

To view the statistics of a cluster node by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to System > Cluster > Nodes.
3. In the details pane, select a node and click Statistics to view the statistics of the node. To view the statistics of all the nodes, click Statistics without selecting a specific node.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Discovering NetScaler Appliances

Jun 18, 2015

You can discover the appliances present in the same subnet as the current node. The required discovered appliances can then be selectively added to the cluster. This operation can be performed to either create a new cluster or to add nodes to an existing cluster.

Note:

- The discover operation can be performed only through the configuration utility.
- This operation cannot discover NetScaler appliances from different networks.
- When performing this operation to add nodes to an existing cluster, the L3 VLAN configurations will be cleared from the node. You must make sure to define these configurations once the appliance is added to the cluster.

To discover appliances by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to System > Cluster > Nodes.
3. In the details pane, at the bottom of the page, click Discover NetScalers.
4. In the Discover NetScalers dialog box, set the following parameters:
   - IP address range - Specify the range of IP addresses within which you want to discover appliances. For example, you can search for all NSIP addresses between 10.102.29.4 to 10.102.29.15 by specifying this option as 10.102.29.4 - 15.
   - Backplane interface - Specify the interfaces to be used as the backplane interface. This is an optional parameter. If you do not specify this parameter, you must update it after the node is added to the cluster.
5. Click OK.
6. Select the appliances that you want to add to the cluster.
7. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Disabling a Cluster Node

Feb 27, 2015

You can temporarily remove a node from a cluster by disabling the cluster instance on that node. A disabled node is not synchronized with the cluster configurations. When the node is enabled again, the cluster configurations are automatically synchronized on it. For more information, see Cluster Synchronization.

A disabled node cannot serve traffic and all existing connections on this node are terminated.

Note: If the configurations of a disabled non-configuration coordinator node are modified (through the NSIP address of the node), the configurations are not automatically synchronized on that node. You must manually synchronize the configurations as described in Synchronizing Cluster Configurations.

To disable a cluster node by using the command line interface

At the command prompt of the node that you want to disable, type:

```
disable cluster instance <clId>
```

Note: To disable the cluster, run the disable cluster instance command on the cluster IP address.

To disable a cluster node by using the configuration utility

1. On the node that you want to disable, navigate to System > Cluster, and click Manage Cluster.
2. In the Configure cluster instance dialog box, unselect the Enable cluster instance check box.
   Note: To disable the cluster instance on all the nodes, perform the above procedure on the cluster IP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Removing a Cluster Node

Jun 17, 2015

When a node is removed from the cluster, the cluster configurations are cleared from the node (by internally executing the clear ns config -extended command). The SNIP addresses, MTU settings of the backplane interface, and all VLAN configurations (except the default VLAN and NSVLAN) are also cleared from the appliance.

Note:

- If the deleted node was the cluster configuration coordinator, another node is automatically selected as the cluster configuration coordinator, and the cluster IP address is assigned to that node. All the current cluster IP address sessions will be invalid and you will have to start a new session.
- To delete the whole cluster, you must remove each node individually. When you remove the last node, the cluster IP address(es) are deleted.
- When an active node is removed, the traffic serving capability of the cluster is reduced by one node. Existing connections on this node are terminated.

To remove a cluster node by using the command line interface

For NetScaler 10.1 and later versions

Log on to the cluster IP address and at the command prompt, type:

rm cluster node <nodeId>

Note: If the cluster IP address is unreachable from the node, execute the rm cluster instance command on the NSIP address of that node itself.

For NetScaler 10

1. Log on to the node that you want to remove from the cluster and remove the reference to the cluster instance.
   rm cluster instance <clId>

   save ns config

2. Log on to the cluster IP address and remove the node from which you removed the cluster instance.
   rm cluster node <nodeId>

   save ns config

   Make sure you do not run the rm cluster node command from the local node as this results in inconsistent configurations between the configuration coordinator and the node.

To remove a cluster node by using the configuration utility

On the cluster IP address, navigate to System > Cluster > Nodes, select the node you want to remove and click Remove.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Removing a Node from a Cluster Deployed Using Cluster Link Aggregation

Feb 13, 2015

To remove a node from a cluster that uses cluster link aggregation as the traffic distribution mechanism, you must make sure that the node is made passive so that it does not receive any traffic and then, on the upstream switch, remove the corresponding interface from the channel.

For detailed information on cluster link aggregation, see Using Cluster Link Aggregation.

To remove a node from a cluster that uses cluster link aggregation as the traffic distribution mechanism

1. Log on to the cluster IP address.
2. Set the state of the cluster node that you want to remove to PASSIVE.
   set cluster node <nodeId> -state PASSIVE
3. On the upstream switch, remove the corresponding interface from the channel by using switch-specific commands.
   Note: You do not have to manually remove the nodes interface on the cluster link aggregation channel. It is automatically removed when the node is deleted in the next step.
4. Remove the node from the cluster.
   rm cluster node <nodeId>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cluster Setup and Usage Scenarios

Jun 17, 2015

This section aims at explaining some scenarios in which the NetScaler cluster can be setup and also how it can be configured for different features and network topologies. These are just some scenarios that we have documented. Provide feedback if you want some other scenarios to be included.

- Creating a Two-Node Cluster
- Migrating an HA Setup to a Cluster Setup
- Transitioning between a L2 and L3 Cluster [From NetScaler 11 onwards]
- Setting Up GSLB in a Cluster [From NetScaler 10.5 onwards]
- Using Cache Redirection in a Cluster
- Using L2 Mode in a Cluster Setup
- Using Cluster LA Channel with Linksets
- Backplane on LA Channel
- Common Interface for Client and Server and Dedicated Interfaces for Backplane
- Common Switch for Client, Server, and Backplane
- Common Switch for Client and Server and Dedicated Switch for Backplane
- Different Switch for Every Node
- Sample Cluster Configurations

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating a Two-Node Cluster

Feb 17, 2015

A two-node cluster is an exception to the rule that a cluster is functional only when a minimum of (n/2 +1) nodes, where n is the number of cluster nodes, are able to serve traffic. If that formula were applied to a two-node cluster, the cluster would fail if one node went down (n/2 +1=2).

A two-node cluster is functional even if only one node is able to serve traffic.

Creating a two node cluster is the same as creating any other cluster. You must add one node as the configuration coordinator and the other node as the other cluster node.

Note: Incremental configuration synchronization is not supported in a two-node cluster. Only full synchronization is supported.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Migrating an HA Setup to a Cluster Setup

Jun 15, 2015

Migrating an existing high availability (HA) setup to a cluster setup requires you to first remove the NetScaler appliances from the HA setup and create a backup of the HA configuration file. You must then use the two appliances to create a cluster and upload the backed-up configuration file to the cluster.

Note:

- Before uploading the backed-up HA configuration file to the cluster, you must modify it to make it cluster compatible. Refer to the relevant step of the procedure below.
- Use the **batch -f <backup_filename>** command to upload the backed-up configuration file.

The above approach is a very basic migration solution which results in downtime for the deployed application. As such, it must be used only in deployments where there is no consideration to application availability.

However, in most deployments, the availability of the application is of paramount importance. For such cases, you must use the approach where an HA setup can be migrated to a cluster setup without any resulting downtime. In this approach, an existing HA setup is migrated to a cluster setup by first removing the secondary appliance and using that appliance to create a single-node cluster. After the cluster becomes operational and serves traffic, the primary appliance of the HA setup is added to the cluster.

To convert a HA setup to cluster setup (without any downtime) by using the command line interface

Let us consider the example of a HA setup with primary appliance (NS1) - 10.102.97.131 and secondary appliance (NS2) - 10.102.97.132.

1. Make sure the configurations of the HA pair are stable.
2. Log on to any one of the HA appliances, go to the shell, and create a copy of the ns.conf file (for example, ns_backup.conf).
3. Log on to the secondary appliance, NS2, and clear the configurations. This operation removes NS2 from the HA setup and makes it a standalone appliance.
   > clear ns config full
   Note:
   - This step is required to make sure that NS2 does not start owning VIP addresses, now that it is a standalone appliance.
   - At this stage, the primary appliance, NS1, is still active and continues to serve traffic.
4. Create a cluster on NS2 (now no longer a secondary appliance) and configure it as a PASSIVE node.
   > add cluster instance 1

   > add cluster node 0 10.102.97.132 -state PASSIVE -backplane 0/1/1

   > add ns ip 10.102.97.133 255.255.255.255 -type CLIP

   > enable cluster instance 1

   > save ns config

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> reboot -warm

5. Modify the backed-up configuration file as follows:
   1. Remove the features that are not supported on a cluster. For the list of unsupported features, see NetScaler Features Supported by a Cluster. This is an optional step. If you do not perform this step, the execution of unsupported commands will fail.
   2. Remove the configurations that have interfaces, or update the interface names from the c/u convention to the n/c/u convention.

      Example

      > add vlan 10 -ifnum 0/1

      should be changed to

      > add vlan 10 -ifnum 0/0/1 1/0/1
   3. The backup configuration file can have SNIP addresses or MIP addresses. These addresses are striped on all the cluster nodes. It is recommended that you add spotted IP addresses for each node.

      Example

      > add ns ip 1.1.1.1 255.255.255.0 -ownerNode 0

      > add ns ip 1.1.1.2 255.255.255.0 -ownerNode 1
   4. Update the hostname to specify the owner node.

      Example

      > set ns hostname ns0 -ownerNode 0

      > set ns hostname ns1 -ownerNode 1
   5. Change all other relevant networking configuration that depend on spotted IPs. For example, L3 VLAN, RNAT configuration which uses SNIPs as NATIP, INAT rules that refers to SNIPs/MIPs).
6. On the cluster, do the following:
   1. Make the topological changes to the cluster by connecting the cluster backplane, the cluster link aggregation channel, and so on.
   2. Apply configurations from the backed-up and modified configuration file to the configuration coordinator through the cluster IP address.

      > batch -f ns_backup.conf
   3. Configure external traffic distribution mechanisms like ECMP or cluster link aggregation.
7. Switch the traffic from the HA setup to the cluster.
   1. Log on to the primary appliance, NS1, and disable all the interfaces on it.

      > disable interface <interface id>
   2. Log on to the cluster IP address and configure NS2 as an ACTIVE node.

      > set cluster node 0 -state ACTIVE

   Note: There might be a small amount (in the order of seconds) of downtime between disabling the interfaces and making the cluster node active.
8. Log on to the primary appliance, NS1, and remove it from the HA setup.
   1. Clear all the configurations. This operation removes NS1 from the HA setup and makes it a standalone appliance.

      > clear ns config full
   2. Enable all the interfaces.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> enable interface <interface id>
```

9. Add NS1 to the cluster.
   1. Log on to the cluster IP address and add NS1 to the cluster.

      ```
      > add cluster node 1 10.102.97.131 -state PASSIVE -backplane 1/1/1
      ```

   2. Log on to NS1 and join it to the cluster by sequentially executing the following commands:

      ```
      > join cluster -clip 10.102.97.133 -password nsroot

      > save ns config

      > reboot -warm
      ```

10. Log on to NS1 and perform the required topological and configuration changes.
11. Log on to the cluster IP address and set NS1 as an ACTIVE node.

    ```
    > set cluster node 1 -state ACTIVE
    ```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Transitioning between a L2 and L3 Cluster

Jun 17, 2015

Note: Supported from NetScaler 11 onwards.

An L2 cluster is one where all the nodes are from the same network and an L3 cluster is one that can include nodes from different networks. You can seamlessly transition from one type of cluster to the other without any downtime for the applications that are deployed on the NetScaler.

- Transitioning a Cluster from L2 to L3
- Transitioning a Cluster from L3 to L2

Transitioning a Cluster from L2 to L3

You can transition to an L3 cluster when you want the cluster to include nodes from other networks.

On the cluster IP address, do the following:

1. Create a nodegroup.
   Example

   ```
   > add cluster nodegroup NG0
   ```

   This nodegroup is used in the next step to group all the nodes from the existing L2 cluster.

2. Transition the L2 cluster to an L3 cluster.
   Example

   ```
   > set cluster instance 1 -inc ENABLED -nodegroup NG0
   ```

   This command achieves the dual purpose of transitioning to L3 cluster and also adding all the nodes of the L2 cluster to the nodegroup.
3. Now, you can add more nodes to the cluster as explained in "Adding a Node to the Cluster".

Transitioning a Cluster from L3 to L2

You can transition to an L2 cluster when you want to retain nodes that belong to a single network.

On the cluster IP address, do the following:

1. Remove the cluster nodes from the networks that you do not want to retain.
   Example

   ```
   > rm cluster node <nodeId>
   ```
2. Transition the L3 cluster to a L2 cluster.
   Example

   ```
   > set cluster instance 1 -inc DISABLED
   ```

   The cluster now includes nodes only of a single network.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting Up GSLB in a Cluster

Jun 17, 2015

Note: Supported from NetScaler 10.5 Build 52.11 onwards.

To set up GSLB in a cluster you must bind the different GSLB entities to a node group. The node group must have a single member node.

Note:

- The parent-child topology of GSLB is not supported in a cluster.
- If you have configured the static proximity GSLB method, make sure that the static proximity database is present on all the cluster nodes. This happens by default if the database file is available at the default location. However, if the database file is maintained in a directory other than /var/netscaler/locdb/, you must manually synch the file to all the cluster nodes.

To set up GSLB in a cluster by using the command line interface

Log on to the cluster IP address and perform the following operations at the command prompt:

1. Configure the different GSLB entities. For information, see Configuring Global Server Load Balancing.
   Note: When creating the GSLB site, make sure that you specify the cluster IP address and public cluster IP address (needed only when the cluster is deployed behind a NAT device). These parameters are required to ensure the availability of the GSLB auto-sync functionality.
   add gslb site <siteName> <siteType> <siteIPAddress> -publicIP <ip_addr> -clip <ip_addr> <publicCLIP>
2. Create a cluster node group.
   add cluster nodegroup <name> [-sticky (YES | NO)]

   Note: Enable the sticky option if you want to set up GSLB based on VPN users.
3. Bind a single cluster node to the node group.
   bind cluster nodegroup <name> -node <nodeId>
4. Bind the local GSLB site to the nodegroup.
   bind cluster nodegroup <name> -gslbSite <string>

   Note: Make sure that the IP address of the local GSLB site IP address is striped (available across all cluster nodes).
5. Bind the ADNS (or ADNS-TCP) service or the DNS (or DNS-TCP) load balancing virtual server to the node group.
   To bind the ADNS service:

   bind cluster nodegroup <name> -service <string>

   To bind the DNS load balancing virtual server:

   bind cluster nodegroup <name> -vServer <string>
6. Bind the GSLB virtual server to the node group.
   bind cluster nodegroup <name> -vServer <string>
7. [Optional] To setup GSLB based on VPN users, bind the VPN virtual server to the GSLB node group.
   bind cluster nodegroup <name> -vServer <string>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

8. Verify the configurations.
   show gslb runningConfig

To set up GSLB in a cluster by using the graphical user interface

Log on to the cluster IP address and perform the following operations in the Configuration tab:

1. Configure the GSLB entities.
   Navigate to Traffic Management > GSLB to perform the required configurations.

2. Create a node group and perform other node group related configurations.
   Navigate to System > Cluster > Node Groups to perform the required configurations.

For the detailed configurations to be performed, see the description provided in the CLI procedure mentioned above.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using Cache Redirection in a Cluster

Feb 17, 2015

Cache redirection in a cluster works in the same way as it does on a standalone NetScaler appliance. The only difference is that the configurations are done on the cluster IP address. For more information on cache redirection, see "Cache Redirection."

Points to remember when using cache redirection in transparent mode on a cluster:

- Before configuring cache redirection, make sure that you have connected all nodes to the external switch and that you have linksets configured. Otherwise, client requests will be dropped.
- When MAC mode is enabled on a load balancing virtual server, make sure MBF mode is enabled on the cluster by using the enable ns mode MBF command. Otherwise, the requests are sent to origin server directly instead of being sent to the cache server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using L2 Mode in a Cluster Setup

Oct 08, 2014

Note: Supported from NetScaler 10.5 and later releases.

To use L2 mode in a cluster setup, you must make sure of the following:

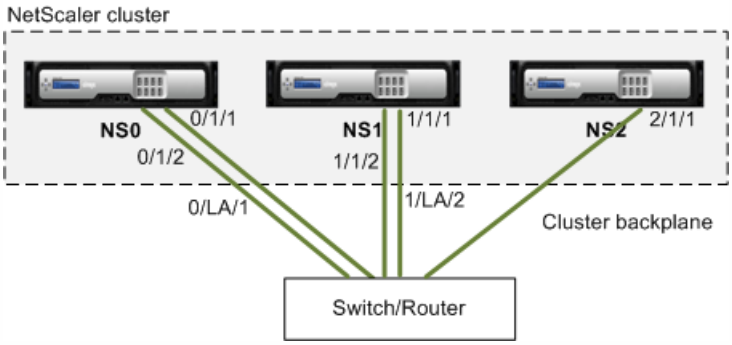
- Spotted IP addresses must be available on all the nodes as required.
- Linksets must be used to communicate with the external network.
- Asymmetric topologies or asymmetric cluster LA groups are not supported.
- Cluster LA group is recommended.
- Traffic is distributed between the cluster nodes only for deployments where services exist.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Backplane on LA Channel

Feb 17, 2015

In this deployment, LA channels are used for the cluster backplane.



NS0 - nodeId: 0, NSIP: 10.102.29.60

NS1 - nodeId: 1, NSIP: 10.102.29.70

NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with the backplane interfaces as LA channels

1. Create a cluster of nodes NS0, NS1, and NS2.
   1. Log on to the first node that you want to add to the cluster and do the following:
      create cluster instance 1
      add cluster node 0 10.102.29.60 -state ACTIVE
      enable cluster instance 1
      add ns ip 10.102.29.61 255.255.255.255 -type CLIP
      save ns config
      reboot -warm
   2. Log on to the cluster IP address and do the following:
      add cluster node 1 10.102.29.70 -state ACTIVE
      add cluster node 2 10.102.29.80 -state ACTIVE
   3. Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.
      join cluster -clip 10.102.29.61 -password nsroot
      save ns config
      reboot -warm

   As seen in the above commands the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

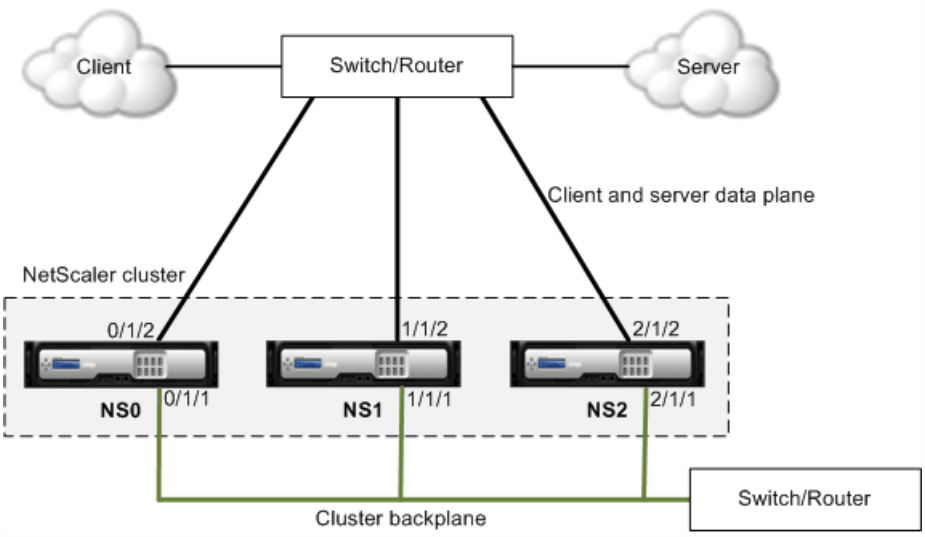
2. Log on to the cluster IP address and do the following:
   1. Create the LA channels for nodes NS0 and NS1.
      add channel 0/LA/1 -ifnum 0/1/1 0/1/2
      add channel 1/LA/2 -ifnum 1/1/1 1/1/2
   2. Configure the backplane for the cluster nodes.
      set cluster node 0 -backplane 0/LA/1
      set cluster node 1 -backplane 1/LA/2
      set cluster node 2 -backplane 2/1/1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Common Interfaces for Client and Server and Dedicated Interfaces for Backplane

Feb 17, 2015

This is a one-arm deployment of the NetScaler cluster. In this deployment, the client and server networks use the same interfaces to communicate with the cluster. The cluster backplane uses dedicated interfaces for inter-node communication.



NS0 - nodeId: 0, NSIP: 10.102.29.60

NS1 - nodeId: 1, NSIP: 10.102.29.70

NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with a common interface for the client and server and a different interface for the cluster backplane

1. Create a cluster of nodes NS0, NS1, and NS2.
   1. Log on to the first node that you want to add to the cluster and do the following:
      create cluster instance 1
      add cluster node 0 10.102.29.60 -state ACTIVE -backplane 0/1/1
      enable cluster instance 1
      add ns ip 10.102.29.61 255.255.255.255 -type CLIP
      save ns config
      reboot -warm
   2. Log on to the cluster IP address and do the following:
      add cluster node 1 10.102.29.70 -state ACTIVE -backplane 1/1/1
      add cluster node 2 10.102.29.80 -state ACTIVE -backplane 2/1/1
   3. Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.
      join cluster -clip 10.102.29.61 -password nsroot
      save ns config

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
reboot -warm
```

As seen in the above commands the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

2. On the cluster IP address, create VLANs for the backplane interfaces and for the client and server interfaces.

```
//For the backplane interfaces
add vlan 10
bind vlan 10 0/1/1 1/1/1 2/1/1

//For the interfaces that are connected to the client and server networks.
add vlan 20
bind vlan 20 0/1/2 1/1/2 2/1/2
```

3. On the switch, create VLANs for the interfaces corresponding to the backplane interfaces and the client and server interfaces. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

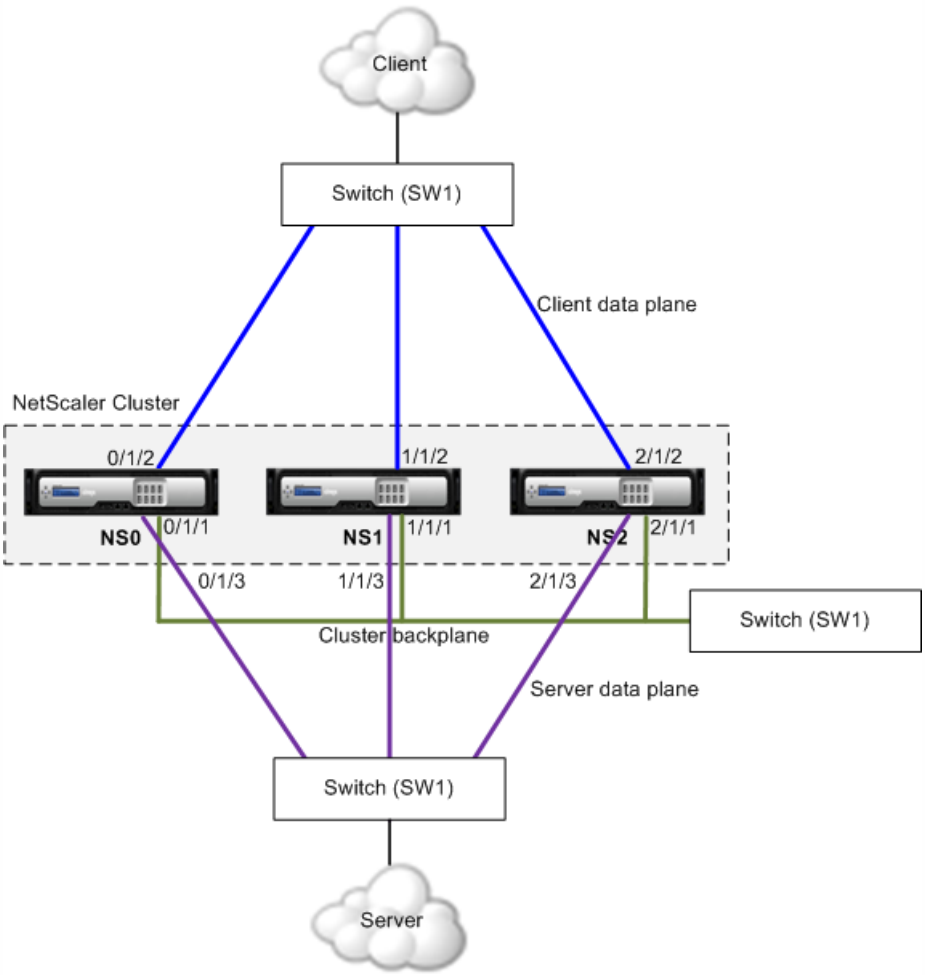
```
//For the backplane interfaces. Repeat for each interface...
interface Ethernet2/47
switchport access vlan 100
switchport mode access
end

//For the interfaces connected to the client and server networks. Repeat for each interface...
interface Ethernet2/47
switchport access vlan 200
switchport mode access
end
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Common Switch for Client, Server, and Backplane

Feb 17, 2015

In this deployment, the client, server, and backplane use dedicated interfaces on the same switch to communicate with the NetScaler cluster.



NS0 - nodeId: 0, NSIP: 10.102.29.60

NS1 - nodeId: 1, NSIP: 10.102.29.70

NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with a common switch for the client, server, and backplane

1. Create a cluster of nodes NS0, NS1, and NS2.
   1. Log on to the first node that you want to add to the cluster and do the following:
      create cluster instance 1
      add cluster node 0 10.102.29.60 -state ACTIVE -backplane 0/1/1
      enable cluster instance 1
      add ns ip 10.102.29.61 255.255.255.255 -type CLIP
      save ns config

© 1999-2017 Citrix Systems, Inc. All rights reserved.

reboot -warm

2. Log on to the cluster IP address and do the following:
   add cluster node 1 10.102.29.70 -state ACTIVE -backplane 1/1/1
   add cluster node 2 10.102.29.80 -state ACTIVE -backplane 2/1/1
3. Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.
   join cluster -clip 10.102.29.61 -password nsroot
   save ns config
   reboot -warm

As seen in the above commands the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

2. On the cluster IP address, create VLANs for the backplane, client, and server interfaces.

```
//For the backplane interfaces
add vlan 10
bind vlan 10 0/1/1 1/1/1 2/1/1

//For the client-side interfaces
add vlan 20
bind vlan 20 0/1/2 1/1/2 2/1/2

//For the server-side interfaces
add vlan 30
bind vlan 30 0/1/3 1/1/3 2/1/3
```

3. On the switch, create VLANs for the interfaces corresponding to the backplane interfaces and the client and server interfaces. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

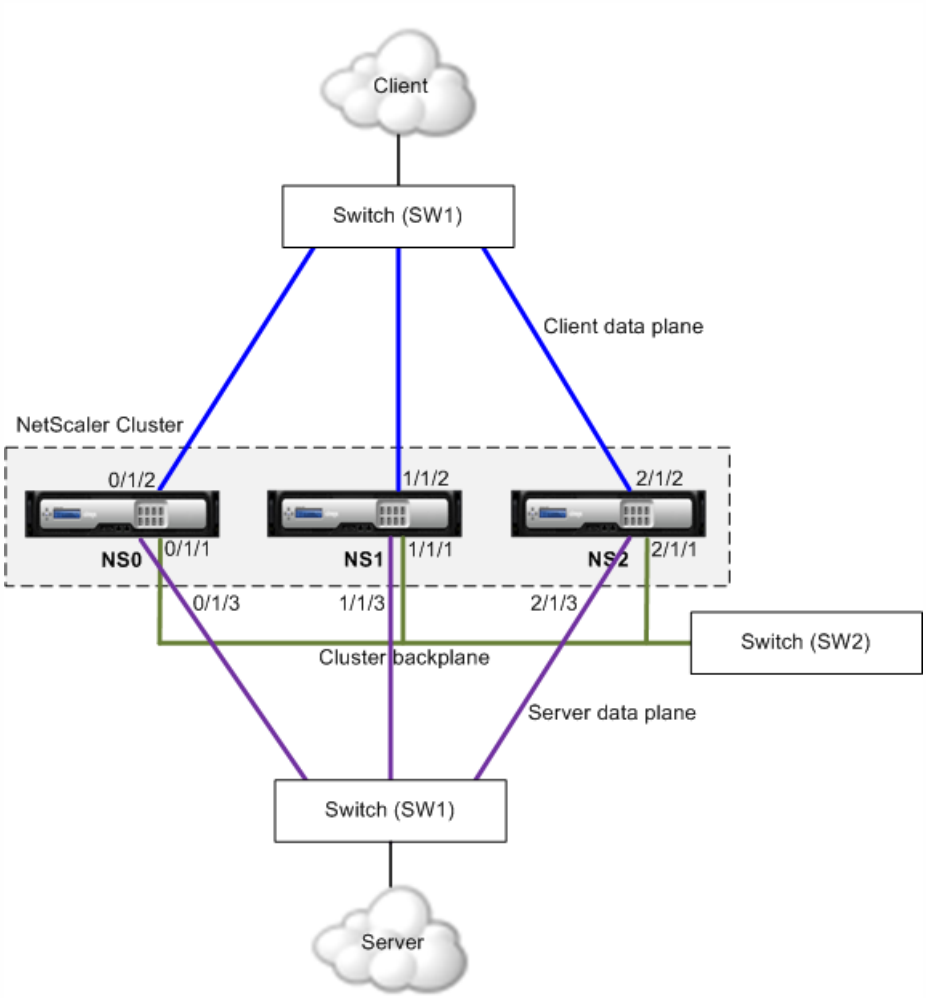
```
//For the backplane interfaces. Repeat for each interface...
interface Ethernet2/47
switchport access vlan 100
switchport mode access
end

//For the client interfaces. Repeat for each interface...
interface Ethernet2/48
switchport access vlan 200
switchport mode access
end

//For the server interfaces. Repeat for each interface...
interface Ethernet2/49
switchport access vlan 300
switchport mode access
end
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Common Switch for Client and Server and Dedicated Switch for Backplane

Feb 17, 2015

In this deployment, the clients and servers use different interfaces on the same switch to communicate with the NetScaler cluster. The cluster backplane uses a dedicated switch for inter-node communication.



NS0 - nodeId: 0, NSIP: 10.102.29.60

NS1 - nodeId: 1, NSIP: 10.102.29.70

NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with the same switch for the clients and servers and a different switch for the cluster backplane

1. Create a cluster of nodes NS0, NS1, and NS2.
   1. Log on to the first node that you want to add to the cluster and do the following:
      create cluster instance 1
      add cluster node 0 10.102.29.60 -state ACTIVE -backplane 0/1/1

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
enable cluster instance 1
add ns ip 10.102.29.61 255.255.255.255 -type CLIP
save ns config
reboot -warm
```

2. Log on to the cluster IP address and do the following:

```
add cluster node 1 10.102.29.70 -state ACTIVE -backplane 1/1/1
add cluster node 2 10.102.29.80 -state ACTIVE -backplane 2/1/1
```

3. Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.

```
join cluster -clip 10.102.29.61 -password nsroot
save ns config
reboot -warm
```

As seen in the above commands the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

2. On the cluster IP address, create VLANs for the backplane, client, and server interfaces.

```
//For the backplane interfaces
add vlan 10
bind vlan 10 0/1/1 1/1/1 2/1/1

//For the client-side interfaces
add vlan 20
bind vlan 20 0/1/2 1/1/2 2/1/2

//For the server-side interfaces
add vlan 30
bind vlan 30 0/1/3 1/1/3 2/1/3
```

3. On the switch, create VLANs for the interfaces corresponding to the backplane interfaces and the client and server interfaces. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

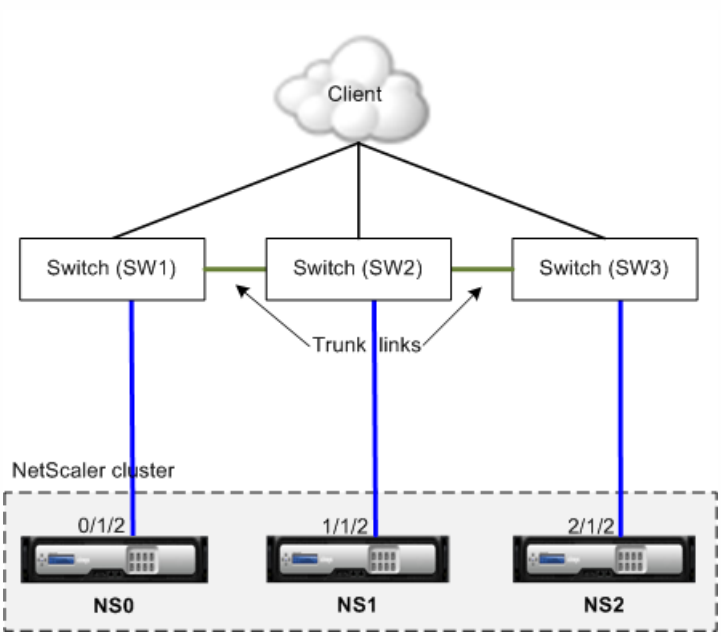
```
//For the backplane interfaces. Repeat for each interface...
interface Ethernet2/47
switchport access vlan 100
switchport mode access
end

//For the client interfaces. Repeat for each interface...
interface Ethernet2/48
switchport access vlan 200
switchport mode access
end

//For the server interfaces. Repeat for each interface...
interface Ethernet2/49
switchport access vlan 300
switchport mode access
end
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Different Switch for Every Node

Feb 17, 2015

In this deployment, each cluster node is connected to a different switch and trunk links are configured between the switches.



The cluster configurations will be the same as the other deployments scenarios. Most of the client-side configurations will be done on the client-side switches.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Sample Cluster Configurations

Feb 09, 2015

The following example can be used to configure a four-node cluster with ECMP, cluster LA, or Linksets.

1. Create the cluster.
   1. Log on to first node.
   2. Add the cluster instance.
      add cluster instance 1
   3. Add the first node to the cluster.
      add cluster node 0 10.102.33.184 -backplane 0/1/1
   4. Enable the cluster instance.
      enable cluster instance 1
   5. Add the cluster IP address.
      add ns ip 10.102.33.185 255.255.255.255 -type CLIP
   6. Save the configurations.
      save ns config
   7. Warm reboot the appliance.
      reboot -warm
2. Add the other three nodes to the cluster.
   1. Log on to cluster IP address.
   2. Add the second node to the cluster.
      add cluster node 1 10.102.33.187 -backplane 1/1/1
   3. Add the third node to the cluster.
      add cluster node 2 10.102.33.188 -backplane 2/1/1
   4. Add the fourth node to the cluster.
      add cluster node 3 10.102.33.189 -backplane 3/1/1
3. Join the added nodes to the cluster. This step is not applicable for the first node.
   1. Log on to each newly added node.
   2. Join the node to the cluster.
      join cluster -clip 10.102.33.185 -password nsroot
   3. Save the configuration.
      save ns config
   4. Warm reboot the appliance.
      reboot -warm
4. Configure the NetScaler cluster through the cluster IP address.

   ```
   // Enable load balancing feature
   enable ns feature lb

   // Add a load balancing virtual server
   add lb vserver first_lbvserver http
   ....
   ....
   ```
5. Configure any one of the following (ECMP, cluster LA, or Linkset) traffic distribution mechanisms for the cluster.
   - **ECMP**

     1. Log on to the cluster IP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Enable the OSPF routing protocol.
   enable ns feature ospf
3. Add a VLAN.
   add vlan 97
4. Bind the interfaces of the cluster nodes to the VLAN.
   bind vlan 97 -ifnum 0/1/4 1/1/4 2/1/4 3/1/4
5. Add a spotted SNIP on each node and enable dynamic routing on it.
   add ns ip 1.1.1.10 255.255.255.0 -ownerNode 0 -dynamicRouting ENABLED
   add ns ip 1.1.1.11 255.255.255.0 -ownerNode 1 -dynamicRouting ENABLED
   add ns ip 1.1.1.12 255.255.255.0 -ownerNode 2 -dynamicRouting ENABLED
   add ns ip 1.1.1.13 255.255.255.0 -ownerNode 3 -dynamicRouting ENABLED
6. Bind one of the SNIP addresses to the VLAN.
   bind vlan 97 -ipAddress 1.1.1.10 255.255.255.0
7. Configure the routing protocol on ZebOS by using vtysh shell.

- **Static cluster LA**

  1. Log on to the cluster IP address.
  2. Add a cluster LA channel.
     add channel CLA/1 -speed 1000
  3. Bind the interfaces to the cluster LA channel.
     bind channel CLA/1 0/1/5 1/1/5 2/1/5 3/1/5
  4. Perform equivalent configuration on the switch.
- **Dynamic cluster LA**

  1. Log on to the cluster IP address.
  2. Add the interfaces to the cluster LA channel.
     set interface 0/1/5 -lacpmode active -lacpkey 5 -lagtype cluster
     set interface 1/1/5 -lacpmode active -lacpkey 5 -lagtype cluster
     set interface 2/1/5 -lacpmode active -lacpkey 5 -lagtype cluster
     set interface 3/1/5 -lacpmode active -lacpkey 5 -lagtype cluster
  3. Perform equivalent configuration on the switch.
- **Linksets.** Assume that the node with nodeId 3 is not connected to the switch. You must configure a linkset so that the unconnected node can use the other node interfaces to communicate with the switch.

  1. Log on to the cluster IP address.
  2. Add a linkset.
     add linkset LS/1
  3. Bind the connected interfaces to the linkset.
     bind linkset LS/1 -ifnum 0/1/6 1/1/6 2/1/6

6. Update the state of the cluster nodes to ACTIVE.
   set cluster node 0 -state ACTIVE
   set cluster node 1 -state ACTIVE
   set cluster node 2 -state ACTIVE
   set cluster node 3 -state ACTIVE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Upgrading or Downgrading the NetScaler Cluster

Feb 19, 2016

All the nodes of a NetScaler cluster must be running the same software version. Therefore, to upgrade or downgrade the cluster, you must upgrade or downgrade each NetScaler appliance of the cluster, one node at a time.

A node that is being upgraded or downgraded is not removed from the cluster. The node continues to be a part of the cluster and serves traffic uninterrupted, except for the down-time when the node reboots after it is upgraded or downgraded.

However, due to software version mismatch among the cluster nodes, configuration propagation is disabled on the cluster and is enabled only after all the cluster nodes are of the same version. Since configuration propagation is disabled during upgrading on downgrading a cluster, you cannot perform any configurations through the cluster IP address during this time.

Important

- Configurations can be lost on downgrading the cluster.
- When you are upgrading a cluster to NetScaler 11.0 build 64.x from an earlier NetScaler 11.0 build, cluster configuration propagation is disabled. This exception arises because the cluster version in build 64.x is different from the one in previous NetScaler 11.0 builds.

  Traditionally, this issue occurred only during an upgrade of a cluster to a different NetScaler version (for example, from 10.5 to 11.0). It must be noted that normally the cluster version matches the NetScaler version.

  Note: Configuration propagation remains disabled until all the cluster nodes are upgraded to Build 64.x.

Points to note before upgrading or downgrading the cluster

- You cannot add cluster nodes while upgrading or downgrading the cluster software version.
- You can perform node-level configurations through the NSIP address of individual nodes, but make sure to perform the same configurations on all the nodes to maintain them in synch.
- You cannot execute the "start nstrace" command from the cluster IP address when the cluster is being upgraded. However, you can get the trace of individual nodes by performing this operation on individual cluster nodes through their NetScaler IP (NSIP) address.
- Owing to changes in cluster licensing that were made in NetScaler 10.5 Build 52.11 (see license requirements), look into the following:
  - If the cluster is setup in a build prior to NetScaler 10.5 Build 52.11, the cluster will work with the separate cluster license file. No changes are required.
  - If the cluster is setup in NetScaler 10.5 Build 52.11 or later releases and then downgraded to a build prior to NetScaler 10.5 Build 52.11, the downgraded cluster will not work as it now expects a separate cluster license file.
- While upgrading from any NetScaler 10.1 build to a later release, syncookie must be disabled on all TCP profiles (using the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

"set ns tcpProfile <name> -synCookie DISABLED" command) and after that a striped SNIP must be added on the CLIP subnet. Once upgraded, syncookie can be enabled again.

- While upgrading the NetScaler appliance from a NetScaler 10.1 build to a NetScaler 10.5 build, do not execute the "show audit messages" command as this can cause the NetScaler appliance to become unresponsive.
- NetScaler 10.5 54.x and 55.x builds are not suitable for cluster deployment. This is because, for services that need probing, SYN packets are processed locally (on the flow receiver) even though syncookie is disabled.
- When a cluster is being upgraded, it is possible that the upgraded nodes have some additional features activated that are not available on the nodes that are not upgraded. This results in a license mismatch warning while the cluster is being upgraded. This warning will be automatically resolved when all the cluster nodes are upgraded.

> **Important**
>
> - Citrix recommends that you wait for the previous node to become active before upgrading or downgrading the next node.
>
> - Citrix recommends that the cluster configuration node must be upgraded/downgraded last to avoid multiple disconnect of cluster IP sessions.

To upgrade or downgrade the software of the cluster nodes

1. Make sure the cluster is stable and the configurations are synchronized on all the nodes.
2. Access each node through its NetScaler IP (NSIP) address and perform the following:
    1. Upgrade or downgrade the cluster node. For detailed information about upgrading and downgrading the software of an appliance, see Upgrading or Downgrading the System Software.
    2. Save the configurations.
    3. Reboot the appliance.
3. Repeat step 2 for each of the other cluster nodes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations Supported on Individual Cluster Nodes

Mar 20, 2015

As a rule, NetScaler appliances that are a part of a cluster cannot be individually configured from their NSIP address. However, there are some operations that are an exception to this rule. These operations, when executed from the NSIP address, are not propagated to other cluster nodes.

The operations are:

- cluster instance (set | rm | enable | disable)
- cluster node (set | rm)
- nstrace (start | show | stop)
- interface (set | enable | disable)
- route (add | rm | set | unset)
- arp (add | rm | send -all)
- force cluster sync
- sync cluster files
- disable ntp sync
- save ns config
- reboot
- shutdown

For example, when you execute the command disable interface 1/1/1 from the NSIP address of a cluster node, the interface is disabled only on that node. Since the command is not propagated, the interface 1/1/1 remains enabled on all the other cluster nodes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

FAQs

Dec 07, 2016

A list of the frequently asked questions about clustering (NetScaler 11.0, 10.5, 10.1). Click a question to view its answer.

- How many NetScaler appliances can be included in a single NetScaler cluster?
- Can a NetScaler appliance be a part of multiple clusters?
- What is a cluster IP address? What is its subnet mask?
- How can I make a specific cluster node as the cluster configuration coordinator?
- Why are the network interfaces of a cluster represented in 3-tuple (n/u/c) notation instead of the regular 2-tuple (u/c) notation?
- How can I set the hostname for a cluster node?
- Can I automatically detect NetScaler appliances so that I can add them to a cluster?
- Is the traffic serving capability of a cluster affected if a node is removed or disabled or reboot or shutdown or made inactive?
- I have multiple standalone appliances, each of which has different configurations. Can I add them to a single clust
- Can I migrate the configurations of a standalone NetScaler appliance or an HA setup to the clustered setup?
- Are backplane interfaces part of the L3 VLANs?
- How can I configure a cluster that includes nodes from different networks?
- How can I configure/un-configure the NSVLAN on a cluster?
- I have a cluster setup where some NetScaler nodes are not connected to the external network. Can the cluster still function normally?
- How can deployments that require MAC-Based Forwarding (MBF) be supported in a clustered setup?
- Can I execute commands from the NSIP address of a cluster node?
- Can I disable configuration propagation among cluster nodes?
- Can I change the NSIP address or change the NSVLAN of a NetScaler appliance when it is a part of the cluster?
- Does the NetScaler cluster support L2 and L3 VLANs?
- How can I configure SNMP on a NetScaler cluster?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- What details must I have available when I contact technical support for cluster-related issues?
- Can I use striped IP addresses as the default gateway of servers?
- Can I view routing configurations of a specific cluster node from the cluster IP address?
- How can I specify the node for which I want to set the LACP system priority?
- How are IP tunnels configured in a cluster setup?
- How can I add a failover interface set (FIS) on the nodes of a NetScaler cluster?
- How are net profiles configured in a cluster setup?
- How can I configure WIonNS in a cluster setup?
- Can the cluster LA channel be used for management access?
- How cluster nodes communicate to each other and what are the different types of traffic that goes through backplane?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting the NetScaler Cluster

Feb 09, 2015

If a failure occurs in a NetScaler cluster, the first step in troubleshooting is to get information on the cluster instance and the cluster nodes by running the show cluster instance <clId> and show cluster node <nodeId> commands respectively.

If you are not able to find the issue by using the above two approaches, you can use one of the following:

- **Isolate the source of the failure.** Try bypassing the cluster to reach the server. If the attempt is successful, the problem is probably with the cluster setup.
- **Check the commands recently executed.** Run the history command to check the recent configurations performed on the cluster. You can also review the ns.conf file to verify the configurations that have been implemented.
- **Check the ns.log files.** Use the log files, available in the /var/log/ directory of each node, to identify the commands executed, status of commands, and the state changes.
- **Check the newnslog files.** Use the newnslog files, available in the /var/nslog/ directory of each node, to identify the events that have occurred on the cluster nodes. You can view multiple newnslog files as a single file, by copying the files to a single directory, and then running the following command:
  nsconmsg -K newnslog-node<id> -K newnslog.node<id> -d current

If you still cannot resolve the issue, you can try tracing the packets on the cluster or use the show techsupport -scope cluster command to send the report to the technical support team.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tracing the Packets of a NetScaler Cluster

Feb 09, 2015

The NetScaler operating system provides a utility called *nstrace* to get a dump of the packets that are received and sent out by an appliance. The utility stores the packets in trace files. You can use these files to debug problems in the flow of packets to the cluster nodes. The trace files must be viewed with the Wireshark application.

Some salient aspects of the nstrace utility are:

- Can be configured to trace packets selectively by using classic expressions and default expressions.
- Can capture the trace in multiple formats: nstrace format (.cap) and TCP dump format (.pcap).
- Can aggregate the trace files of all cluster nodes on the configuration coordinator.
- Can merge multiple trace files into a single trace file (only for .cap files).

You can use the nstrace utility from the NetScaler command line or the NetScaler shell.

To trace packets of a standalone appliance

Run the start nstrace command on the appliance. The command creates trace files in the /var/nstrace/<date-timestamp> directory. The trace file names are of the form nstrace<id>.cap.

You can view the status by executing the show nstrace command. You can stop tracing the packets by executing the stop nstrace command.

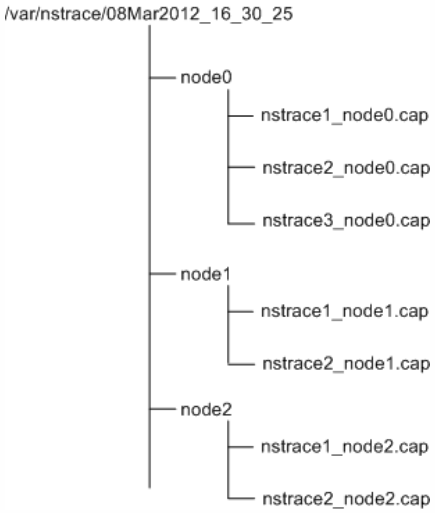
Note: You can also run the nstrace utility from the NetScaler shell by executing the nstrace.sh file. However, it is recommended that you use the nstrace utility through the NetScaler command line interface.

To trace packets of a cluster

You can trace the packets on all the cluster nodes and obtain all the trace files on the configuration coordinator.

Run the start nstrace command on the cluster IP address. The command is propagated and executed on all the cluster nodes. The trace files are stored in individual cluster nodes in the /var/nstrace/<date-timestamp> directory. The trace file names are of the form nstrace<id>_node<id>.cap.

You can use the trace files of each node to debug the nodes operations. But if you want the trace files of all cluster nodes in one location, you must run the stop nstrace command on the cluster IP address. The trace files of all the nodes are downloaded on the cluster configuration coordinator in the /var/nstrace/<date-timestamp> directory as follows:



Merge multiple trace files

You can prepare a single file from the trace files (supported only for .cap files) obtained from the cluster nodes. The single trace files gives you a cumulative view of the trace of the cluster packets. The trace entries in the single trace file are sorted based on the time the packets were received on the cluster.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To merge the trace files, at the NetScaler shell, type:

nstracemerge.sh -srcdir <DIR> -dstdir <DIR> -filename <name> -filesize <num>

where,

- srcdir is the directory from which the trace files are merged. All trace files within this directory are merged into a single file.
- dstdir is the directory where the merged trace file are created.
- filename is the name of the trace file that is created.
- filesize is the size of the trace file.

Examples

Following are some examples of using the nstrace utility to filter packets.

- To trace the packets on the backplane interfaces of three nodes:
  Using classic expressions:

  start nstrace -filter "INTF == 0/1/1 && INTF == 1/1/1 && INTF == 2/1/1"
  Using default expressions:

  start nstrace -filter "CONNECTION.INTF.EQ("0/1/1") && CONNECTION.INTF.EQ("1/1/1") && CONNECTION.INTF.EQ("2/1/1")"
- To trace the packets from a source IP address 10.102.34.201 or from a system whose source port is greater than 80 and the service name is not "s1":
  Using classic expressions

  start nstrace -filter "SOURCEIP == 10.102.34.201 || (SVCNAME != s1 && SOURCEPORT > 80)"
  Using default expressions

  start nstrace -filter "CONNECTION.SRCIP.EQ(10.102.34.201) || (CONNECTION.SVCNAME.NE("s1") && CONNECTION.SRCPORT.GT(80))"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting Common Issues

Dec 16, 2015

- While joining a node to the cluster, I get the following message, "ERROR: Invalid interface name/number." What must I do to resolve this error?
- While joining a node to the cluster, I get the following message, "ERROR: Clustering cannot be enabled, because the local node is not a member of the cluster." What must I do to resolve this error?
- While joining a node to the cluster, I get the following message, "ERROR: Connection refused." What must I do to resolve this error?
- While joining a node to the cluster, I get the following message, "ERROR: License mismatch between the configuration coordinator and the local node." What must I do to resolve this error?
- What must I do when the configurations of a cluster node are not in synch with the cluster configurations?
- When configuring a cluster node, I get the following message, "ERROR: Session is read-only; connect to the cluster IP address to modify the configuration."
- Why does the node state show "INACTIVE" when the node health shows "UP"?
- How can I resolve the health of a node when its health shows "NOT UP"?
- What must I do when the health of a node shows as "NOT UP" and the reason indicates that configuration commands have failed on a node?
- When I run the set vserver command, I get the following message, "No such resource." What must I do to resolve this issue?
- I cannot configure the cluster over a Telnet session. What must I do?
- I notice a significant time difference across the cluster nodes. What must I do to resolve this issue?
- What must I do, if there is no connectivity to the cluster IP address and the NSIP address of a cluster node?
- What must I do to recover a cluster node that has connectivity issues?
- Some nodes of the cluster have two default routes. How can I remove the second default route from the cluster node?
- The cluster functionality gets affected when an existing cluster node comes online. What must I do to resolve this issue?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

CloudBridge Connector

Dec 14, 2016

Note: The current NetScaler 1000V release does not support this feature.

The CloudBridge Connector feature of the Citrix NetScaler appliance connects enterprise datacenters to external clouds and hosting environments, making the cloud a secure extension of your enterprise network. Cloud-hosted applications appear as though they are running on one contiguous enterprise network. With Citrix CloudBridge Connector, you can augment your datacenters with the capacity and efficiency available from cloud providers.

The CloudBridge Connector enables you to move your applications to the cloud to reduce costs and increase reliability.

In addition to using CloudBridge Connector between a datacenter and a cloud, you can use it to connect two datacenters for a high-capacity secure and accelerated link.

Understanding CloudBridge Connector

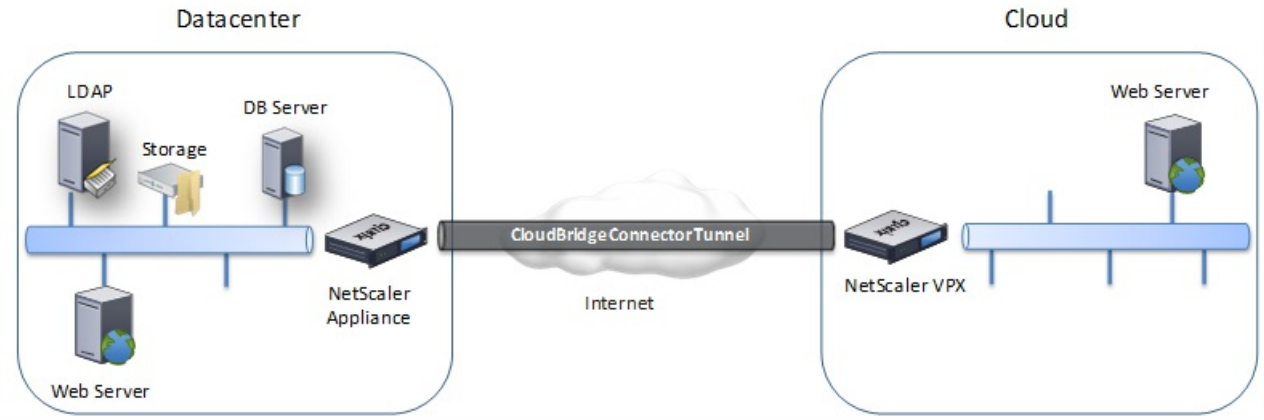
Updated: 2014-04-14

To implement the Citrix CloudBridge Connector solution, you connect a datacenter to another datacenter or an external cloud by setting up a tunnel called the CloudBridge Connector tunnel.

To connect a datacenter to another datacenter, you set up a CloudBridge Connector tunnel between two NetScaler appliances, one in each datacenter.

To connect a datacenter to an external cloud (for example, Amazon AWS cloud), you set up a CloudBridge Connector tunnel between a NetScaler appliance in the datacenter and a virtual appliance (VPX) that resides in the Cloud. The remote end point can be a CloudBridge Connector or a NetScaler VPX with platinum license.

The following illustration shows a CloudBridge Connector tunnel set up between a datacenter and an external cloud.



The appliances between which a CloudBridge Connector tunnel is set up are called the *end points* or *peers* of the CloudBridge Connector tunnel.

A CloudBridge Connector tunnel uses the following protocols:

- Generic Routing Encapsulation (GRE) protocol
- Open-standard IPSec Protocol suite, in transport mode

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The GRE protocol provides a mechanism for encapsulating packets, from a wide variety of network protocols, to be forwarded over another protocol. GRE is used to:

- Connect networks running non-IP and non-routable protocols.
- Bridge across a wide area network (WAN).
- Create a transport tunnel for any type of traffic that needs to be sent unchanged across a different network.

The GRE protocol encapsulates packets by adding a GRE header and a GRE IP header to the packets.

The Internet Protocol security (IPSec) protocol suite secures communication between peers in the CloudBridge Connector tunnel.

In a CloudBridge Connector tunnel, IPSec ensures:

- Data integrity
- Data origin authentication
- Data confidentiality (encryption)
- Protection against replay attacks

IPSec uses the transport mode in which the GRE encapsulated packet is encrypted. The encryption is done by the Encapsulating Security Payload (ESP) protocol. The ESP protocol ensures the integrity of the packet by using a HMAC hash function, and ensures confidentiality by using an encryption algorithm. After the packet is encrypted and the HMAC is calculated, an ESP header is generated. The ESP header is inserted after the GRE IP header and, an ESP trailer is inserted at the end of the encrypted payload.

Peers in the CloudBridge Connector tunnel use the Internet Key Exchange version (IKE) protocol (part of the IPSec protocol suite) to negotiate secure communication, as follows:

- The two peers mutually authenticate with each other, using one of the following authentication methods:
  - **Pre-shared key authentication**. A text string called a pre-shared key is manually configured on each peer. The pre-shared keys of the peers are matched against each other for authentication. Therefore, for the authentication to be successful, you must configure the same pre-shared key on each of the peers.
  - **Digital certificates authentication**. The initiator (sender) peer signs message interchange data by using its private key, and the other receiver peer uses the sender's public key to verify the signature. Typically, the public key is exchanged in messages containing an X.509v3 certificate. This certificate provides a level of assurance that a peer's identity as represented in the certificate is associated with a particular public key.
- The peers then negotiate to reach agreement on:
  - An encryption algorithm.
  - Cryptographic keys for encrypting data in one peer and decrypting the data in the other.

This agreement upon the security protocol, encryption algorithm and cryptographic keys is called a Security Association (SA). SAs are one-way (simplex). For example, when two peers, CB1 and CB2, are communicating through a Connector tunnel, CB1 has two Security Associations. One SA is used for processing out-bound packets, and the other SA is used for processing inbound packets.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

SAs expire after a specified length of time, which is called the *lifetime*. The two peers use the Internet Key Exchange (IKE) protocol (part of the IPSec protocol suite) to negotiate new cryptographic keys and establish new SAs. The purpose of the limited lifetime is to prevent attackers from cracking a key.

The following table lists some IPSec propeties supported by a NetScaler appliance:

IPSec Properties	Types Supported
IKE Versions	• V1 • V2
IKE Authentication Methods	• Pre-shared key authentication • Digital certificates authentication
Encryption Algorithm	• AES • 3DES
Hash Algorithm	• HMAC SHA1 • HMAC SHA256 • HMAC SHA384 • HMAC SHA512 • HMAC MD5

© 1999-2017 Citrix Systems, Inc. All rights reserved.

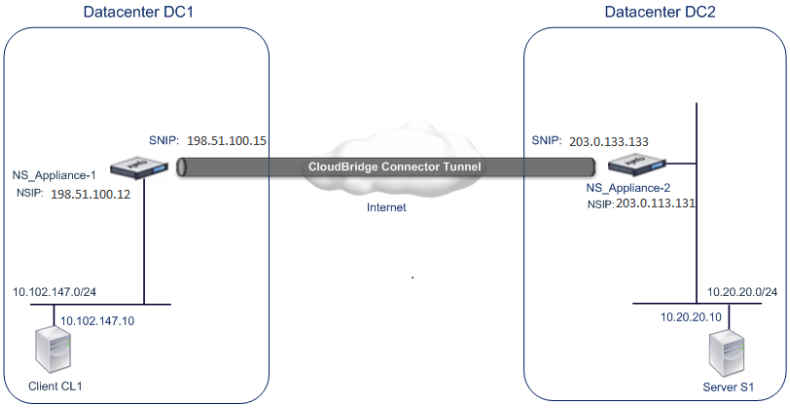
Configuring a CloudBridge Connector Tunnel between two Datacenters

Dec 01, 2015

You can configure a CloudBridge Connector tunnel between two different datacenters to extend your network without reconfiguring it, and leverage the capabilities of the two datacenters. A CloudBridge Connector tunnel between the two geographically separated datacenters enables you to implement redundancy and safeguard your setup from failure. The CloudBridge Connector tunnel helps achieve optimal utilization of infrastructure and resources across datacenters. The applications available across the two datacenters appear as local to the user.

To connect a datacenter to another datacenter, you set up a CloudBridge Connector tunnel between a NetScaler appliance in one datacenter and a NetScaler appliance in the other datacenter.

As an illustration of CloudBridge Connector tunnel between datacenters, consider an example in which a CloudBridge Connector tunnel is set up between NetScaler appliance NS_Appliance-1 in datacenter DC1 and NetScaler appliance NS_Appliance-2 in datacenter DC2.



Both NS_Appliance-1 and NS_Appliance-2 function in L2 and L3 mode. They enable communication between private networks in datacenters DC1 and DC2. In L3 mode, NS_Appliance-1 and NS_Appliance-2 enable communication between client CL1 in datacenter DC1 and server S1 in the datacenter DC2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

Because client CL1 and server S1 are on different private networks, L3 mode is enabled on NS_Appliance-1 and NS_Appliance-2, and routes are updated as follows:

- CL1 has a route to NS_Appliance-1 for reaching S1.
- NS_Appliance-1 has a route to NS_Appliance-2 for reaching S1.
- S1 has a route to NS_Appliance-2 for reaching CL1.
- NS_Appliance-2 has a route to NS_Appliance-1 for reaching CL1.

The following table lists the settings on NetScaler appliance NS_Appliance-1 in datacenter DC1.

Entity	Name	Details
The NSIP address		198.51.100.12
SNIP address		198.51.100.15
CloudBridge Connector tunnel	Cloud_Connector_DC1-DC2	• Local endpoint IP address of the CloudBridge Connector tunnel: 198.51.100.15 • Remote endpoint IP address of the CloudBridge Connector tunnel: 203.0.113.133 **GRE Tunnel Details** • Name = Cloud_Connector_DC1-DC2 **IPSec Profile Details** • Name = Cloud_Connector_DC1-DC2 • Encryption algorithm = AES • Hash algorithm = HMAC SHA1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following table lists the settings on NetScaler appliance NS_Appliance-2 in datacenter DC2.

Entity	Name	Details
The NSIP address		203.0.113.131
SNIP address		203.0.113.133
CloudBridge Connector tunnel	Cloud_Connector_DC1-DC2	• Local endpoint IP address of the CloudBridge Connector tunnel: 203.0.113.133 • Remote endpoint IP address of the CloudBridge Connector tunnel: 198.51.100.15 **GRE Tunnel Details** • Name = Cloud_Connector_DC1-DC2 **IPSec Profile Details** • Name = Cloud_Connector_DC1-DC2 • Encryption algorithm = AES • Hash algorithm = HMAC SHA1

Following is the traffic flow in the CloudBridge Connector tunnel:

1. Client CL1 sends a request to server S1.
2. The request reaches NetScaler appliance NS-Appliance-1.
3. NS_Appliance-1 checks its routing table and finds that the destination IP address of the request packets belong to a subnet in datacenter DC2.
4. NS_Appliance-1 uses the GRE protocol to encapsulate each of the request packets by adding a GRE header and a GRE IP header to the packet. The GRE IP header has the destination IP address set to the IP address of the CloudBridge tunnel (CC-DC1-DC2) end point in DC2. This IP Address is a public SNIP address configured on NS_Appliance-2.
5. For CloudBridge Connector tunnel CC-DC1-DC2, NS_Appliance-1 checks the stored IPSec security association (SA) parameters for processing outbound packets, as agreed between NS_Appliance-1 and NS_Appliance-2. The IPSec Encapsulating Security Payload (ESP) protocol in NS_Appliance-1 uses these SA parameters for outbound packets to encrypt the payload of the GRE encapsulated packet.
6. The ESP protocol ensures the packet's integrity and confidentiality by using the HMAC hash function and the encryption algorithm specified for CloudBridge Connector tunnel CC-DC1-DC2. The ESP protocol, after encrypting the GRE payload and calculating the HMAC, generates an ESP header and an ESP trailer and inserts them before and at the end of the encrypted GRE payload, respectively.
7. The resulting packet is sent to NS_Appliance-2.
8. NS_Appliance-2 checks the stored IPSec security association (SA) parameters for processing inbound packets, as agreed between NS_Appliance-1 and NS_Appliance-2 for the CloudBridge Connector tunnel CC-DC1-DC2. The IPSec ESP protocol on NS_Appliance-2 uses these SA parameters for inbound packets, and the ESP header of the request packet, to decrypt the packet.
9. NS_Appliance-2 then decapsulates the packet by removing the GRE header.
10. The resulting packet is the same packet as the one received by NS_Appliance-1 in step 2. This packet has the destination IP address set to the IP address of server S1. NS_Appliance-2 forwards this packet to server S1.
11. S1 processes the request packets and sends response packets. The destination IP address in the response packets is the IP address of client CL1, and the source IP address is the IP address of server S1.
12. The response packets reach NS_Appliance-2.
13. NS_Appliance-2 encapsulates and encrypts the response packets in the same way that NS_Appliance-1 encrypted the request packets in steps 3-6.
14. NS_Appliance-2 sends the resulting packets to NS_Appliance-1.
15. NS_Appliance-1, upon receiving the packets from NS_Appliance-2, decrypts and decapsulates them in the same way that NS_Appliance-2 did with the request packets in steps 9-11.

Prerequisites for Configuring a CloudBridge Connector tunnel between two Datacenters

Before setting up a CloudBridge Connector tunnel, verify that the following tasks have been completed:

1. Deploy and set up a NetScaler appliance in each of the two datacenters.
2. Make sure that the CloudBridge Connector tunnel end-point IP addresses are accessible to each other.

Configuration Procedure

To set up a CloudBridge Connector tunnel between a NetScaler appliance that resides in one datacenter and another NetScaler appliance that resides

© 1999-2017 Citrix Systems, Inc. All rights reserved.

in the other datacenter, use the configuration utility or the command line interface of one of the NetScaler appliances.

When you use the configuration utility, the CloudBridge Connector tunnel configuration created on the first NetScaler appliance is automatically pushed to the other endpoint (the other NetScaler appliance) of the CloudBridge Connector tunnel. Therefore, you do not have to access the configuration utility of the other NetScaler appliance to create the corresponding CloudBridge Connector tunnel configuration on it.

The CloudBridge Connector tunnel configuration on each of the NetScaler appliances consists of the following entities:

- **IPSec profile**—An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used by the IPSec protocol in the CloudBridge Connector tunnel.
- **GRE tunnel**—An IP tunnel specifies the local IP address (a public SNIP address configured on the local NetScaler appliance), remote IP address (a public SNIP address configured on the remote NetScaler appliance), protocol (GRE) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity.
- **Create a PBR rule and associate the IP tunnel with it**—A PBR entity specifies a set of conditions and an IP tunnel entity. The source IP address range and the destination IP range are the conditions for the PBR entity. You must set the source IP address range and the destination IP address range to specify the subnet whose traffic is to traverse the CloudBridge Connector tunnel. For example, consider a request packet that originates from a client on the subnet in the first datacenter and is destined to a server on the subnet in the second datacenter. If this packet matches the source and destination IP address range of the PBR entity on the NetScaler appliance in the first datacenter, it is sent across the CloudBridge Connector tunnel associated with the PBR entity.

To create an IPSEC profile by using the command line interface

At the command prompt, type:

add ipsec profile <name> [-**ikeVersion** (V1 | V2)] [-**encAlgo** (AES | 3DES) ...] [-**hashAlgo** <hashAlgo> ...] [-**lifetime** <positive_integer>] (-**psk** | (-**publickey** <string> -**privatekey** <string> -**peerPublicKey** <string>)) [-**livenessCheckInterval** <positive_integer>] [-**replayWindowSize** <positive_integer>] [-**ikeRetryInterval** <positive_integer>] [-**retransmissiontime** <positive_integer>]

Example

add ipsec profile Cloud_Connector_DC1-DC2
-encAlgo AES -hashAlgo HMAC_SHA1

To create an IP tunnel and bind the IPSEC profile to it by using the command line interface

At the command prompt, type:

add ipTunnel <name> <remote><remoteSubnetMask> <local> [-**protocol** <protocol>] [-**ipsecProfileName** <string>]

Example

add ipTunnel Cloud_Connector_DC1-DC2 203.0.113.133 255.255.255.255 198.51.100.15 -protocol GRE -ipsecProfileName Cloud_Connector_DC1-DC2

To create a PBR rule and bind the IPSEC tunnel to it by using the command line interface

At the command prompt, type:

- **add ns pbr** <pbr_name> ALLOW -**srcIP** = <local_subnet_range> -**destIP** = <remote_subnet_range> -**ipTunnel** <tunnel_name>
- **apply ns pbrs**

Example

- add ns pbr PBR-DC1-DC2 ALLOW -srcIP 198.51.100.15 -destIP 203.0.113.133 ipTunnel Cloud_Connector_DC1-DC2
- apply ns pbrs

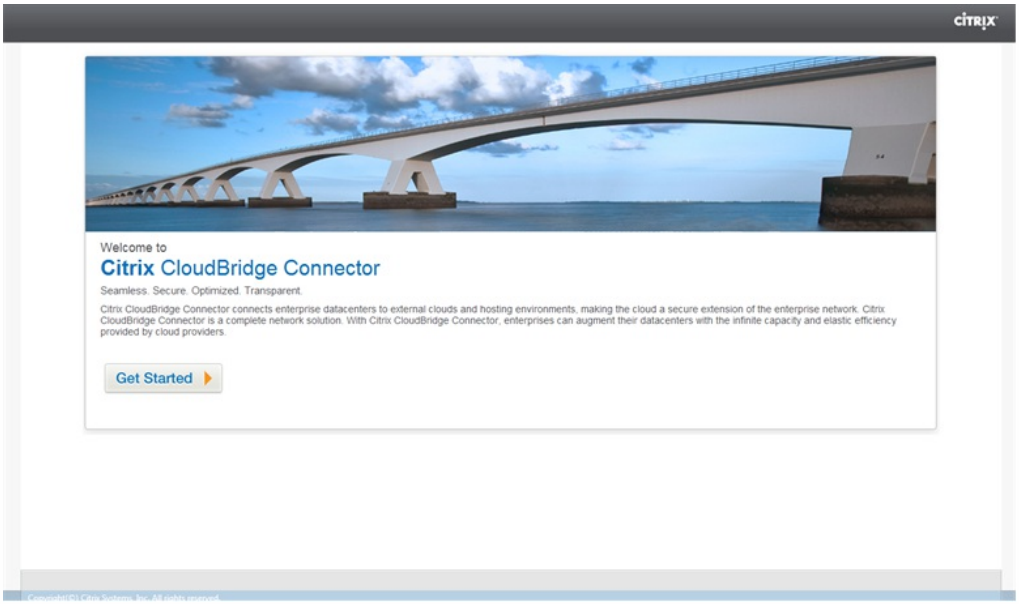
To configure a CloudBridge Connector tunnel in a NetScaler appliance by using the configuration utility

1. Type the NSIP address of a NetScaler appliance in the address line of a web browser.
2. Log on to the configuration utility of the NetScaler appliance by using your account credentials for the appliance.
3. Navigate to System > CloudBridge Connector.
4. In the right pane, under **Getting Started**, click **Create/Monitor CloudBridge**.

   The first time you configure a CloudBridge Connector tunnel on the appliance, a Welcome screen appears.

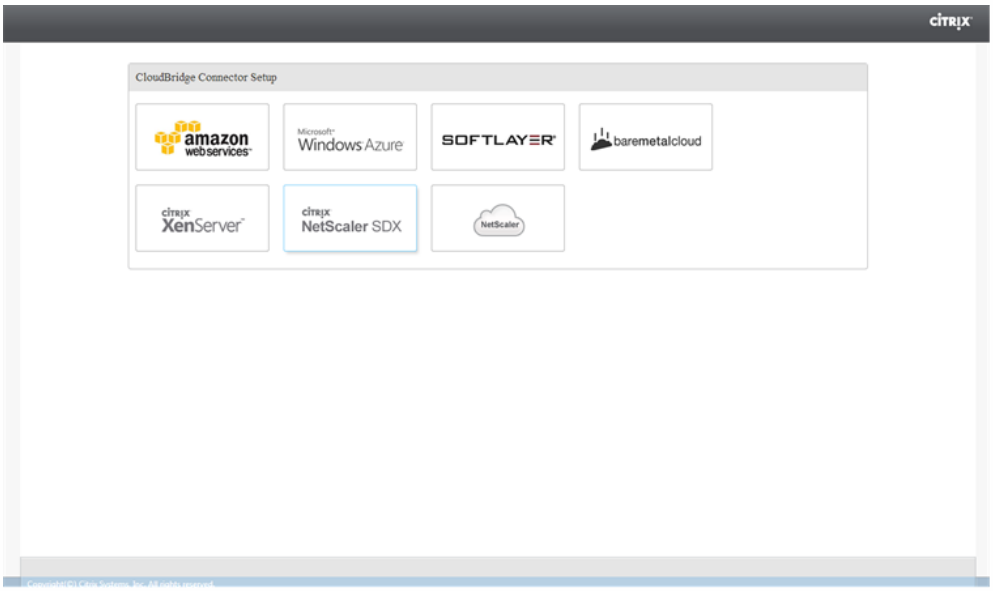
5. On the Welcome screen click **Get Started.**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.



Note: If you already have a CloudBridge Connector tunnel configured on the NetScaler appliance, the Welcome screen does not appear, so you do not click **Get Started**.

6. In the **CloudBridge Connector Setup** pane, click **NetScaler**.



7. In the **NetScaler** pane, provide your account credentials for the remote NetScaler appliance. Click **Continue**.
8. In the **CloudBridge Connector Setting** pane, set the following parameter:
   - **CloudBridge Connector Name**—Name for the CloudBridge Connector configuration on the local appliance. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the CloudBridge Connector configuration is created.
9. Under **Local Setting**, set the following parameter:
   - **Subnet IP**—IP address of the local endpoint of the CloudBridge Connector tunnel. Must be a public IP address of type SNIP.
10. Under **Remote Setting**, set the following parameter:
    - **Subnet IP**—IP address of the peer endpoint of the CloudBridge Connector tunnel. Must be a public IP address of type SNIP.
11. Under **PBR Setting**, set the following parameters:
    - **Operation**—Either the equals (=) or does not equal (!=) logical operator.
    - **Source IP Low***—Lowest source IP address to match against the source IP address of an outgoing IPv4 packet.
    - **Source IP High**—Highest source IP address to match against the source IP address of an outgoing IPv4 packet.
    - **Operation**—Either the equals (=) or does not equal (!=) logical operator.
    - **Destination IP Low***—Lowest destination IP address to match against the destination IP address of an outgoing IPv4 packet.
    - **Destination IP High**—Highest destination IP address to match against the destination IP address of an outgoing IPv4 packet.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

12. (Optional) Under **Security Settings**, set the following IPSec protocol parameters for the CloudBridge Connector tunnel:
    - **Encryption Algorithm**—Encryption algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
    - **Hash Algorithm**—Hash algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
    - **Key**—Select one of the following IPSec authentication methods to be used by the two peers to mutually authenticate.
      - **Auto Generate Key**—Authentication based on a text string, called a pre-shared key (PSK), generated automatically by the local appliance. The PSKs keys of the peers are matched against each other for authentication.
      - **Specific Key**—Authentication based on a manually entered PSK. The PSKs of the peers are matched against each other for authentication.
        - **Pre Shared Security Key**—The text string entered for pre-shared key based authentication.
      - **Upload Certificates**—Authentication based on digital certificates.
        - **Public Key**—A local digital certificate to be used to authenticate the local NetScaler appliance to the peer before establishing IPSec security associations. The same certificate should be present and set for the Peer Public Key parameter in the peer.
        - **Private Key**—Private key of the local digital certificate.
        - **Peer Public Key**—Digital certificate of the peer. Used to authenticate the peer to the local end point before establishing IPSec security associations. The same certificate should be present and set for the Public key parameter in the peer.
13. Click **Done**.

The new CloudBridge Connector tunnel configuration on both the NetScaler appliances appears on the Home tab of the respective configuration utility. The current status of the CloudBridge connector tunnel is indicated in the Configured CloudBridge Connectors pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

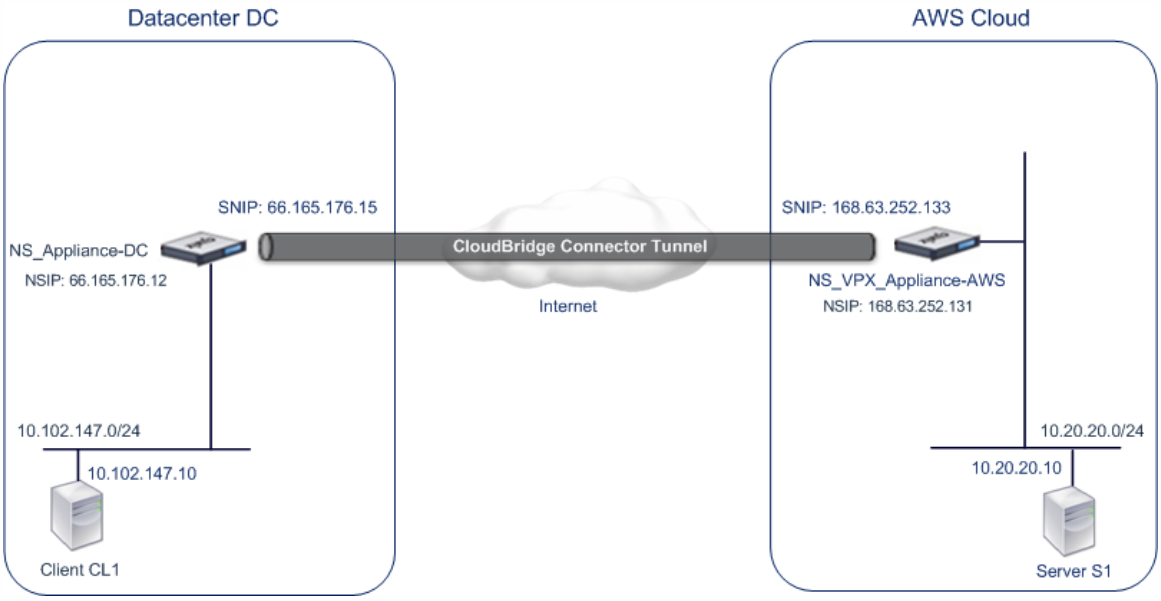
Configuring CloudBridge Connector between Datacenter and AWS Cloud

Jan 28, 2011

You can configure a CloudBridge Connector tunnel between a datacenter and AWS cloud to leverage the infrastructure and computing capabilities of the data center and the AWS cloud. With AWS, you can extend your network without initial capital investment or the cost of maintaining the extended network infrastructure. You can scale your infrastructure up or down, as required. For example, you can lease more server capabilities when the demand increases.

To connect a datacenter to AWS cloud, you set up a CloudBridge Connector tunnel between a NetScaler appliance that resides in the datacenter and a NetScaler virtual appliance (VPX) that resides in AWS cloud.

As an illustration of a CloudBridge Connector tunnel between a datacenter and Amazon AWS cloud, consider an example in which a CloudBridge Connector tunnel is set up between NetScaler appliance NS_Appliance-DC, in datacenter DC, and NetScaler virtual appliance (VPX) NS_VPX_Appliance-AWS.



Both NS_Appliance-DC and NS_VPX_Appliance-AWS function in L3 mode. They enable communication between private networks in datacenter DC and the AWS cloud. NS_Appliance-DC and NS_VPX_Appliance-AWS enable communication between client CL1 in datacenter DC and server S1 in the AWS cloud through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

Note: AWS does not support L2 mode, hence it is necessary to have only L3 mode enabled on both the endpoints.

For proper communication between CL1 and S1, L3 mode is enabled on NS_Appliance-DC and NS_VPX_Appliance-AWS and routes are updated as such:

- CL1 have a route to NS_Appliance-DC for reaching S1.
- NS_Appliance-DC have a route to NS_VPX_Appliance-AWS for reaching S1.
- S1 should have a route to NS_VPX_Appliance-AWS for reaching CL1.
- NS_VPX_Appliance-AWS have a route to NS_Appliance-DC for reaching CL1.

The following table lists the settings on NetScaler appliance NS_Appliance-DC in datacenter DC.

Entity	Name	Details
The NSIP address		66.165.176.12

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SNIP address		66.165.176.15
CloudBridge Connector tunnel	CC_Tunnel_DC-AWS	• Local endpoint IP address of the CloudBridge Connector tunnel: 66.165.176.15 • Remote endpoint IP address of the CloudBridge Connector tunnel: 168.63.252.133 **GRE Tunnel Details** • Name= CC_Tunnel_DC-AWS **IPSec Profile Details** • Name= CC_Tunnel_DC-AWS • Encryption algorithm= AES • Hash algorithm= HMAC SHA1

The following table lists the settings on NetScaler VPX NS_VPX_Appliance-AWS on AWS cloud.

Entity	Name	Details
NSIP address		10.102.25.30
Public EIP address mapped to the NSIP address		168.63.252.131
SNIP address		10.102.29.30
Public EIP address mapped to the SNIP address		168.63.252.133
CloudBridge Connector tunnel	CC_Tunnel_DC-AWS	• Local endpoint IP address of the CloudBridge Connector tunnel: 168.63.252.133 • Remote endpoint IP address of the CloudBridge Connector tunnel: 66.165.176.15 **GRE Tunnel Details** • Name= CC_Tunnel_DC-AWS **IPSec Profile Details** • Name= CC_Tunnel_DC-AWS • Encryption algorithm= AES • Hash algorithm= HMAC SHA1

Prerequisites

Updated: 2015-06-01

Before setting up a CloudBridge Connector tunnel, verify that the following tasks have been completed:

1. Install, configure, and launch an instance of NetScaler Virtual appliance (VPX) on AWS cloud. For instructions on installing NetScaler VPX on AWS, see http://support.citrix.com/proddocs/topic/netscaler-vpx-10-5/nsvpx-aws-ns-vpxaws-con.html.

2. Deploy and configure a NetScaler physical appliance, or provisioning and configuring a NetScaler virtual appliance (VPX) on a virtualization platform in the datacenter.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- For instructions on installing NetScaler vitrual appliances on Xenserver, see http://support.citrix.com/proddocs/topic/netscaler-vpx-10-5/ns-vpx-install-wrapper-con.html.
- For instructions on installing NetScaler vitrual appliances on VMware ESX or ESXi, see http://support.citrix.com/proddocs/topic/netscaler-vpx-10-5/ns-vpx-install-on-esx-wrapper-con.html.
- For instructions on installing NetScaler vitrual appliances on Microsoft Hyper-V, see http://support.citrix.com/proddocs/topic/netscaler-vpx-10-5/ns-vpx-install-on-msft-hyperv-wrapper-con.html.

3. Make sure that the CloudBridge Connector tunnel end-point IP addresses are accessible to each other.

NetScaler VPX License

After the initial instance launch, NetScaler VPX for AWS requires a license. If you are bringing your own license (BYOL), see the VPX Licensing Guide at: http://support.citrix.com/article/CTX122426.

You have to:
1. Use the licensing portal within MyCitrix to generate a valid license.
2. Upload the license to the instance.

If this is a **paid** marketplace instance, then you do not need to install a license. The correct feature set and performance will activate automatically.

Configuration Steps

To set up a CloudBridge Connector tunnel between a NetScaler appliance that resides in a datacenter and a NetScaler virtual appliance (VPX) that resides on the AWS cloud, use the configuration utility of the NetScaler appliance.

When you use the configuration utility, the CloudBridge Connector tunnel configuration created on the NetScaler appliance, is automatically pushed to the other endpoint or peer (the NetScaler VPX on AWS) of the CloudBridge Connector tunnel. Therefore, you do not have to access the configuration utility (GUI) of the NetScaler VPX on AWS to create the corresponding CloudBridge Connector tunnel configuration on it.

The CloudBridge Connector tunnel configuration on both peers (the NetScaler appliance that resides in the datacenter and the NetScaler virtual appliance (VPX) that resides on the AWS cloud) consists of the following entities:
- **IPSec profile**—An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used by the IPSec protocol in both the peers of the CloudBridge Connector tunnel.
- **GRE tunnel**—An IP tunnel specifies a local IP address (a public SNIP address configured on the local peer), remote IP address (a public SNIP address configured on the remote peer), protocol (GRE) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity.
- **Create a PBR rule and associate the IP tunnel with it**—A PBR entity specifies a set of conditions and an IP tunnel entity. The source IP address range and the destination IP range are the conditions for the PBR entity. You must set the source IP address range and the destination IP address range to specify the subnet whose traffic is to traverse the CloudBridge Connector tunnel. For example, consider a request packet that originates from a client on the subnet in the datacenter and is destined to a server on the subnet in the AWS cloud. If this packet matches the source and destination IP address range of the PBR entity on the NetScaler appliance in the datacenter, it is sent across the CloudBridge Connector tunnel associated with the PBR entity.

To create an IPSEC profile by using the command line interface

At the command prompt, type:

add ipsec profile <name> [-**ikeVersion** (V1 | V2)] [-**encAlgo** (AES | 3DES) ...] [-**hashAlgo** <hashAlgo> ...] [-**lifetime** <positive_integer>] (-**psk** | (-**publickey** <string> -**privatekey** <string> -**peerPublicKey** <string>)) [-**livenessCheckInterval** <positive_integer>] [-**replayWindowSize** <positive_integer>] [-**ikeRetryInterval** <positive_integer>] [-**retransmissiontime** <positive_integer>]

Example

add ipsec profile CC_Tunnel_DC-AWS -encAlgo AES -hashAlgo HMAC_SHA1

To create an IP tunnel and bind the IPSEC profile to it by using the command line interface

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

add ipTunnel <name> <remote><remoteSubnetMask> <local> [-**protocol** <protocol>] [-**ipsecProfileName** <string>]

Example

add ipTunnel CC_Tunnel_DC-AWS 168.63.252.133 255.255.255.0 66.165.176.15 –protocol GRE -ipsecProfileName CC_Tunnel_DC-AWS

To create a PBR rule and bind the IPSEC tunnel to it by using the command line interface

At the command prompt, type:

- **add ns pbr** <pbr_name> ALLOW -**srcIP** = <local_subnet_range> -**destIP** = <remote_subnet_range> -**ipTunnel** <tunnel_name>
- **apply ns pbrs**

Example

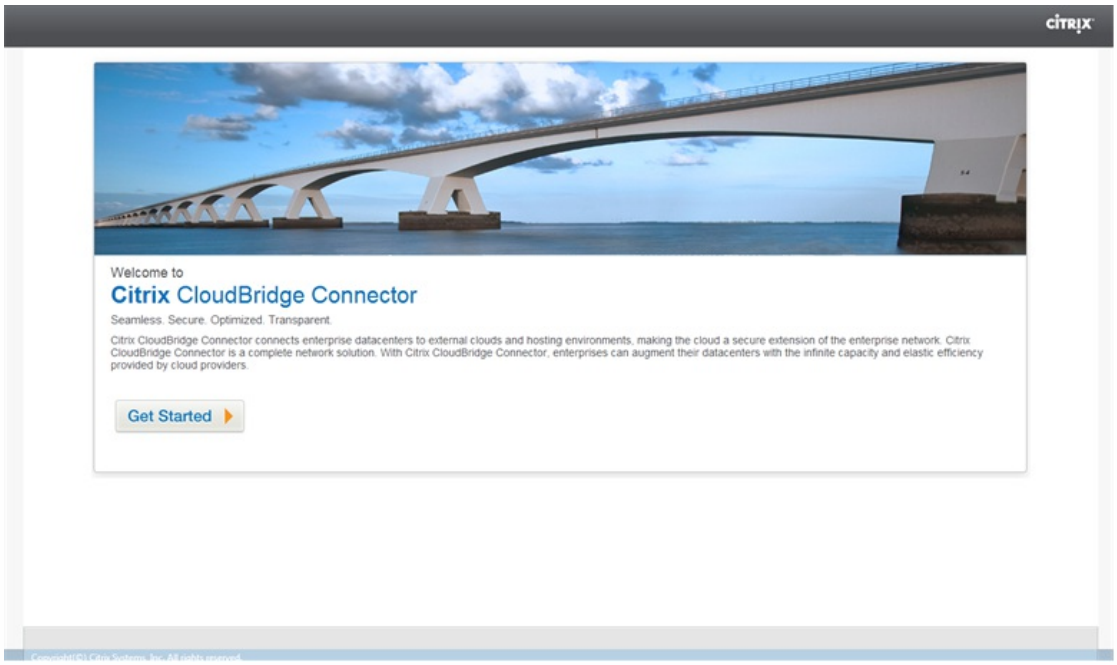
- add ns pbr PBR-DC-AWS ALLOW –srcIP 66.165.176.15 –destIP 168.63.252.133 ipTunnel CC_Tunnel_DC-AWS
- apply ns pbrs

To configure a CloudBridge Connector tunnel in a NetScaler appliance by using the configuration utility

1. Type the NSIP address of a NetScaler appliance in the address line of a web browser.
2. Log on to the configuration utility of the NetScaler appliance by using your account credentials for the appliance.
3. Navigate to System > CloudBridge Connector.
4. In the right pane, under **Getting Started**, click **Create/Monitor CloudBridge**.

   The first time you configure a CloudBridge Connector tunnel on the appliance, a Welcome screen appears.

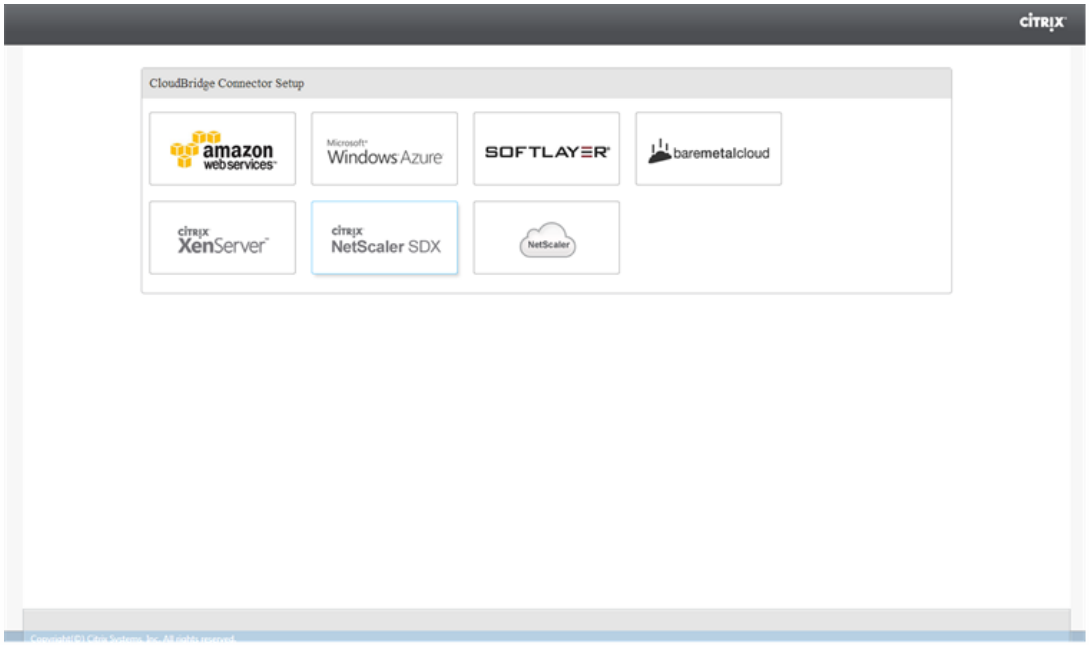
5. On the Welcome screen click **Get Started**.



   Note: If you already have a CloudBridge Connector tunnel configured on the NetScaler appliance, the Welcome screen does not appear, so you do not click **Get Started**.

6. In the **CloudBridge Connector Setup** pane, click **amazon web services**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



7. In the **Amazon** pane, provide your AWS account credentials: AWS Access Key ID and AWS Secret Access Key. You can obtain these access keys from the AWS GUI console. Click **Continue**.
8. In the **NetScaler** pane, select the NSIP address of the NetScaler virtual appliance running on AWS. Then, provide your account credentials for the NetScaler virtual appliance. Click **Continue**.
9. In the **CloudBridge Connector Setting** pane, set the following parameter:
   - **CloudBridge Connector Name**— Name for the CloudBridge Connector configuration on the local appliance. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the CloudBridge Connector configuration is created.
10. Under **Local Setting**, set the following parameter:
    - **Subnet IP**— IP address of the local endpoint of the CloudBridge Connector tunnel. Must be a public IP address of type SNIP.
11. Under **Remote Setting**, set the following parameter:
    - **Subnet IP**— IP address of the CloudBridge Connector tunnel end point on the AWS side. Must be an IP address of type SNIP on the NetScaler VPX instance on AWS.
    - **NAT**— Public IP address (EIP) in AWS that is mapped to the SNIP configured on the NetScaler VPX instance on AWS.
12. Under **PBR Setting**, set the following parameters:
    - **Operation**— Either the equals (=) or does not equal (!=) logical operator.
    - **Source IP Low***— Lowest source IP address to match against the source IP address of an outgoing IPv4 packet.
    - **Source IP High**— Highest source IP address to match against the source IP address of an outgoing IPv4 packet.
    - **Operation**— Either the equals (=) or does not equal (!=) logical operator.
    - **Destination IP Low***— Lowest destination IP address to match against the destination IP address of an outgoing IPv4 packet.
    - **Destination IP High**— Highest destination IP address to match against the destination IP address of an outgoing IPv4 packet.
13. (Optional) Under **Security Settings**, set the following IPSec protocol parameters for the CloudBridge Connector tunnel:
    - **Encryption Algorithm**— Encryption algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
    - **Hash Algorithm—**Hash algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
    - **Key**— Select one of the following IPSec authentication methods to be used by the two peers to mutually authenticate.
      - **Auto Generate Key**— Authentication based on a text string, called a pre-shared key (PSK), generated automatically by the local appliance. The PSKs keys of the peers are matched against each other for authentication.
      - **Specific Key—**Authentication based on a manually entered PSK. The PSKs of the peers are matched against each other for authentication.
        - **Pre Shared Security Key**— The text string entered for pre-shared key based authentication.
      - **Upload Certificates**— Authentication based on digital certificates.
        - **Public Key**— A local digital certificate to be used to authenticate the local peer to the remote peer before establishing IPSec security associations. The same certificate should be present and set for the Peer Public Key parameter in the peer.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Private Key**— Private key of the local digital certificate.
- **Peer Public Key**— Digital certificate of the peer. Used to authenticate the peer to the local end point before establishing IPSec security associations. The same certificate should be present and set for the Public key parameter in the peer.

14. Click **Done**.

The new CloudBridge Connector tunnel configuration on the NetScaler appliance in the datacenter appears on the Home tab of the configuration utility.

The corresponding new CloudBridge Connector tunnel configuration on the NetScaler VPX appliance in the AWS cloud appears on the configuration utility.

The current status of the CloudBridge connector tunnel is indicated in the Configured CloudBridge pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a CloudBridge Connector Tunnel Between a NetScaler Appliance and Virtual Private Gateway on AWS

Feb 03, 2016

To connect a datacenter to Amazon Web Services (AWS), you can configure a CloudBridge Connector tunnel between a NetScaler appliance in the datacenter and a virtual private gateway on AWS. The NetScaler appliance and the virtual private gateway form the endpoints of the CloudBridge Connector tunnel and are called peers.

Note: You can also set up a CloudBridge Connector tunnel between a NetScaler appliance in a datacenter and a NetScaler VPX instance (instead of a virtual private gateway) on AWS. For more information, see Configuring CloudBridge Connector between Datacenter and AWS Cloud.

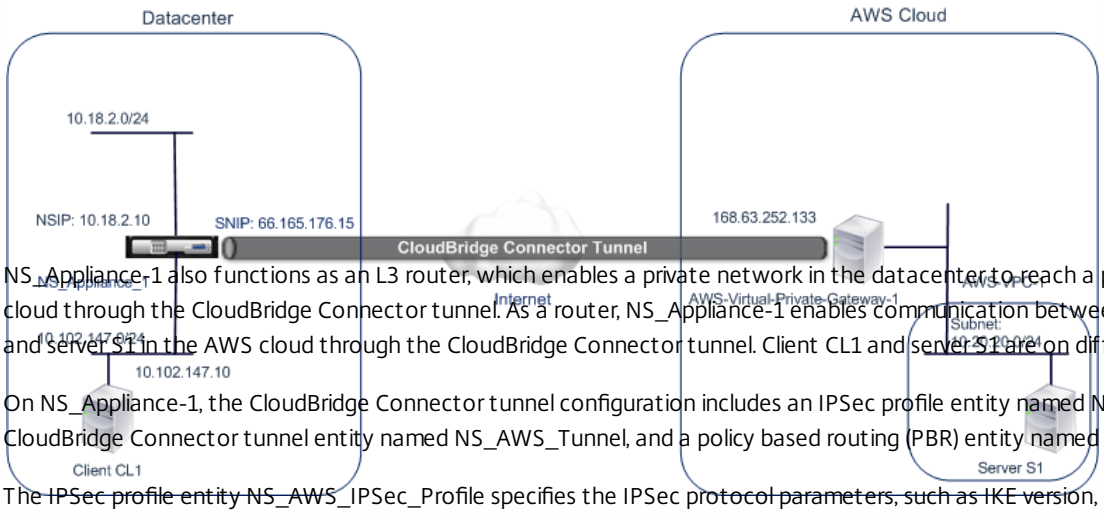
Virtual private gateways on AWS support the following IPSec settings for a CloudBridge Connector tunnel. Therefore, you must specify the same IPSec settings when you configure the NetScaler appliance for the CloudBridge Connector tunnel.

IPSec Properties	Setting
IPSec mode	Tunnel mode
IKE version	Version 1
IKE Authentication method	Pre-Shared Key
Encryption algorithm	AES
Hash algorithm	HMAC SHA1

Example of CloudBridge Connector Tunnel Configuration and Data Flow

As an illustration of the traffic flow in a CloudBridge Connector tunnel, consider an example in which a CloudBridge Connector tunnel is set up between NetScaler appliance NS_Appliance-1 in a datacenter and virtual private gateway gateway AWS-Virtual-Private-Gateway-1 on AWS cloud.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



NS_Appliance-1 also functions as an L3 router, which enables a private network in the datacenter to reach a private network in the AWS cloud through the CloudBridge Connector tunnel. As a router, NS_Appliance-1 enables communication between client CL1 in the datacenter and server S1 in the AWS cloud through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On NS_Appliance-1, the CloudBridge Connector tunnel configuration includes an IPSec profile entity named NS_AWS_IPSec_Profile, a CloudBridge Connector tunnel entity named NS_AWS_Tunnel, and a policy based routing (PBR) entity named NS_AWS_Pbr.

The IPSec profile entity NS_AWS_IPSec_Profile specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, and hash algorithm, to be used by the IPSec protocol in the CloudBridge Connector tunnel. NS_AWS_IPSec_Profile is bound to IP tunnel entity NS_AWS_Tunnel.

CloudBridge Connector tunnel entity NS_AWS_Tunnel specifies the local IP address (a public IP—SNIP—address configured on the NetScaler appliance), the remote IP address (the IP address of the AWS-Virtual-Private-Gateway-1), and the protocol (IPSec) used to set up the CloudBridge Connector tunnel. NS_AWS_Tunnel is bound to policy based routing (PBR) entity NS_AWS_Pbr.

The PBR entity NS_AWS_Pbr specifies a set of conditions and a CloudBridge Connector tunnel entity (NS_AWS_Tunnel). The source IP address range and the destination IP address range are the conditions for NS_AWS_Pbr. The source IP address range and the destination IP address range are specified as a subnet in the datacenter and a subnet in the AWS cloud, respectively. Any request packet originating from a client in the subnet in the datacenter and destined to a server in the subnet on the AWS cloud matches the conditions in NS_AWS_Pbr. This packet is then considered for CloudBridge Connector processing and is sent across the CloudBridge Connector tunnel (NS_AWS_Tunnel) bound to the PBR entity.

The following table lists the settings used in this example.

Entity	Name	Details
Settings highlight of the CloudBridge Connector tunnel setup		
IP address of the CloudBridge Connector tunnel end point (NS_Appliance-1) in the datacenter side		66.165.176.15
IP address of the CloudBridge Connector tunnel end point (AWS-Virtual-Private-Gateway-1) in the AWS		168.63.252.133
Datacenter Subnet, the traffic of which is to traverse the CloudBridge Connector tunnel		10.102.147.0/24
AWS Subnet, the traffic of which is to traverse the CloudBridge Connector tunnel		10.20.20.0/24
Settings on NetScaler appliance NS_Appliance-1 in Datacenter		
	SNIP1(for reference purposes only)	66.165.176.15

© 1999-2017 Citrix Systems, Inc. All rights reserved.

IPSec profile	NS_AWS_IPSec_Profile	• IKE version = v1 • Encryption algorithm = AES • Hash algorithm = HMAC SHA1
CloudBridge Connector tunnel	NS_AWS_Tunnel	• Remote IP = 168.63.252.133 • Local IP= 66.165.176.15 • Tunnel protocol = IPSec • IPSec profile= NS_AWS_IPSec_Profile
Policy based route	NS_AWS_Pbr	• Source IP range = Subnet in the datacenter =10.102.147.0-10.102.147.255 • Destination IP range =Subnet in AWS =10.20.20.0-10.20.20.255 • IP Tunnel = NS_AWS_Tunnel
Settings on Amazon AWS		
Customer Gateway	AWS-Customer-Gateway-1	• Routing = Static • IP Address = Internet-routable CloudBridge Connector tunnel endpoint IP address on the NetScaler side = 66.165.176.15
Virtual Private Gateway	AWS-Virtual-Private-Gateway-1	• Associated VPC = AWS-VPC-1

VPN Connection	AWS-VPN-Connection-1	• Customer Gateway = AWS-Customer-Gateway-1
		• Virtual Private Gateway= Virtual-Private-Gateway-1
		• Routing Options
		  • Type = Static
		  • Static IP Prefixes = Subnets on the NetScaler side = 10.102.147.0/24

Following is the traffic flow in the CloudBridge Connector tunnel:

1. Client C1 sends a request to server S1.
2. The request packet is received by NS_Appliance-1. The request packet matches the condition in NS_AWS_Pbr, because the source IP address and the destination IP address of the request packet are in the source IP range and destination IP range specified in NS_AWS_Pbr.
3. Because CloudBridge Connector tunnel entity NS_AWS_Tunnel is bound to NS_AWS_Pbr, the appliance prepares the packet to be sent across the NS_AWS_Tunnel.
4. IPSec's Encapsulating Security Payload (ESP) protocol encrypts the request packet and then attaches an ESP header and an ESP trailer to the encrypted IP packet. The IPSec protocol encapsulates the resulting packet by adding an IP header before the ESP header. The destination address in the IP header is the IP address of AWS-Virtual-Private-Gateway-1, and the source address is the SNIP1 address.
5. The resulting packet is sent to AWS-Virtual-Private-Gateway-1.
6. AWS-Virtual-Private-Gateway-1, upon receiving the packet from NS_Appliance-1, decapsulates the packet by removing the IPSec IP header, decrypts the packet, and then removes the ESP header and ESP trailer. The resulting packet is the same packet as the one received by NS_Appliance-1 in step 2. This packet has the destination IP address set to the IP address of server S1.
7. AWS-Virtual-Private-Gateway-1 forwards this packet to server S1.
8. S1 processes the request packet and sends a response packet. The destination IP address in the response packet is the IP address of

© 1999-2017 Citrix Systems, Inc. All rights reserved.

client CL1, and the source IP address is the IP address of server S1.
9. The response packet reaches AWS-Virtual-Private-Gateway-1. The IPSec protocol in AWS encrypts and encapsulates the response packet.
10. AWS-Virtual-Private-Gateway-1 sends the resulting packet to NS_Appliance-1.
11. NS_Appliance-1, upon receiving the packet from AWS-Virtual-Private-Gateway-1, decapsulates and decrypts the packet. The resulting packet is the same packet that was received by AWS-Virtual-Private-Gateway-1 in step 9. This response packet has the destination IP address set to the IP address of server CL1.
12. NS_Appliance-1 forwards the response packet to client CL1.

Points to Consider for a CloudBridge Connector Tunnel Configuration

Before configuring a CloudBridge Connector tunnel between a NetScaler appliance and AWS gateway, consider the following points:

1. AWS supports the following IPSec settings for a CloudBridge Connector tunnel. Therefore, you must specify the same IPSec settings when you configure the NetScaler appliance for the CloudBridge Connector tunnel.

- IKE version = v1
- Encryption algorithm = AES
- Hash algorithm = HMAC SHA1

2. You must configure the firewall at the NetScaler end to allow the following.

- Any UDP packets for port 500
- Any UDP packets for port 4500
- Any ESP (IP protocol number 50) packets

3. You must configure Amazon AWS before specifying the tunnel configuration on the NetScaler, because the public IP address of the AWS end (gateway) of the tunnel and the PSK are automatically generated when you set up the tunnel configuration in AWS. You need this information for specifying the tunnel configuration on the NetScaler appliance.

4. AWS gateway supports static routes and the BGP protocol for route updates. The NetScaler appliance does not support the BGP protocol in a CloudBridge Connector tunnel to AWS gateway. Therefore, appropriate static routes must be used on both sides of the CloudBridge Connector tunnel for proper routing of traffic through the tunnel. Bottom of Form

Configuring Amazon AWS for the CloudBridge Connector Tunnel

To create a CloudBridge Connector tunnel configuration on Amazon AWS, use the Amazon AWS Management Console, which is a web based graphical interface for creating and managing resources on Amazon AWS.

Before you begin the CloudBridge Connector tunnel configuration on AWS cloud, make sure that:

- You have a user account for Amazon AWS cloud.
- You have a virtual private cloud whose networks you want to connect to the networks at the NetScaler side through the CloudBridge Connector tunnel.
- You are familiar with the Amazon AWS Management Console.

Note: The procedures for configuring Amazon AWS for a CloudBridge Connector tunnel might change over time, depending on the Amazon AWS release cycle. Citrix recommends the following Amazon AWS documentation for the latest procedures.

- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_VPN.html

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure a CloudBridge connector tunnel between a NetScaler and AWS gateway perform the following tasks on the AWS Management Console:

- **Create a Customer Gateway**. A customer gateway is an AWS entity that represents a CloudBridge Connector tunnel endpoint. For a CloudBridge Connector tunnel between a NetScaler appliance and AWS gateway, the customer gateway represents the NetScaler appliance on AWS. The customer gateway specifies a name, the type of routing (static or BGP) used in the tunnel, and the CloudBridge Connector tunnel endpoint IP address on the NetScaler side. The IP address can be an Internet-routable NetScaler owned subnet IP (SNIP) address or, if the NetScaler appliance is behind a NAT device, an Internet-routable NAT IP address that represents the SNIP address.
- **Create a Virtual Private Gateway and attach it to a VPC**. A virtual private gateway is a CloudBridge Connector tunnel endpoint at the AWS side. When you create a virtual private gateway, you assigned it a name or allow AWS to assign the name. You then associate the virtual private gateway with a VPC. This association enables the subnets of the VPC to connect to the subnets at the NetScaler side through the CloudBridge Connector tunnel.
- **Create a VPN Connection**. A VPN connection specifies a customer gateway and a virtual private gateway between which a CloudBridge Connector tunnel is to be created. It also specifies an IP prefix for the networks at the NetScaler side. Only IP prefixes that are known to the virtual private gateway (through static route entry) can receive traffic from the VPC through the tunnel. Also, the virtual private gateway does not route any traffic not destined to the specified IP prefixes through the tunnel. After configuring a VPN connection, you might have to wait few minutes for it to be created.
- **Configure Routing Options**. For the VPC's network to reach the networks at the NetScaler side through the CloudBridge Connector tunnel, you must configure the VPC's routing table to include routes for the networks at the NetScaler side and point those routes to the virtual private gateway. You can include routes in a VPC's routing table in one of the following ways:
  - **Enable Route Propagation**. You can enable route propagation for your routing table, so that routes are automatically propagated to the table. The static IP prefixes that you specify for VPN configuration are propagated to the routing table after you've created the VPN connection.
  - **Enter Static Routes Manually**. If you do not enable route propagation, you must manually enter the static routes for the networks at the NetScaler side.

- **Download Configuration**. After the CloudBridge Connector tunnel (VPN connection) configuration is created on AWS, download the configuration file of the VPN connection to your local system. You might need the information in the configuration file for configuring the CloudBridge Connector tunnel on the NetScaler appliance.

To create a customer gateway

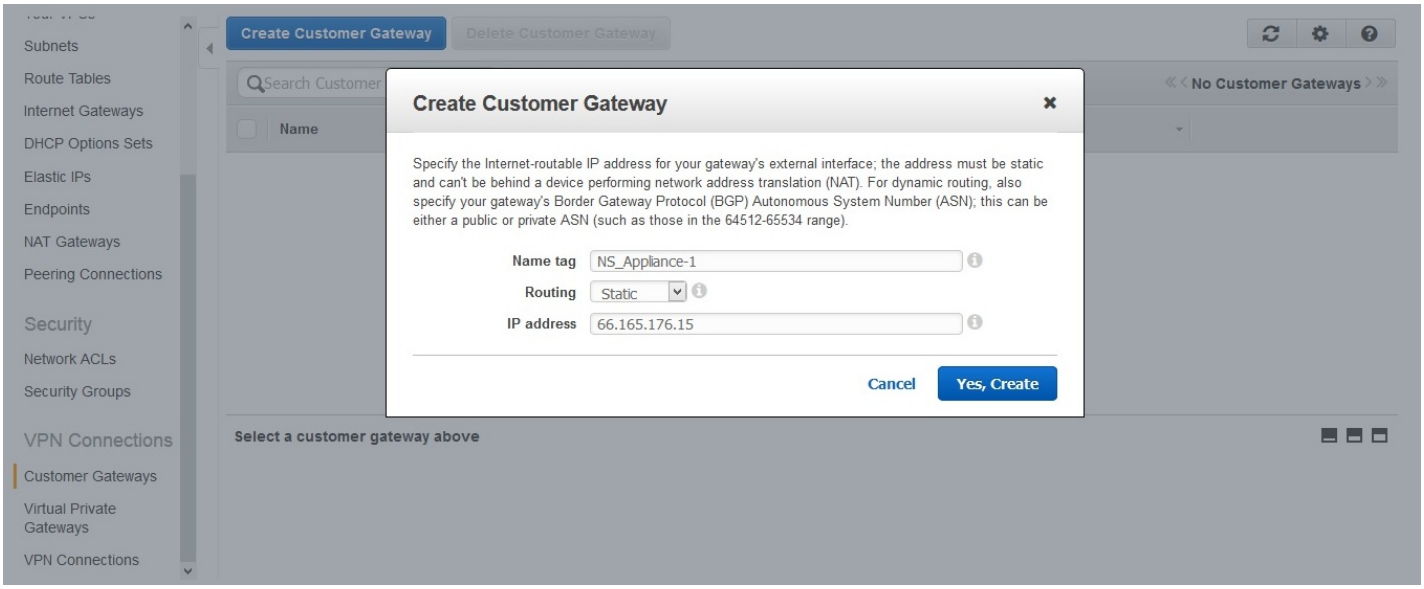
1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Navigate to **VPN Connections** > **Customer Gateways** and click on **Create Customer Gateway**.
3. In the **Create Customer Gateway** dialog box, set the following parameters and then click **Yes, Create**:

- **Name tag**. A name for the customer gateway.
- **Routing list**. Type of routing between NetScaler appliance and AWS virtual private gateway for advertising routes to each other through the CloudBridge Connector tunnel. Select **Static Routing** from the **Routing** list.

  Note: The NetScaler appliance does not support the BGP protocol in a CloudBridge Connector tunnel to AWS gateway. Therefore, appropriate static routes must be used on both sides of the CloudBridge Connector tunnel for proper routing of traffic through the tunnel.

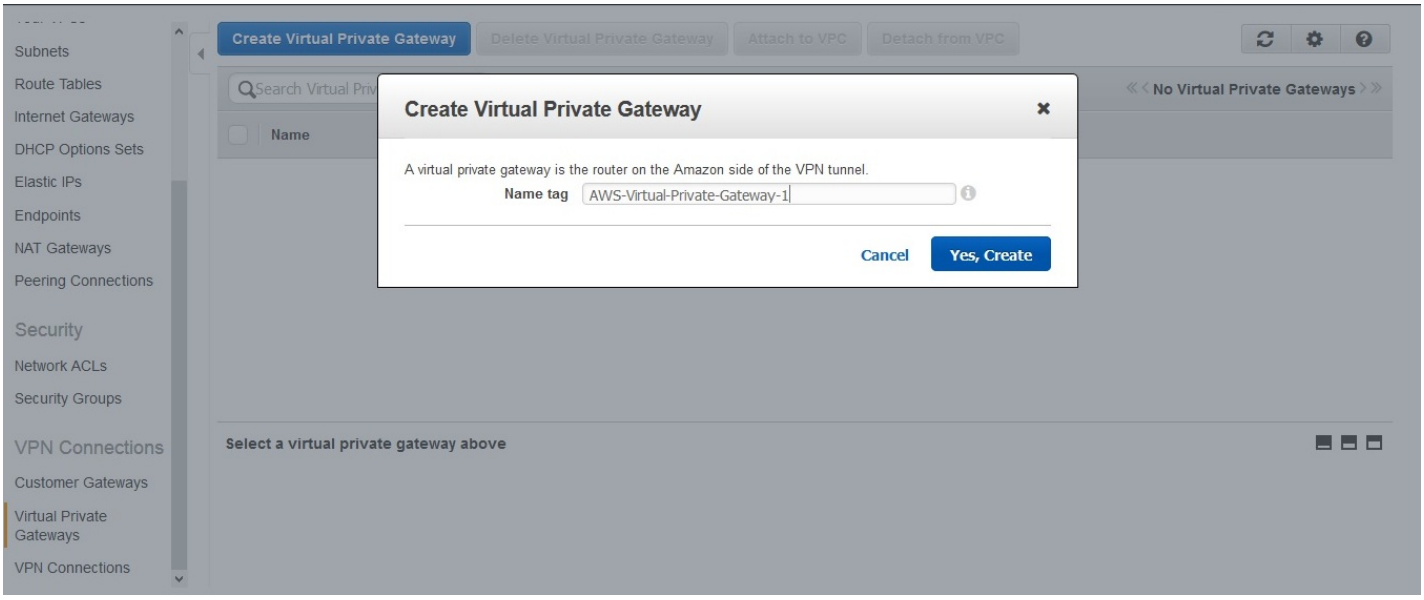
- **IP Address**. Internet-routable CloudBridge Connector tunnel endpoint IP address on the NetScaler side. The IP address can be an Internet-routable NetScaler owned subnet IP (SNIP) address or, if the NetScaler appliance is behind a NAT device, an Internet-routable NAT IP address that represents the SNIP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



To create a virtual private gateway and attach it to a VPC

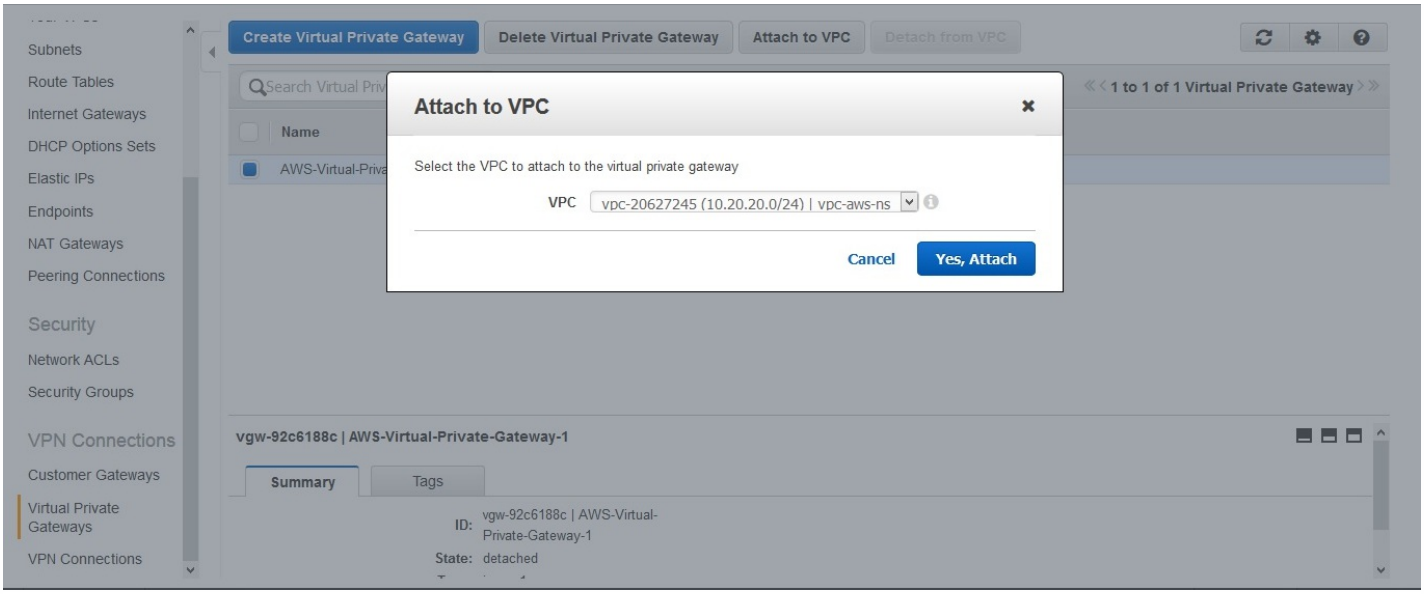
1. Navigate to VPN Connections > Virtual Private Gateways, and then click Create Virtual Private Gateway.
2. Enter a name for the virtual private gateway, and then click Yes, Create.



3. Select the virtual private gateway that you created, and then click Attach to VPC.

4. In the Attach to VPC dialog box, select your VPC from the list, and then choose Yes, Attach.

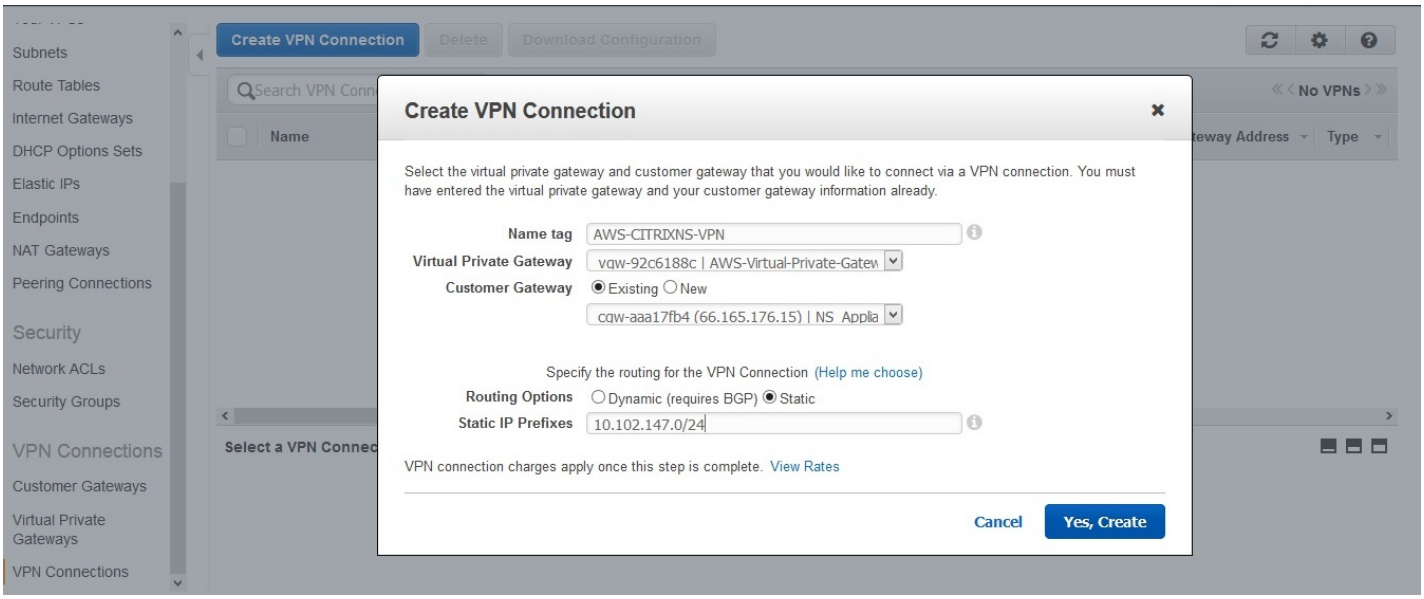
© 1999-2017 Citrix Systems, Inc. All rights reserved.



To create a VPN connection

1. Navigate to VPN Connections > VPN Connections and then click Create VPN Connection.
2. In the Create VPN Connection dialog box set the following parameters and then choose Yes, Create:

- Name tag. A name for the VPN connection.
- Virtual Private Gateway. Select the virtual private gateway that you created earlier.
- Customer Gateway. Select Existing. Then, from the drop down list, select the customer gateway that you created earlier.
- Routing Options. Type of routing between the virtual private gateway and customer gateway (NetScaler appliance). Select Static. In the Static IP Prefixes field, specify the IP prefixes for the subnet on the NetScaler side, separated by commas.



To enable route propagation

1. Navigate to Route Tables and select the routing table that's associated with the subnet whose traffic is to traverse the CloudBridge Connector tunnel.
   Note: By default, this is the main routing table for the VPC.
2. On the Route Propagation tab in the details pane, choose Edit, select the virtual private gateway, and then choose Save.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To manually enter static routes

1. Navigate to Route Tables and select your routing table.

2. On the Routes tab, click Edit.

3. In the Destination field, enter the static route used by your CloudBridge Connector tunnel (VPN connection).

4. Select the virtual private gateway ID from the Target list, and then choose Save.

To download the configuration file

1. Navigate to VPN Connection, select a VPN connection, and then click Download Configuration.
2. In the Download Configuration dialog box, set the following parameters, and then click Yes, Download.

- Vendor. Select Generic.
- Platform. Select Generic.
- Software. Select Vendor Agnostic.

Configuring the NetScaler Appliance for the CloudBridge Connector Tunnel

To configure a CloudBridge Connector tunnel between a NetScaler appliance and a virtual private gateway on AWS cloud, perform the following tasks on the NetScaler appliance. You can use either the NetScaler command line or the configuration utility:

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm and PSK to be used by the IPSec protocol in the CloudBridge Connector tunnel.
- **Create an IP tunnel that uses IPSec protocol and associate the IPSec profile with it**. An IP tunnel specifies the local IP address (a SNIP address configured on the NetScaler appliance), remote IP address (the public IP address of the virtual private gateway in AWS), protocol (IPSec) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.
- **Create a PBR rule and associate it with the IP tunnel**. A PBR entity specifies a set of rules and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP address range are the conditions for the PBR entity. Set the source IP address range to specify the NetScaler-side subnet whose traffic is to traverse the tunnel, and set the destination IP address range to specify the AWS VPC subnet whose traffic is to traverse the CloudBridge Connector tunnel. Any request packet that originates from a client in the subnet on the NetScaler side and is destined to a server in the AWS cloud subnet, and matches the source and destination IP range of the PBR entity, is sent across the CloudBridge Connector tunnel associated with the PBR entity.

To create an IPSEC profile by using the NetScaler command line

At the Command prompt, type:

- **add ipsec profile** <name> -**psk** <string> -**ikeVersion** v1

To create an IPSEC tunnel and bind the IPSEC profile to it by using the NetScaler command line

At the Command prompt, type:

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> -**protocol** IPSEC –**ipsecProfileName** <string>

To create a PBR rule and bind the IPSEC tunnel to it by using the NetScaler command line

At the Command prompt, type:

- **add pbr** <pbrName> ALLOW –**srcIP** <subnet-range> -**destIP** <subnet-range> -**ipTunnel** <tunnelName>
- **apply pbrs**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Sample configuration COPY

```
The following commands create all settings of NetScaler appliance NS_Appliance-1 used in "Example of CloudBridge Connector Configuration and Data Flow

> add ipsec profile NS_AWS_IPSec_Profile -psk  DkiMgMdcbqvYREEuIvxsbKkW0Foyabcd -ikeVersion v1 –lifetime 31536000

Done

> add iptunnel NS_AWS_Tunnel 168.63.252.133 255.255.255.255 66.165.176.15 –protocol IPSEC –ipsecProfileName NS_AWS_IPSec_Profile

Done

> add pbr NS_AWS_Pbr -srcIP 10.102.147.0-10.102.147.255 –destIP 10.20.0.0-10.20.255.255 –ipTunnel NS_AWS_Tunnel

Done


> apply pbrs

Done
```

To create an IPSEC profile by using the configuration utility

1. Navigate to **System** > **CloudBridge Connector** > **IPSec Profile**.
2. In the details pane, click **Add**.
3. In the **Add IPSec Profile** dialog box, set the following parameters:

- Name
- Encryption Algorithm
- Hash Algorithm
- IKE Protocol Version (select V1)

4. Select the **Pre-shared Key Authentication** method and set the **Pre-Shared Key Exists** parameter.
5. Click **Create**, and then click **Close**.

To create an IP tunnel and bind the IPSEC profile to it by using the configuration utility

1. Navigate to **System** > **CloudBridge Connector** > **IP Tunnels**.
2. On the **IPv4 Tunnels** tab, click Add.
3. In the **Add IP Tunnel** dialog box, set the following parameters:

- Name
- Remote IP
- Remote Mask
- Local IP Type (In the Local IP Type drop down list, select Subnet IP).
- Local IP (All the configured IPs of the selected IP type are in the Local IP drop down list. Select the desired IP from the list.)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Protocol
- IPSec Profile

4. Click **Create**, and then click **Close**.

To create a PBR rule and bind the IPSEC tunnel to it by using the configuration utility

1. Navigate to **System** > **Network** > **PBR**.

2. On the **PBR** tab, click **Add**.

3. In the **Create PBR** dialog box, set the following parameters:

- Name
- Action
- Next Hop Type (Select IP Tunnel)
- IP Tunnel Name
- Source IP Low
- Source IP High
- Destination IP Low
- Destination IP High

4. Click **Create**, and then click **Close**.

The corresponding new CloudBridge Connector tunnel configuration on the NetScaler appliance appears in the configuration utility.

The current status of the CloudBridge connector tunnel is shown in the Configured CloudBridge Connector pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

Monitoring the CloudBridge Connector Tunnel

You can view statistics for monitoring the performance of a CloudBridge Connector tunnel between the NetScaler appliance and the AWS gateway. To view CloudBridge Connector tunnel statistics on the NetScaler appliance, use the NetScaler command line.

The following table lists the statistical counters available for monitoring CloudBridge Connector tunnels on a NetScaler appliance.

Statistical counter	Specifies
Bytes Received	Total number of bytes received by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Bytes Sent	Total number of bytes sent by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Packets Received	Total number of packets received by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Packets Sent	Total number of packets sent by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.

All these counters are reset to 0 when the NetScaler appliance is restarted. They do not increment during the following phases:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Internet Key Exchange (IKE) authentication (pre-shared key) phase on any configured CloudBridge Connector tunnel.
- IKE Security Association (SA) establishment phase on any configured CloudBridge Connector tunnel.

To display CloudBridge Connector tunnel statistics by using the command line

At the command prompt, type:

- **stat ipsec counters**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a CloudBridge Connector Tunnel Between a Datacenter and Azure Cloud

Aug 23, 2016

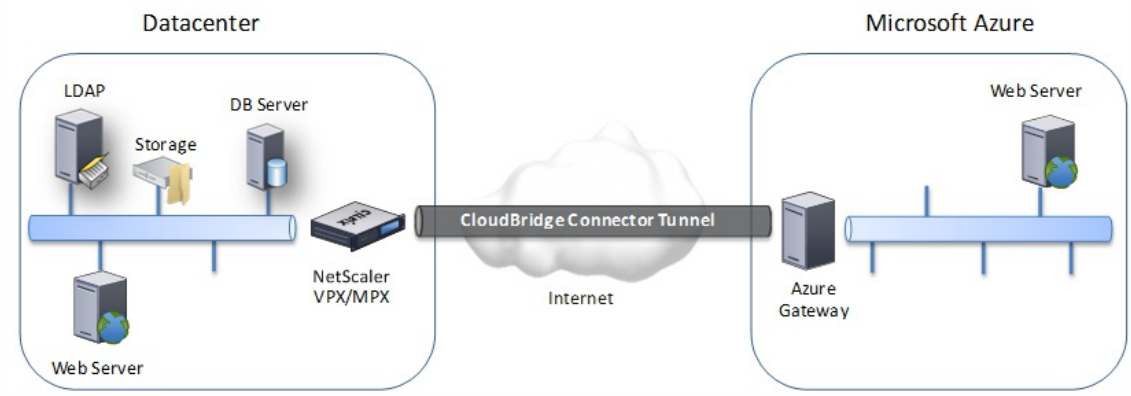
The NetScaler appliance provides connectivity between your enterprise datacenters and the Microsoft cloud hosting provider, Azure, making Azure a seamless extension of the enterprise network. NetScaler encrypts the connection between the enterprise datacenter and Azure cloud so that all data transferred between the two is secure.

This section includes the following:

- How CloudBridge Connector Tunnel Works
- Example of CloudBridge Connector Tunnel Configuration and Data Flow
- Points to Consider for a CloudBridge Connector tunnel Configuration
- Configuring the CloudBridge Connector Tunnel
- Monitoring the CloudBridge Connector Tunnel

How CloudBridge Connector Tunnel Works

To connect a datacenter to Azure cloud, you set up a CloudBridge Connector tunnel between a NetScaler appliance that resides in the datacenter and a gateway that resides in the Azure cloud. The NetScaler appliance in the datacenter and the gateway in Azure cloud are the end points of the CloudBridge Connector tunnel and are called peers of the CloudBridge Connector tunnel.



A CloudBridge Connector tunnel between a datacenter and Azure cloud uses the open-standard Internet Protocol security (IPSec) protocol suite, in tunnel mode, to secure communications between peers in the CloudBridge Connector tunnel. In a CloudBridge Connector tunnel, IPSec ensures:

- Data integrity
- Data origin authentication
- Data confidentiality (encryption)
- Protection against replay attacks

IPSec uses the tunnel mode in which the complete IP packet is encrypted and then encapsulated. The encryption uses the Encapsulating Security Payload (ESP) protocol, which ensures the integrity of the packet by using a HMAC hash function and ensures confidentiality by using an encryption algorithm. The ESP protocol, after encrypting the payload and calculating the HMAC, generates an ESP header and inserts it before the encrypted IP packet. The ESP protocol also generates an ESP trailer and inserts it at the end of the packet.

The IPSec protocol then encapsulates the resulting packet by adding an IP header before the ESP header. In the IP header, the destination IP address is set to the IP address of the CloudBridge Connecter peer.

Peers in the CloudBridge Connector tunnel use the Internet Key Exchange version 1 (IKEv1) protocol (part of the IPSec protocol suite) to negotiate secure communication, as follows:

1. The two peers mutually authenticate with each other, using pre-shared key authentication, in which the peers exchange a text string called a pre-shared key (PSK). The pre-shared keys are matched against each other for authentication. Therefore, for the authentication to be successful, you must configure the same pre-shared key on each of the peers.
2. The peers then negotiate to reach agreement on:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- An encryption algorithm
- Cryptographic keys for encrypting data on one peer and decrypting it on the other.

This agreement upon the security protocol, encryption algorithm and cryptographic keys is called a Security Association (SA). SAs are one-way (simplex). For example, when a CloudBridge Connector tunnel is set up between a NetScaler appliance in a datacenter and a gateway in an Azure cloud, both the datacenter appliance and the Azure gateway have two SAs. One SA is used for processing out-bound packets, and the other SA is used for processing inbound packets. SAs expire after a specified interval of time, which is called the lifetime.

Example of CloudBridge Connector Tunnel Configuration and Data Flow

As an illustration of CloudBridge Connector Tunnel, consider an example in which a CloudBridge Connector tunnel is set up between NetScaler appliance CB_Appliance-1 in a datacenter and gateway Azure_Gateway-1 in Azure cloud.

CB_Appliance-1 also functions as an L3 router, which enables a private network in the datacenter to reach a private network in the Azure cloud through the CloudBridge Connector tunnel. As a router, CB_Appliance-1 enables communication between client CL1 in the datacenter and server S1 in the Azure cloud through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On CB_Appliance-1, the CloudBridge Connector tunnel configuration includes an IPSec profile entity named CB_Azure_IPSec_Profile, a CloudBridge Connector tunnel entity named CB_Azure_Tunnel, and a policy based routing (PBR) entity named CB_Azure_Pbr.

The IPSec profile entity CB_Azure_IPSec_Profile specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, and hash algorithm, to be used by the IPSec protocol in the CloudBridge Connector tunnel. CB_Azure_IPSec_Profile is bound to IP tunnel entity CB_Azure_Tunnel.

CloudBridge Connector tunnel entity CB_Azure_Tunnel specifies the local IP address (a public IP (SNIP) address configured on the NetScaler appliance), the remote IP address (the IP address of the Azure_Gateway-1), and the protocol (IPSec) used to set up the CloudBridge Connector tunnel. CB_Azure_Tunnel is bound to the PBR entity CB_Azure_Pbr.

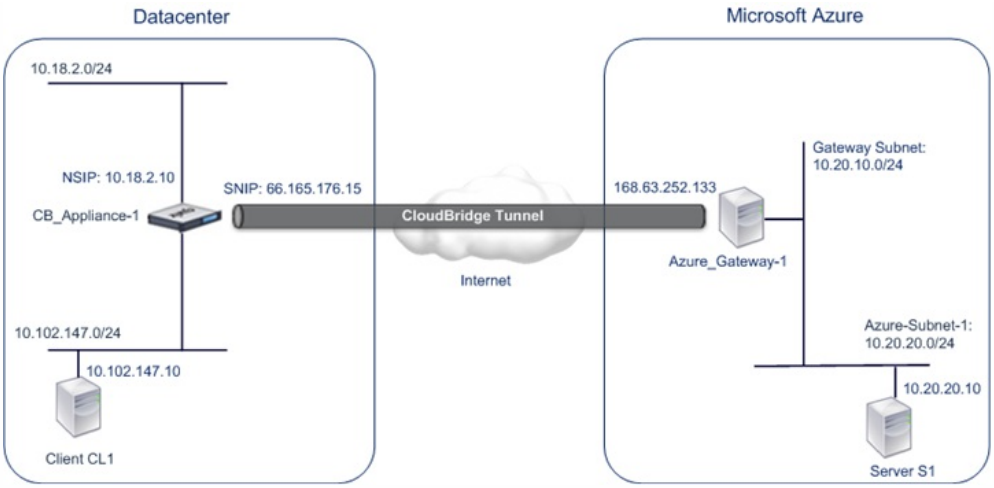
The PBR entity CB_Azure_Pbr specifies a set of conditions and a CloudBridge Connector tunnel entity (CB_Azure_Tunnel). The source IP address range and the destination IP address range are the conditions for CB_Azure_Pbr. The source IP address range and the destination IP address range are specified as a subnet in the datacenter and a subnet in the Azure cloud, respectively. Any request packet originating from a client in the subnet in the datacenter and destined to a server in the subnet on the Azure cloud matches the conditions in CB_Azure_Pbr. This packet is then considered for CloudBridge processing and is sent across the CloudBridge Connector tunnel (CB_Azure_Tunnel) bound to the PBR entity.

On Microsoft Azure, the CloudBridge Connector tunnel configuration includes a local network entity named My-Datacenter-Network, a virtual network entity named Azure-Network-for-CloudBridge-Tunnel, and a gateway named Azure_Gateway-1.

The local (local to Azure) network entity My-Datacenter-Network specifies the IP address of the NetScaler appliance on the datacenter side, and the datacenter subnet whose traffic is to traverse the CloudBridge Connector tunnel. The virtual network entity Azure-Network-for-CloudBridge-Tunnel defines a private subnet named Azure-Subnet-1 in Azure. The traffic of the subnet traverses the CloudBridge Connector tunnel. The server S1 is provisioned in this subnet.

The local network entity My-Datacenter-Network is associated with the virtual network entity Azure-Network-for-CloudBridge-Tunnel. This association defines the remote and local network details of the CloudBridge Connector tunnel configuration in Azure. Gateway Azure_Gateway-1 was created for this association to become the CloudBridge end point at the Azure end of the CloudBridge Connector tunnel.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The following table lists the settings used in this example.

Entity	Name	Details
Settings highlight of the CloudBridge Connector tunnel setup		
IP address of the CloudBridge Connector tunnel end point (CB_Appliance-1) in the datacenter side	66.165.176.15	
IP address of the CloudBridge Connector tunnel end point (Azure_Gateway-1) in the Azure	168.63.252.133	
Datacenter Subnet, the traffic of which is to traverse the CloudBridge Connector tunnel	10.102.147.0/24	
Azure Subnet, the traffic of which is to traverse the CloudBridge Connector tunnel	10.20.0.0/16	
Settings on NetScaler appliance CB_Appliance-1 in Datacenter		
	SNIP1(for reference purposes only)	66.165.176.15
IPSec profile	CB_Azure_IPSec_Profile	• IKE version = v1 • Encryption algorithm = AES • Hash algorithm = HMAC SHA1
CloudBridge Connector tunnel	CB_Azure_Tunnel	• Remote IP = 168.63.252.133 • Local IP= 66.165.176.15 • Tunnel protocol = IPSec • IPSec profile= CB_Azure_IPSec_Profile
Policy based route	CB_Azure_Pbr	• Source IP range = Subnet in the datacenter =10.102.147.0-10.102.147.255 • Destination IP range =Subnet in Azure =10.20.0.0-10.20.255.255 • IP Tunnel = CB_Azure_Tunnel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity	Name	Details
Settings on Microsoft Azure		
Public IP Address of the Azure_Gateway-1		168.63.252.133
Local Network	My-Datacenter-Network	• VPN Device IP address =SNIP address of the NetScaler appliance = 66.165.176.15 • Address space= Subnet in datacenter =10.102.147.0/24
Virtual Network	Azure-Network-for-CloudBridge-Tunnel	• Address Space= 10.20.0.0/16 • Subnet in Azure=Azure-Subnet-1= 10.20.20.0/24 • Local Network=My-Datacenter-Network • Gateway Subnet=10.20.10.0/24

Following is the traffic flow in the CloudBridge Connector tunnel:

1. Client C1 sends a request to server S1.
2. The request reaches NetScaler appliance CB_Appliance-1.
3. The request packet in CB_Appliance-1 matches the condition specified in the PBR entity CB_Azure_Pbr as the source IP address and the destination IP address of the request packet belonging to the source IP range and destination IP range, respectively, set in CB_Azure_Pbr.
4. Because CloudBridge Connector tunnel entity CB_Azure_Tunnel is bound to CB_Azure_Pbr, the appliance prepares the packet to be sent across the CB_Azure_Tunnel.
5. For CloudBridge Connector tunnel CB_Azure_Tunnel, CB_Appliance-1 checks the stored IPSec security association (SA) parameters for processing outbound packets, as agreed between CB_Appliance-1 in the datacenter and Azure_Gateway-1 in the Azure cloud. The IPSec Encapsulating Security Payload (ESP) protocol in the NetScaler appliance uses these SA parameters for outbound packets to encrypt the request packet.
6. The ESP protocol ensures the packet's integrity by using a HMAC hash function and the packet's confidentiality by using the AES encryption algorithm. The ESP protocol, after encrypting the request packet and calculating the HMAC, generates an ESP header and then inserts it before the encrypted IP packet. The ESP protocol also generates an ESP trailer and then inserts it at the end of the encrypted IP packet.
7. The IPSec protocol encapsulates the resulting packet by adding an IP header before the ESP header. The destination address in the IP header is the IP address of Azure-gateway-1, and the source address is the SNIP2 address.
8. The resulting packet is sent to Azure_Gateway-1. There is
9. Azure-gateway-1, upon receiving the packet from CB_Appliance-1, decapsulates the packet by removing the IPSec IP header.
10. Azure-gateway-1 then checks the stored IPSec security association (SA) parameters for processing inbound packets, as agreed between CB_Appliance-1 and Azure_Gateway-1. The IPSec ESP protocol on Azure_Gateway-1 uses these SA parameters for inbound packets, and the ESP header of the decapsulated request packet, to decrypt the packet.
11. The resulting packet is the same packet as the one received by CB_Appliance-1 in step 2. This packet has the destination IP address set to the IP address of server S1. Azure_Gateway-1 forwards this packet to server S1.
12. S1 processes the request packet and sends out a response packet. The destination IP address in the response packet is the IP address of client CL1, and source IP address is the IP address of server S1.
13. The response packet reaches Azure_Gateway-1. Microsoft Azure checks the stored IPSec security association (SA) parameters for processing outbound packets, as agreed between CB_Appliance-1 and Azure_Gateway-1. Microsoft Azure encrypts and encapsulates the response packet in the same way that CB_Appliance-1 encrypted and encapsulated the request packet in steps 5, 6, and 7.
14. Azure_Gateway-1 sends the resulting packet to CB_Appliance-1.
15. CB_Appliance-1, upon receiving the packet from Azure_Gateway-1, decapsulates and decrypts the packet in the same way that Azure_Gateway-1 decapsulated and decrypted the request packet in steps 9 and 10.
16. The resulting packet is the same packet that was received by Azure_Gateway-1 in step 13. This response packet has the destination IP address set to the IP address of server CL1. CB_Appliance-1 forwards the response packet to client CL1.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Points to Consider for a CloudBridge Connector tunnel Configuration

Updated: 2014-04-15

Before configuring a CloudBridge Connector tunnel between a NetScaler appliance in datacenter and Microsoft Azure, consider the following points:

1. The NetScaler appliance must have a public facing IPv4 address (type SNIP) to use as a tunnel end-point address for the CloudBridge Connector tunnel. Also, the NetScaler appliance should not be behind a NAT device.
2. Azure supports the following IPSec settings for a CloudBridge Connector tunnel. Therefore, you must specify the same IPSec settings while configuring the NetScaler for the CloudBridge Connector tunnel.
   - IKE version = v1
   - Encryption algorithm = AES
   - Hash algorithm = HMAC SHA1
3. You must configure the firewall in the datacenter edge to allow the following.
   - Any UDP packets for port 500
   - Any UDP packets for port 4500
   - Any ESP (IP protocol number 50) packets
4. IKE re-keying, which is renegotiation of new cryptographic keys between the CloudBridge Connector tunnel end points to establish new SAs, is not supported. When the Security Associations (SAs) expire, the tunnel goes into the DOWN state. Therefore, you must set a very large value for the lifetimes of SAs.
5. You must configure Microsoft Azure before specifying the tunnel configuration on the NetScaler, because the public IP address of the Azure end (gateway) of the tunnel, and the PSK, are automatically generated when you set up the tunnel configuration in Azure. You need this information for specifying the tunnel configuration on the NetScaler.

Configuring the CloudBridge Connector Tunnel

Updated: 2014-04-15

For setting up a CloudBridge Connector tunnel between your datacenter and Azure, you must install CloudBridge VPX/MPX in your datacenter, configure Microsoft Azure for the CloudBridge Connector tunnel, and then configure the NetScaler appliance in the data center for the CloudBridge Connector tunnel.

Configuring a CloudBridge Connector tunnel between a NetScaler appliance in datacenter and Microsoft Azure consists of the following tasks:

1. **Setting up the NetScaler appliance in the datacenter**. This task involves deploying and configuring a NetScaler physical appliance (MPX), or provisioning and configuring a NetScaler virtual appliance (VPX) on a virtualization platform in the datacenter.
2. **Configuring Microsoft Azure for the CloudBridge Connector tunnel**. This task involves creating local network, virtual network, and gateway entities in Azure. The local network entity specifies the IP address of the CloudBridge Connector tunnel end point (the NetScaler appliance) on the datacenter side, and the datacenter subnet whose traffic is to traverse the CloudBridge Connector tunnel. The virtual network defines a network on Azure. Creating the virtual network includes defining a subnet whose traffic is to traverse the CloudBridge Connector tunnel to be formed. You then associate the local network with the virtual network. Finally, you create a gateway that becomes the end point at the Azure end of the CloudBridge Connector tunnel.
3. **Configuring the NetScaler appliance in the datacenter for the CloudBridge Connector tunnel**. This task involves creating an IPSec profile, an IP tunnel entity, and a PBR entity in the NetScaler appliance in datacenter. The IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used in the CloudBridge Connector tunnel. The IP tunnel specifies the IP address of both the CloudBridge Connector tunnel end points (the NetScaler appliance in datacenter and the gateway in Azure) and the protocol to be used in the CloudBridge Connector tunnel. You then associate the IPSec profile entity with the IP tunnel entity. The PBR entity specifies the two subnets, in the datacenter and in the Azure cloud, that are to communicate with each other through the CloudBridge Connector tunnel. You then associate the IP tunnel entity with the PBR entity.

Configuring Microsoft Azure for the CloudBridge Connector tunnel

Updated: 2014-04-15

To create a CloudBridge Connector tunnel configuration on Microsoft Azure, use the Microsoft Windows Azure Management Portal, which is a web based graphical interface for creating and managing resources on Microsoft Azure.

Before you begin the CloudBridge Connector tunnel configuration on Azure cloud, make sure that:

- You have a user account for Microsoft Azure.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- You have a conceptual understanding of Microsoft Azure.
- You are familiar with the Microsoft Windows Azure Management Portal.

Note: The procedures for configuring Microsoft Azure for a CloudBridge Connector tunnel might change over time, depending on the Microsoft Azure release cycle. Citrix recommends the following Microsoft Azure documentation for the latest procedures.

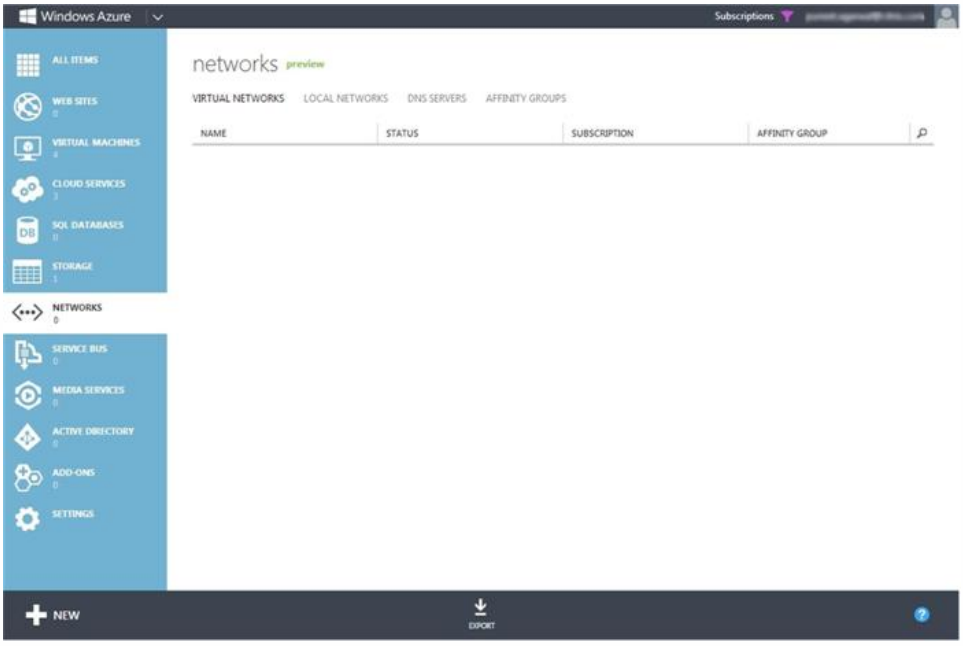
- http://www.windowsazure.com/en-us/manage/services/networking/cross-premises-connectivity/

To configure a CloudBridge Connector tunnel between a datacenter and an Azure cloud, perform the following tasks on Microsoft Azure by using the Microsoft Windows Azure Management Portal:

- **Create a local network entity**. Create a local network entity in Windows Azure for specifying the network details of the datacenter. A local network entity specifies the IP address of the CloudBridge Connector tunnel end point (the NetScaler) on the datacenter side and the datacenter subnet whose traffic is to traverse the CloudBridge Connector tunnel.
- **Create a Virtual Network**. Create virtual network entity that defines a network on Azure. This task includes defining a private address space, where you provide a range of private addresses and subnets belonging to the range specified in the address space. The traffic of the subnets will traverse the CloudBridge Connector tunnel. You then associate a local network entity with the virtual network entity. This association lets Azure create a configuration for a CloudBridge Connector tunnel between the virtual network and the data center network. A gateway (to be created) in Azure for this virtual network will be the CloudBridge end point at the Azure end of the CloudBridge Connector tunnel. You then define a private subnet for the gateway to be created. This subnet belongs to the range specified in the address space in the virtual network entity.
- **Create a gateway in Windows Azure**. Create a gateway that becomes the end point at the Azure end of the CloudBridge Connector tunnel. Azure, from its pool of public IP addresses, assigns an IP address to the gateway created.
- **Gather the public IP address of the gateway and the pre-shared key.** For a CloudBridge Connector tunnel configuration on Azure, the public IP address of the gateway and the pre-shared Key (PSK) are automatically generated by Azure. Make a note of this information. You will need it for configuring the CloudBridge Connector tunnel on the NetScaler in datacenter.

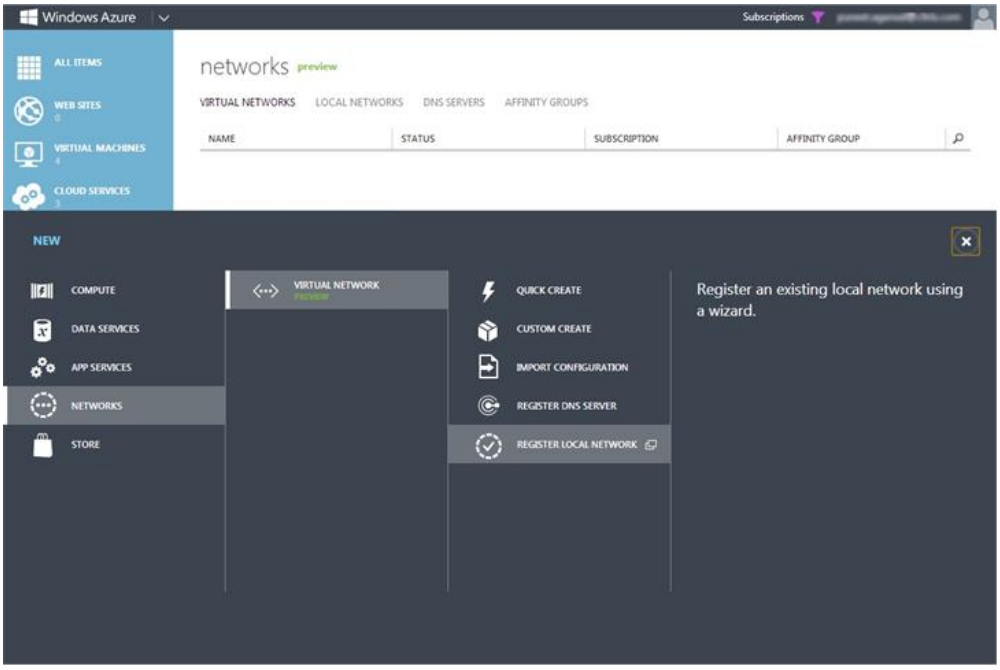
To specify a local network by using the Microsoft Windows Azure Management Portal

1. In the left pane, click NETWORKS.
2. In the lower left-hand corner of the screen, click + NEW.

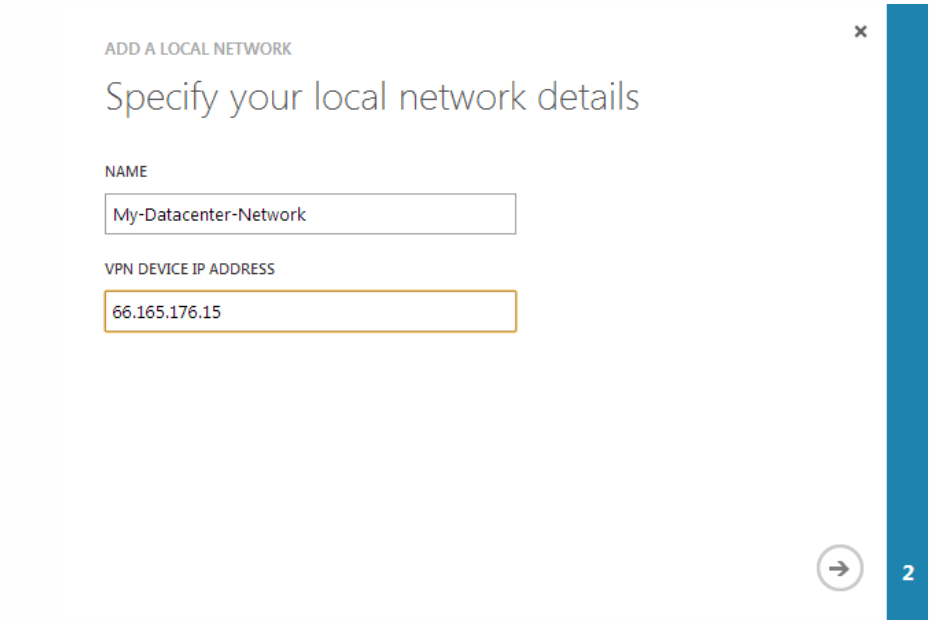


3. In the NEW navigation pane, click NETWORK, then click VIRTUAL NETWORK, and then click REGISTER LOCAL NETWORK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

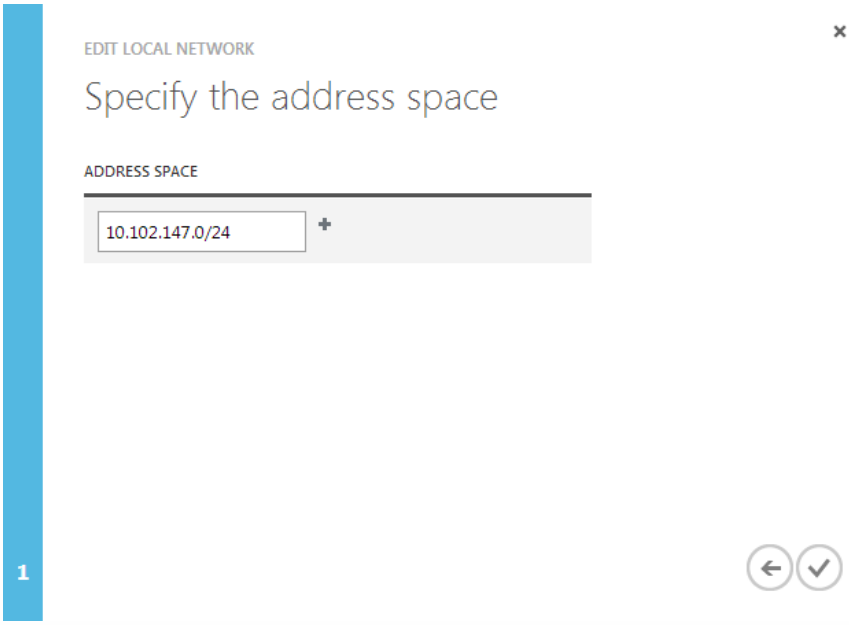


4. In the ADD A LOCAL NETWORK wizard, in the specify your local network details screen, set the following parameters:
   - NAME
   - VPN DEVICE IP ADDRESS

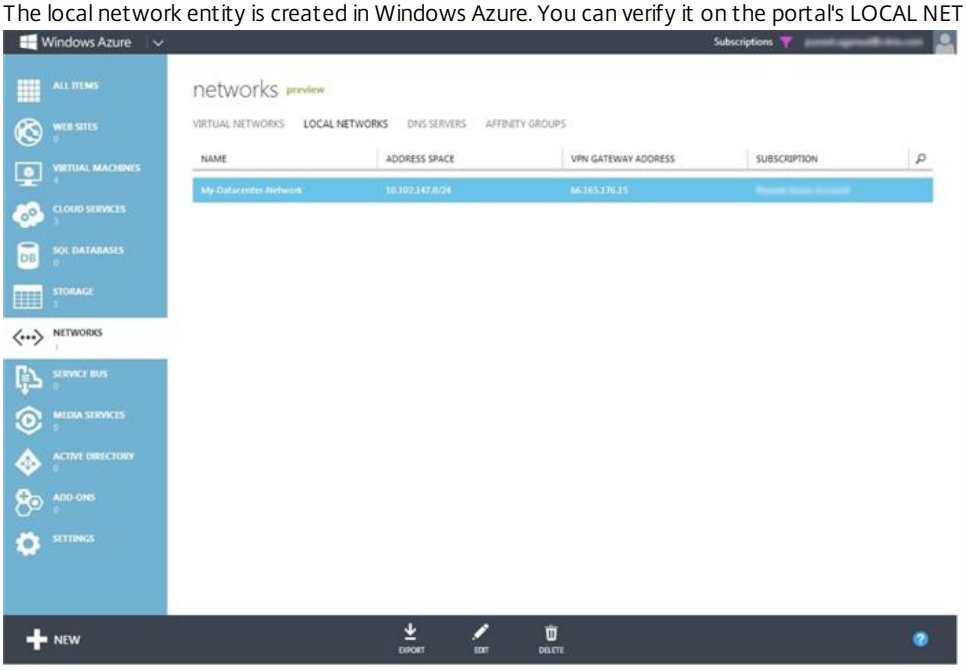


5. In the lower right corner of the screen, click -> (forward arrow mark).
6. On the Specify the address space screen, set the following parameter:
   - ADDRESS SPACE

© 1999-2017 Citrix Systems, Inc. All rights reserved.



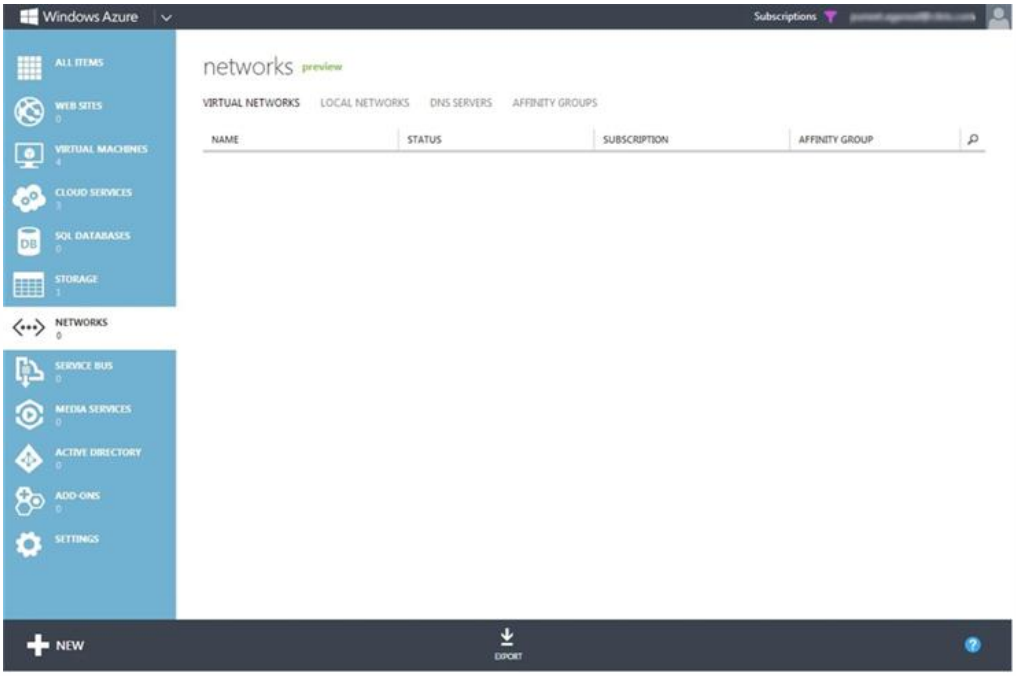
7. In the lower right corner of the screen, click the check mark.
8. The local network entity is created in Windows Azure. You can verify it on the portal's LOCAL NETWORK tab.



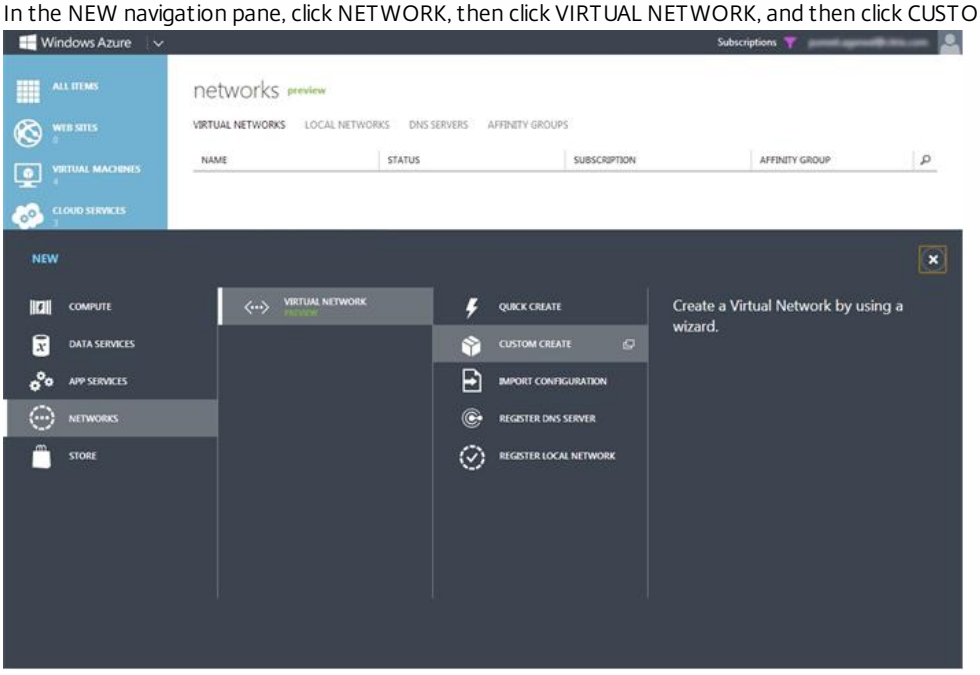
To create a virtual network in Azure by using the Microsoft Windows Azure Management Portal

1. In the left pane, click NETWORKS.
2. In the lower left-hand corner of the screen, click + New.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

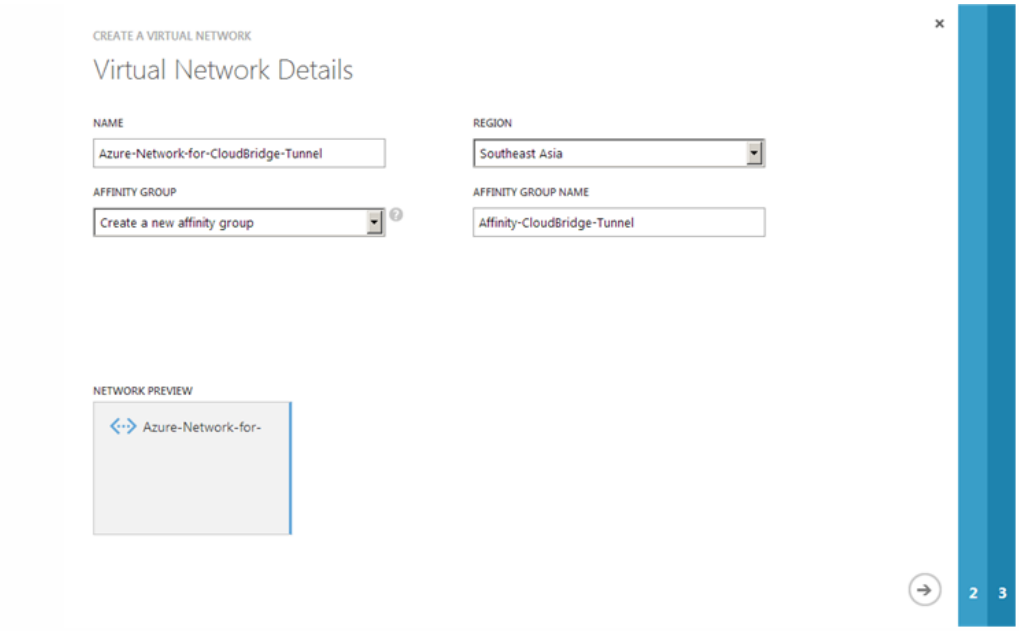


3. In the NEW navigation pane, click NETWORK, then click VIRTUAL NETWORK, and then click CUSTOM CREATE.

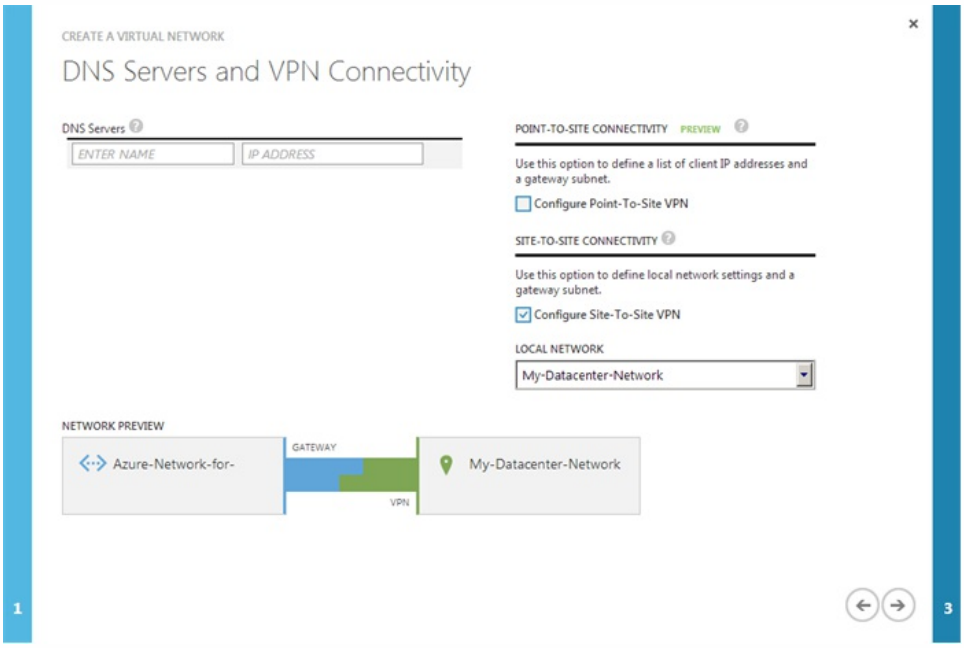


4. In the CREATE A VIRTUAL NETWORK wizard, in the Virtual Network Details screen, set the following parameters:
   - NAME
   - AFFINITY GROUP
   - REGION
   - AFFINITY GROUP NAME

© 1999-2017 Citrix Systems, Inc. All rights reserved.

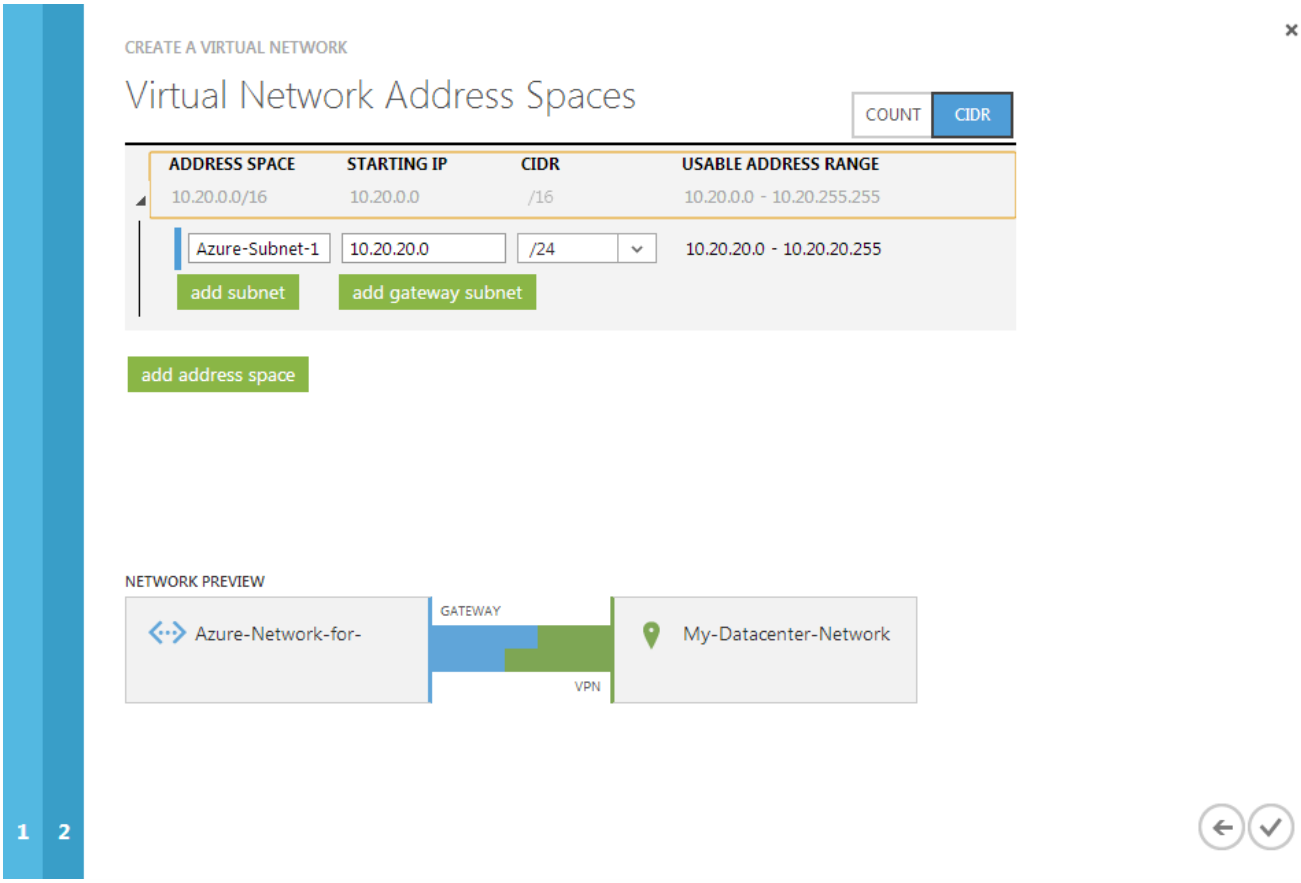


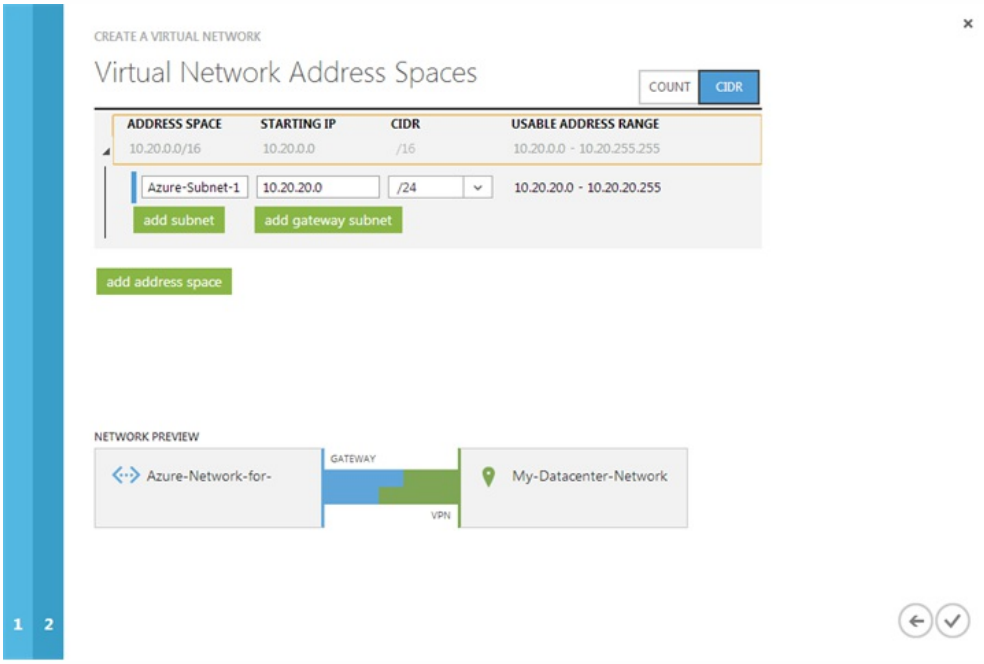
5. Click -> (forward arrow mark) in the lower right-hand corner of the screen.
6. In the DNS Servers and VPN Connectivity screen, in SITE-TO-SITE CONNECTIVITY, select Configure Site-To-Site VPN and set the following parameter:
   - LOCAL NETWORK



7. In the Address Space and Subnets screen, set the following parameters:
   - ADDRESS SPACE
   - SUBNETS
   - Gateway

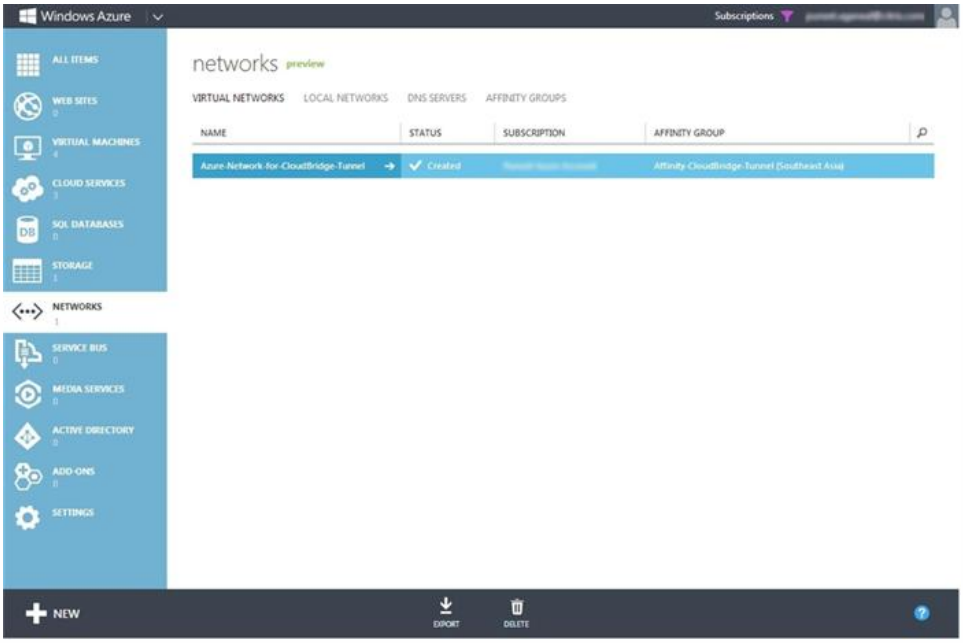
© 1999-2017 Citrix Systems, Inc. All rights reserved.





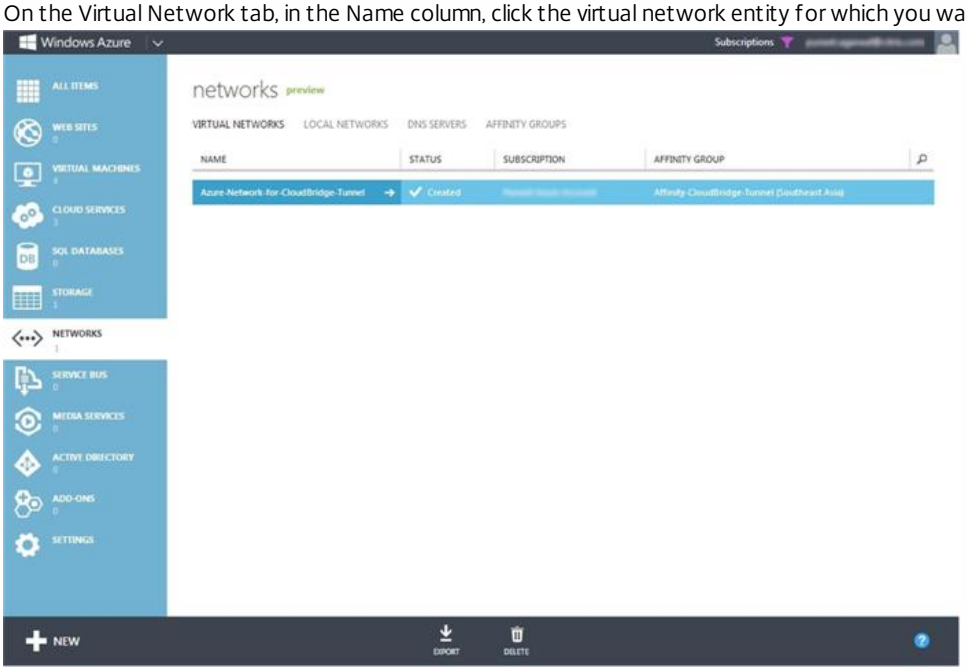
8. Click the check mark in the lower right-hand corner of the screen.
9. The virtual network is created in Windows Azure and is listed on the VIRTUAL NETWORK tab.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



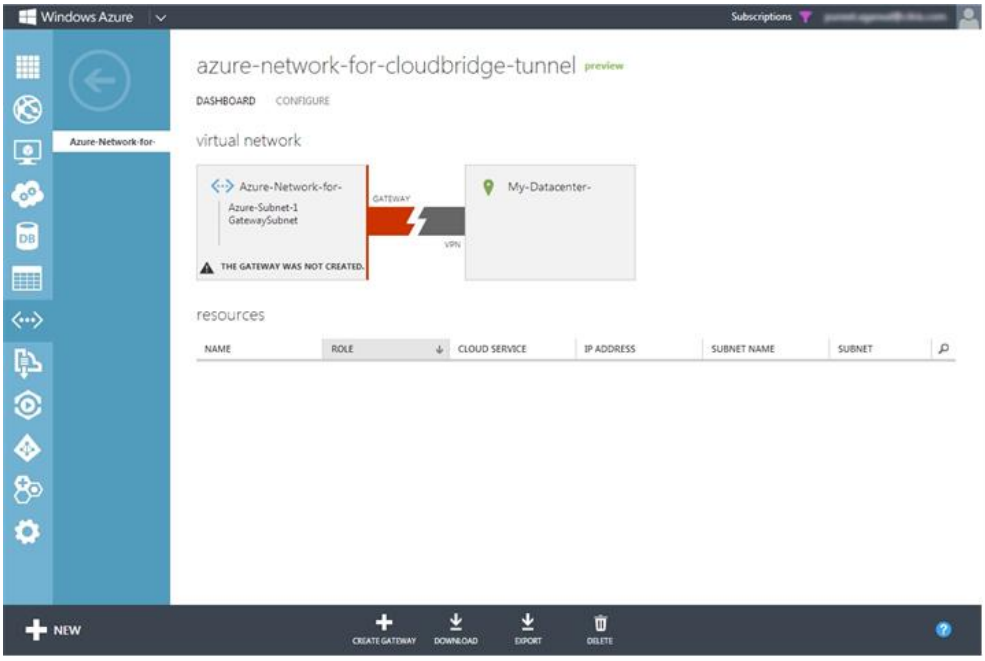
To create a gateway by using the Microsoft Windows Azure Management Portal

1. In the left pane, click NETWORKS.
2. On the Virtual Network tab, in the Name column, click the virtual network entity for which you want to create a gateway.

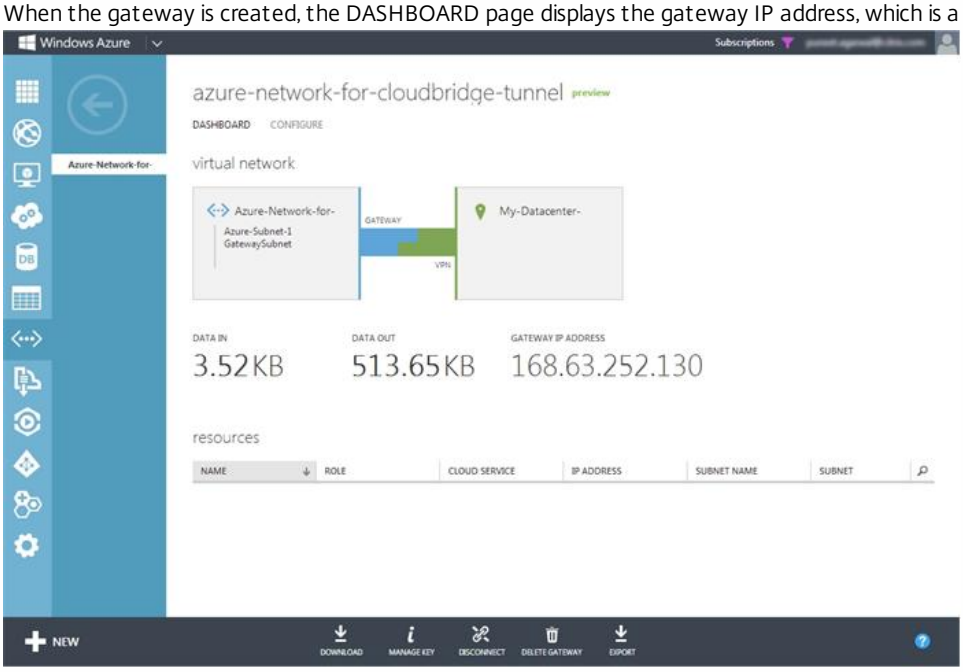


3. On the DASHBOARD page of the virtual network, at the bottom of the page, click + Create Gateway.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



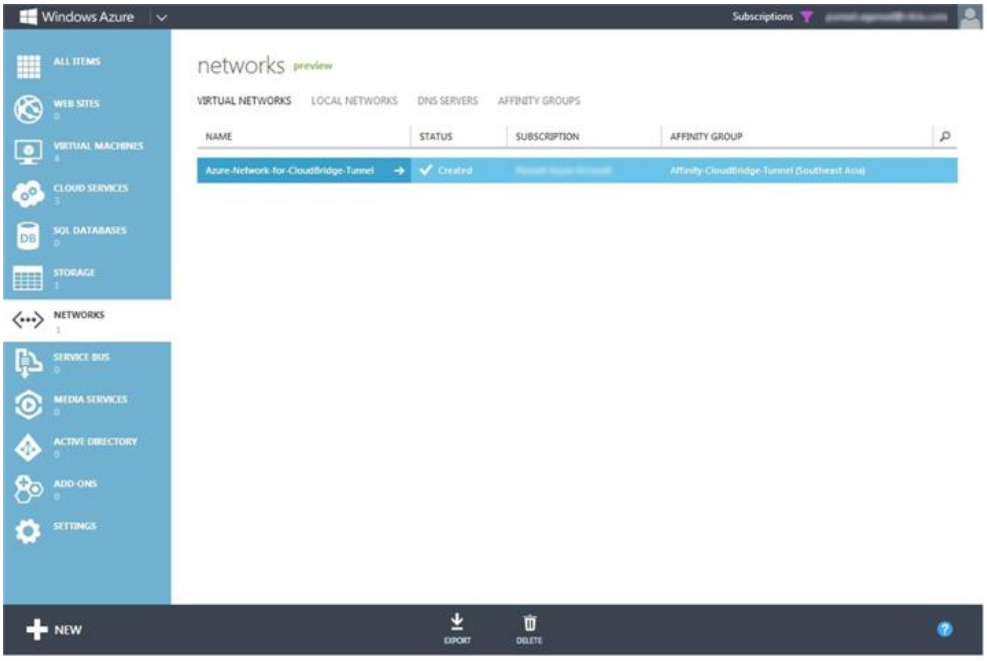
4. When prompted to confirm you want the gateway created, click YES. Creating the gateway can take up to 15 minutes.
5. When the gateway is created, the DASHBOARD page displays the gateway IP address, which is a public IP address.



To gather public IP address of the gateway and the pre-shared key information by using the Microsoft Windows Azure Management Portal

1. In the left pane, click NETWORKS.
2. On the Virtual Network tab, in the Name column, click the virtual network entity.

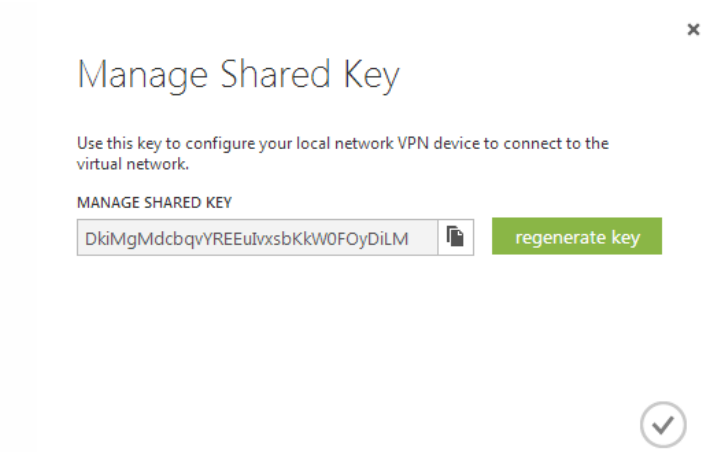
© 1999-2017 Citrix Systems, Inc. All rights reserved.



3. On the DASHBOARD page of the virtual network, copy the Gateway IP Address.

4. For the Pre Shared Key (PSK), at the bottom of the page, click MANAGE KEY.
5. In the MANAGE SHARED KEY dialog box, copy the SHARED KEY.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Configuring the NetScaler Appliance in the Datacenter for the CloudBridge Connector Tunnel

Updated: 2014-04-15

To configure a CloudBridge Connector tunnel between a datacenter and an Azure cloud, perform the following tasks on the NetScaler in the datacenter. You can use either the NetScaler command line or the configuration utility:

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used by the IPSec protocol in the CloudBridge Connector tunnel.
- **Create an IP tunnel with IPSec protocol and associate the IPSec profile to it**. An IP tunnel specifies the local IP address (a public SNIP address configured on the NetScaler appliance), remote IP address (the public IP address of the gateway in Azure), protocol (IPSec) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.
- **Create a PBR rule and associate the IP tunnel to it**. A PBR entity specifies a set of conditions and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP range are the conditions for the PBR entity. You must set the source IP address range to specify the datacenter subnet whose traffic is to traverse the tunnel, and the destination IP address range to specify the Azure subnet whose traffic is to traverse the CloudBridge Connector tunnel. Any request packet originated from a client in the subnet on the datacenter and destined to a server in the subnet on the Azure cloud matches the source and destination IP range of the PBR entity. This packet is then considered for CloudBridge Connector tunnel processing and is sent across sent across the CloudBridge Connector tunnel associated with the PBR entity.

The configuration utility combines all these tasks in a single wizard called the CloudBridge Connector wizard.

To create an IPSEC profile by using the NetScaler command line

At the Command prompt, type:

- add ipsec profile <name> -psk <string> -ikeVersion v1

To create an IPSEC tunnel and bind the IPSEC profile to it by using the NetScaler command line

At the Command prompt, type:

- add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC –ipsecProfileName <string>

To create a PBR rule and bind the IPSEC tunnel to it by using the NetScaler command line

At the Command prompt, type:

- add pbr <pbrName> ALLOW –srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>
- apply pbrs

Sample Configuration

The following commands create all settings of NetScaler appliance CB_Appliance-1 used in "Example of CloudBridge Connector Configuration and Data Flow".

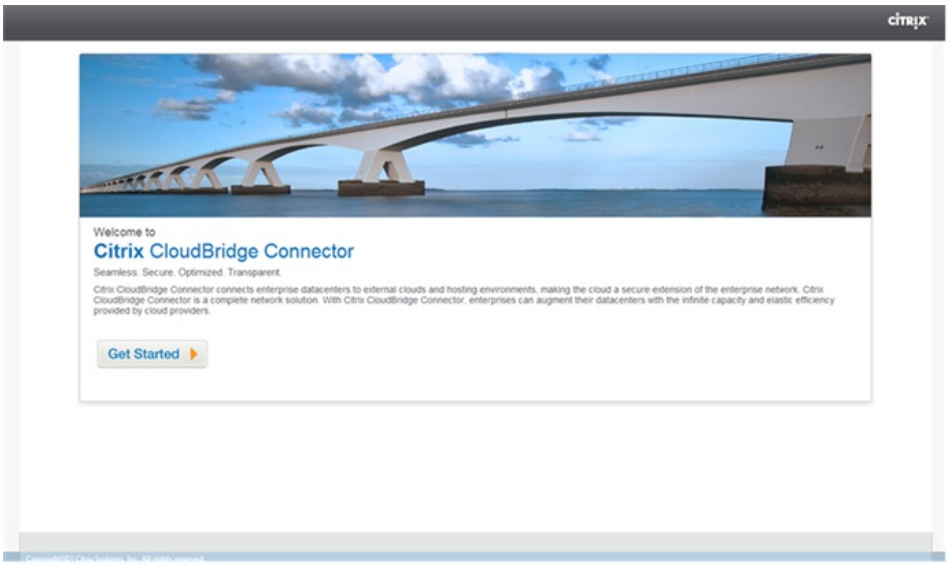
```
> add ipsec profile CB_Azure_IPSec_Profile -psk  DkiMgMdcbqvYREEuIvxsbKkW0FOyDiLM  -ikeVersion v1 –lifetime 31536000
Done
> add iptunnel CB_Azure_Tunnel 168.63.252.133 255.255.255.255 66.165.176.15 –protocol IPSEC –ipsecProfileName CB_Azure_IPSec_Profile
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 Done
> add pbr CB_Azure_Pbr -srcIP 10.102.147.0-10.102.147.255 –destIP 10.20.0.0-10.20.255.255 –ipTunnelCB_Azure_Tunnel
 Done
> apply pbrs
Done
```

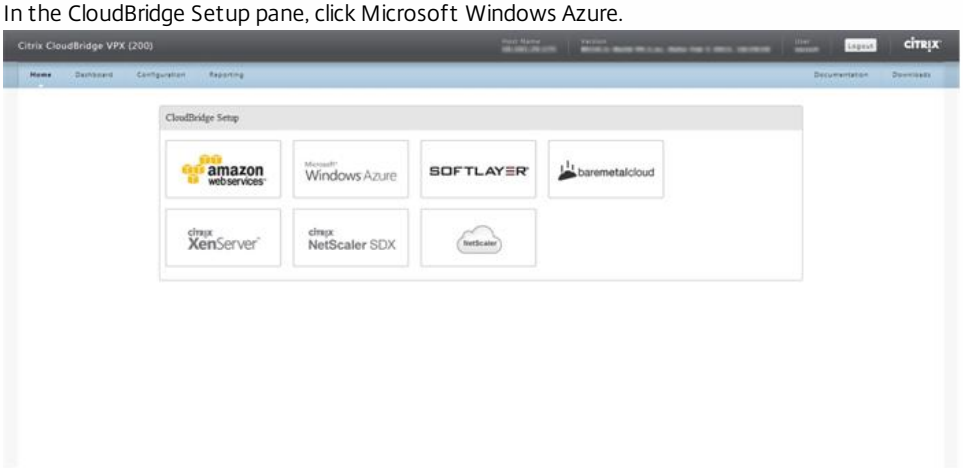
To configure a CloudBridge Connector tunnel in a NetScaler appliance by using the configuration utility

1. Access the configuration utility by using a web browser to connect to the IP address of the NetScaler appliance in the datacenter.
2. Navigate to System > CloudBridge Connector.
3. In the right pane, under Getting Started, click Create/Monitor CloudBridge.
4. Click Get Started.



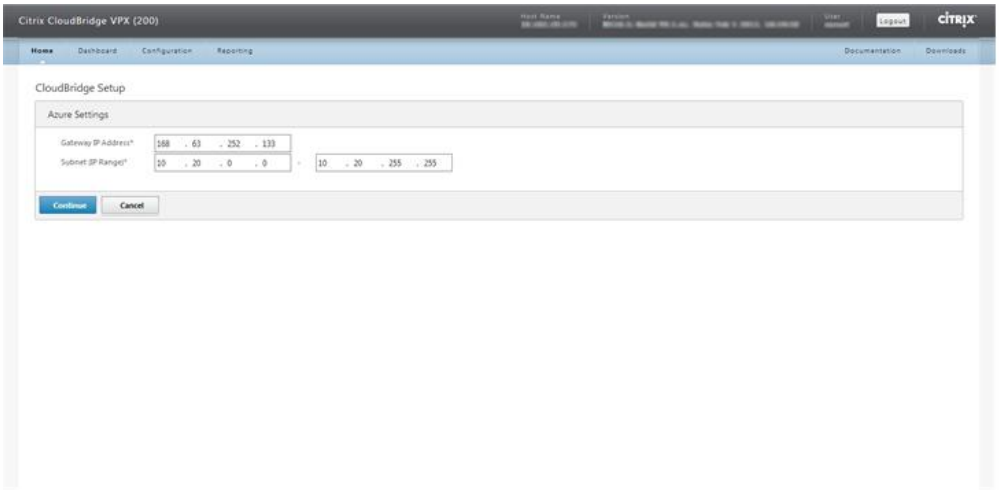
Note: If you already have any CloudBridge Connector tunnel configured on the NetScaler appliance, this screen does not appear, and you are taken to the CloudBridge Connector Setup pane.

5. In the CloudBridge Setup pane, click Microsoft Windows Azure.

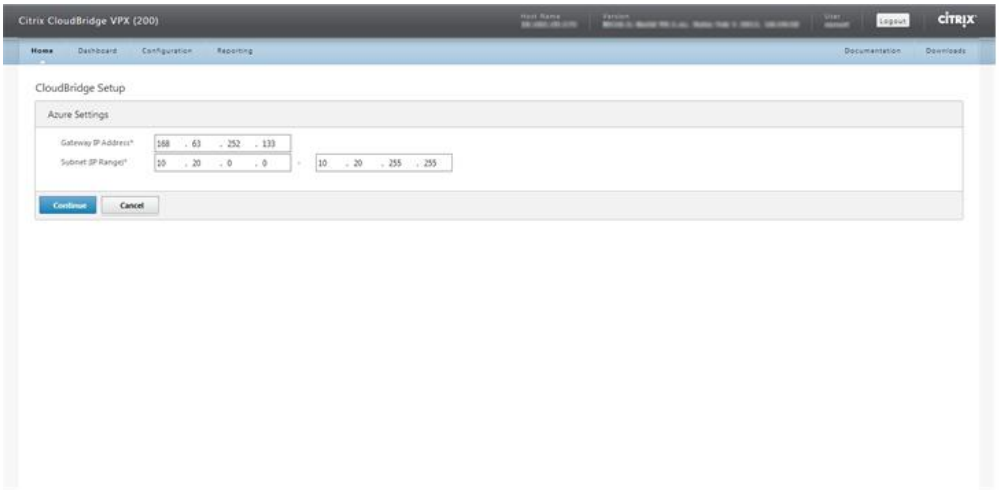


6. In the Azure Settings pane, in the Gateway IP Address* field, type the IP address of the Azure gateway. The CloudBridge Connector tunnel is then set up between the NetScaler appliance and the gateway. In the Subnet (IP Range)* text boxes, specify a subnet range (in Azure cloud), the traffic of which is to traverse the CloudBridge Connector tunnel. Click Continue.

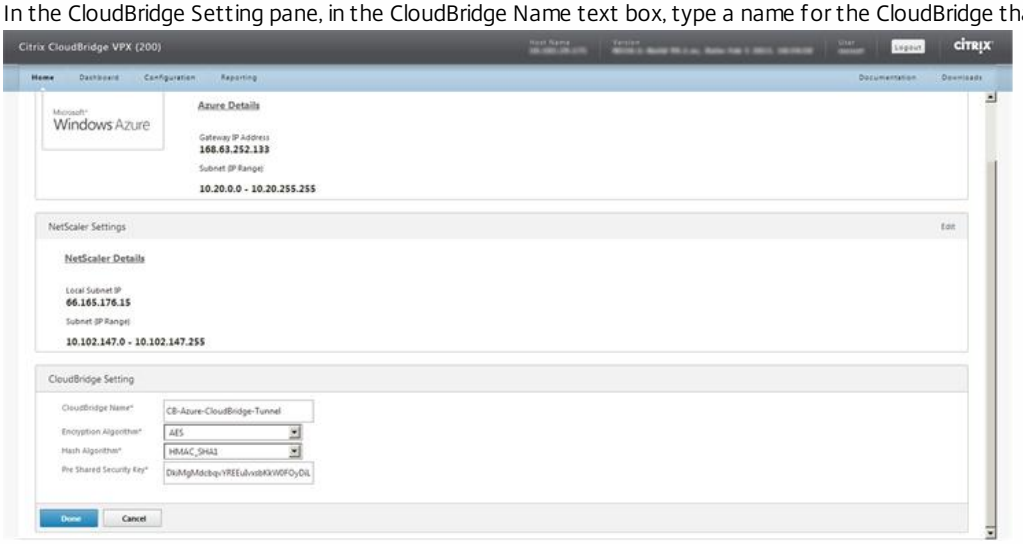
© 1999-2017 Citrix Systems, Inc. All rights reserved.



7. In the NetScaler Settings pane, from the Local Subnet IP* drop-down list, select a publicly accessible SNIP address configured on the NetScaler appliance. In Subnet (IP Range)* text boxes, specify a local subnet range, the traffic of which is to traverse the CloudBridge Connector tunnel. Click Continue.



8. In the CloudBridge Setting pane, in the CloudBridge Name text box, type a name for the CloudBridge that you want to create.



9. From the Encryption Algorithm and Hash Algorithm drop-down lists, select the AES and HMAC_SHA1 algorithms, respectively. In the Pre Shared Security Key text box, type the security key.
10. Click Done.

Monitoring the CloudBridge Connector Tunnel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2014-04-15

You can view statistics for monitoring the performance of a CloudBridge Connector tunnel between the NetScaler appliance in the datacenter and Microsoft Azure. To view CloudBridge Connector tunnel statistics on the NetScaler appliance, use the configuration utility or the NetScaler command line. To view CloudBridge Connector tunnel statistics in Microsoft Azure, use the Microsoft Windows Azure Management Portal.

Displaying CloudBridge Connector tunnel Statistics in the NetScaler appliance

The following table lists the statistical counters available for monitoring CloudBridge Connector tunnels on a NetScaler appliance.

Statistical counter	Specifies
Bytes Received	Total number of bytes received by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Bytes Sent	Total number of bytes sent by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Packets Received	Total number of packets received by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Packets Sent	Total number of packets sent by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.

All these counters are reset to 0 when the NetScaler appliance is restarted. They do not increment during the following phases:

- Internet Key Exchange (IKE) authentication (pre-shared key) phase on any configured CloudBridge Connector tunnel.
- IKE Security Association (SA) establishment phase on any configured CloudBridge Connector tunnel.

To display CloudBridge Connector tunnel statistics by using the NetScaler command line

At the command prompt, type:

- stat ipsec counters

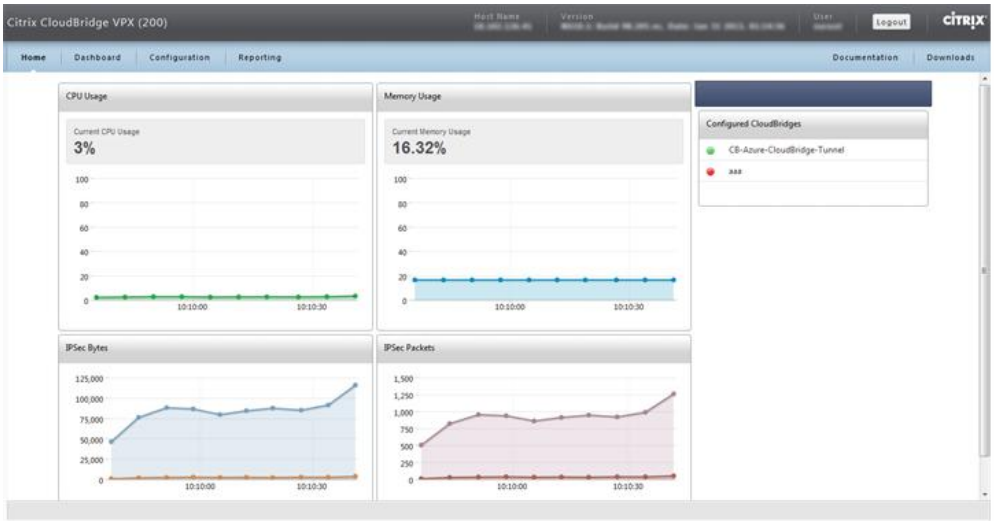
Example

```
> stat ipsec counters
Secure tunnel(s) summary
        Rate (/s)        Total
Bytes Received  0  2811248
Bytes Sent  0  157460630
Packets Received 0  56787
Packets Sent     0  200910
Done
>
```

To display CloudBridge Connector tunnel statistics by using the Configuration utility

1. Access the configuration utility by using a web browser to connect to the IP address of the NetScaler appliance.
2. On the Home tab, the IPSec Bytes and IPSec Packets charts display the statistics of all the CloudBridge Connector tunnels configured on the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



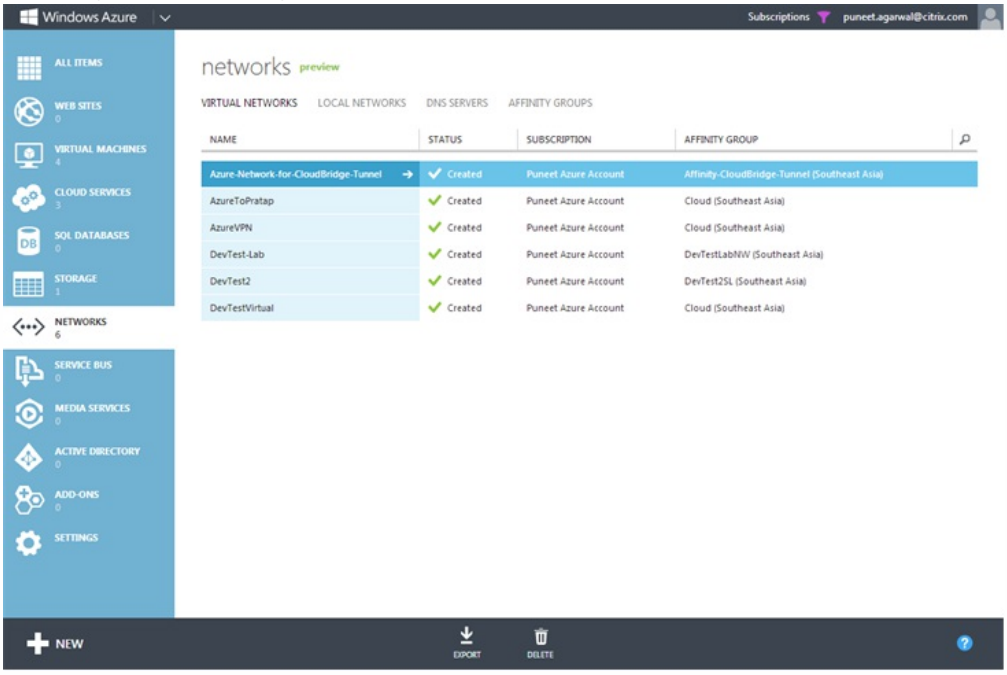
Displaying CloudBridge Connector tunnel Statistics in Microsoft Azure

The following table lists the statistical counters available for monitoring CloudBridge Connector tunnels in Microsoft Azure.

Statistical counter	Specifies
DATA IN	Total number of kilobytes received by the Azure gateway through the CloudBridge Connector tunnel since the gateway was created.
DATA OUT	Total number of kilobytes sent by the Azure gateway through the CloudBridge Connector tunnel since the gateway was created.

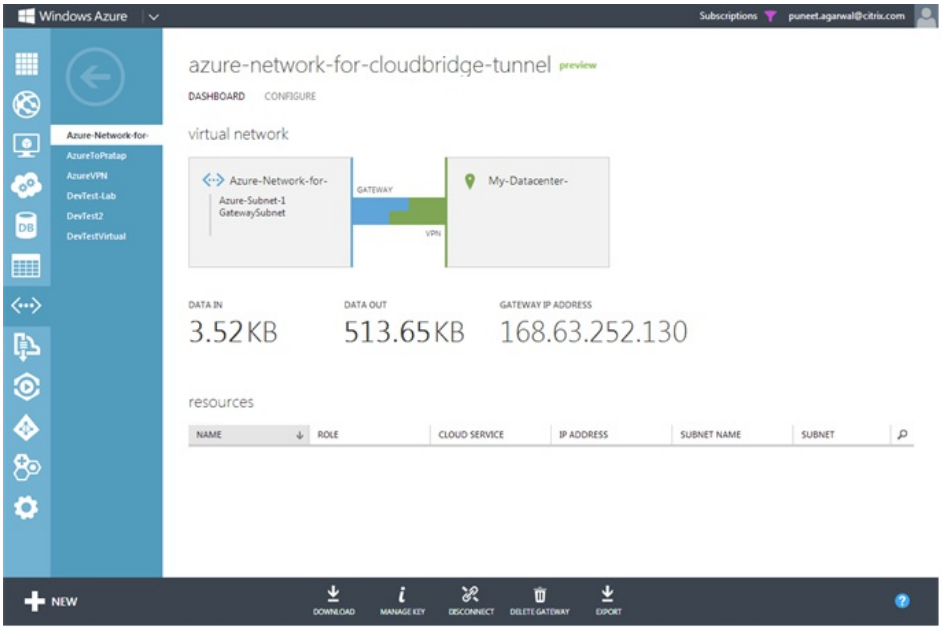
To display CloudBridge Connector tunnel statistics by using the Microsoft Windows Azure Management Portal

1. Log on to the Windows Azure Management Portal (https://manage.windowsazure.com/) by using your Microsoft Azure account credentials.
2. In the left pane, click NETWORKS.
3. On the Virtual Network tab, in the Name column, select the virtual network entity associated with a CloudBridge Connector tunnel whose statistics you want to display.



4. On the DASHBOARD page of the virtual network, view the DATA IN and DATA OUT counters for the CloudBridge Connector tunnel.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring CloudBridge Connector Tunnel between Datacenter and SoftLayer Enterprise Cloud

Jan 31, 2014

The configuration utility includes a wizard that helps you to easily configure a CloudBridge Connector tunnel between a NetScaler appliance in a datacenter and NetScaler VPX instances on the SoftLayer enterprise cloud.

When you use the wizard of the NetScaler appliance in the datacenter, the CloudBridge Connector tunnel configuration created on the NetScaler appliance, is automatically pushed to the other endpoint or peer (the NetScaler VPX on SoftLayer) of the CloudBridge Connector tunnel.

Using the wizard of the NetScaler appliance in the datacenter, you perform the following steps to configure a CloudBridge Connector tunnel.

1. Connect to the Softlayer enterprise cloud by providing the user log on credentials.
2. Select the Citrix XenServer that is running the NetScaler VPX appliance.
3. Select the NetScaler VPX appliance.
4. Provide CloudBridge Connector tunnel parameters to:
   - Configure a GRE Tunnel.
   - Configure IPsec on the GRE tunnel.
   - Create a netbridge, which is a logical representation of the CloudBridge connector, by specifying a name.
   - Bind the GRE Tunnel to the netbridge.

To configure a CloudBridge Connector tunnel by using the configuration utility

1. Log on to the configuration utility of the NetScaler appliance in the datacenter by using your account credentials for the appliance.
2. Navigate to System > CloudBridge Connector .
3. In the right pane, under Getting Started, click Create/Monitor CloudBridge Connector.
4. Click Get Started.
   Note: If you already have any CloudBridge Connector tunnel configured on the NetScaler appliance, this screen does not appear, and you are taken to the CloudBridge Connector Setup pane.
5. In the CloudBridge Connector Setup pane, click Softlayer, and then follow the instructions in the wizard.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a CloudBridge Connector Tunnel Between a NetScaler Appliance and Cisco IOS Device

Feb 03, 2016

You can configure a CloudBridge Connector tunnel between a NetScaler appliance and a Cisco device to connect two datacenters or extend your network to a Cloud provider. The NetScaler appliance and the Cisco IOS device form the end points of the CloudBridge Connector tunnel and are called peers.

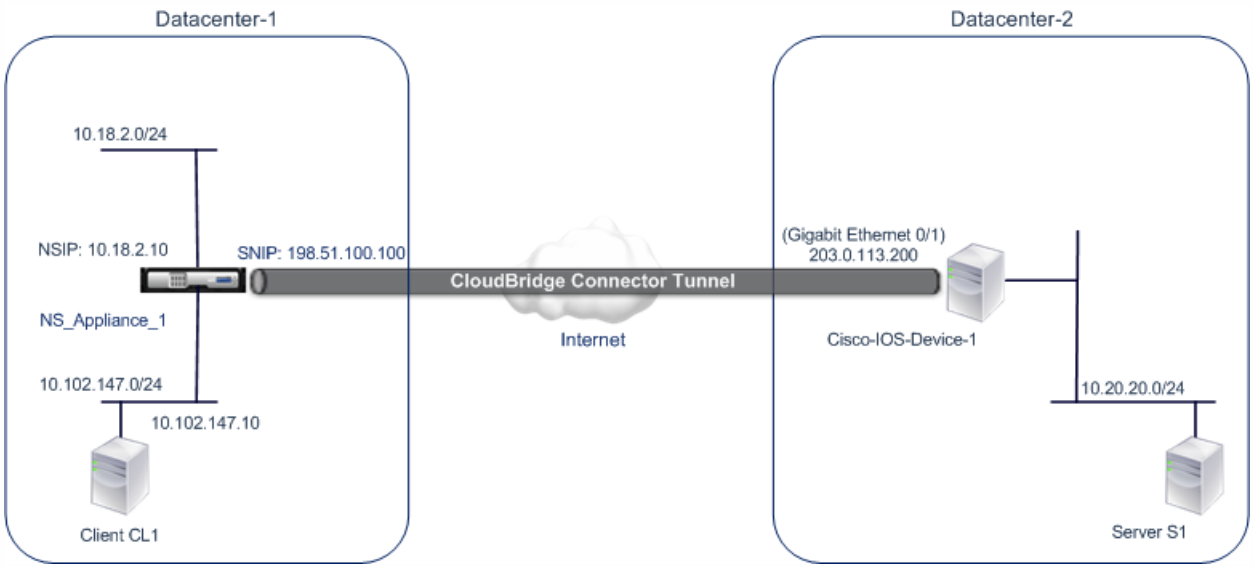
Example of CloudBridge Connector Tunnel Configuration and Data Flow

As an illustration of the traffic flow in a CloudBridge Connector tunnel, consider an example in which a CloudBridge Connector tunnel is set up between the following devices:

- NetScaler appliance NS_Appliance-1 in a datacenter designated as Datacenter-1
- Cisco IOS device Cisco-IOS-Device-1 in a datacenter designated as Datacenter-2

NS_Appliance-1 and Cisco-IOS-Device-1 enable communication between private networks in Datacenter-1 and Datacenter-2 through the CloudBridge Connector tunnel. In the example, NS_Appliance-1 and Cisco-IOS-Device-1 enable communication between client CL1 in Datacenter-1 and server S1 in Datacenter-2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On NS_Appliance-1, the CloudBridge Connector tunnel configuration includes IPSec profile entity NS_Cisco_IPSec_Profile, CloudBridge Connector tunnel entity NS_Cisco_Tunnel, and policy based routing (PBR) entity NS_Cisco_Pbr.



The following table lists the settings used in this example.

Entity	Name	Details

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Main settings of the CloudBridge Connector tunnel setup		
IP address of the CloudBridge Connector tunnel end point (NS_Appliance-1) in Datacenter-1		198.51.100.100
IP address of the CloudBridge Connector tunnel end point (Cisco-IOS-Device-1) in Datacenter-2		203.0.113.200
Datacenter–1's subnet whose traffic is to be protected over the CloudBridge Connector tunnel		10.102.147.0/24
Datacenter–2's subnet whose traffic is to be protected over the CloudBridge Connector tunnel		10.20.20.0/24
Settings on NetScaler appliance NS_Appliance-1 in Datacenter-1		
	SNIP1(for reference purposes only)	198.51.100.100
IPSec profile	• NS_Cisco_IPSec_Profile	• IKE version: v1 • Encryption algorithm = 3DES • Hash algorithm = HMAC_SHA256 • psk = examplepresharedkey (**Note**: This is an example of a pre-share key, for illustration. Do not use this string in your CloudBridge Connector configuration)
CloudBridge Connector tunnel	NS_Cisco_Tunnel	• Remote IP = 203.0.113.200 • Local IP= 198.51.100.100 • Tunnel protocol = IPSec (IPSec in tunnel mode) • IPSec profile= NS_Cisco_IPSec_Profile
Policy based route	NS_Cisco_Pbr	• Source IP range = Subnet in the Datacenter-1=10.102.147.0-10.102.147.255 • Destination IP range = Subnet in Datacenter-2 = 10.20.20.0-10.20.20.255 • IP Tunnel = NS_Cisco_Tunnel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Settings on Cisco IOS device Cisco-IOS-Device-1 in Datacenter-2		
IKE policy		• Priority: 1 • Encryption algorithm: 3des • Hash algorithm: sha256 • Authentication: pre-share • Diffie-Hellman group identifier: 2 (1024-bit Diffie-Hellman group)
Pre-share key		• Key string: examplepresharedkey (**Note**: This is an example of a pre-share key, for illustration. Do not use this string in your CloudBridge Connector configuration) • Peer address: 198.51.100.100 (IP address of type SNIP configured on NS_Appliance-1)
Crypto IPSec Transform Set	NS-CISCO-TS	• ESP Authentication: esp-sha256-hmac • ESP Encryption Algorithm: esp-3des • Mode: tunnel
Crypto access list	111	• Source: 10.20.20.0 (Subnet in the Datacenter-1) • Source wildcard: 0.0.0.255 • Destination: 10.102.147.0 (Subnet in the Datacenter-2) • Destination Wildcard: 0.0.0.255
Crypto Map	NS-CISCO-CM	• Peer address: 198.51.100.100 (IP address of type SNIP configured on NS_Appliance-1) • Crypto access list: 111 • Crypto transform set: NS-CISCO-TS
Associated Interface for the CloudBridge Connector tunnel	Gigabit Ethernet 0/1	• IP address: 203.0.113.200 • Crypto map: NS-CISCO-CM

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Following is the traffic flow in the CloudBridge Connector tunnel:

1. Client C1 sends a request to server S1.
2. The request packet is received by NS_Appliance-1. The request packet matches the condition in PBR entity NS_Cisco_Pbr, because the source IP address and the destination IP address of the request packet are in the source IP range and destination IP range specified in NS_Cisco_Pbr.
3. Because CloudBridge Connector tunnel entity NS_Cisco_Tunnel is bound to NS_Cisco_Pbr, the appliance prepares the packet to be sent across the NS_Cisco_Tunnel.
4. IPSec's Encapsulating Security Payload (ESP) protocol encrypts the request packet and then attaches an ESP header and an ESP trailer to the encrypted IP packet. The IPSec protocol encapsulates the resulting packet by adding an IP header before the ESP header. The destination address in the IP header is the IP address of Cisco-IOS-Device-1, and the source address is the SNIP1 address of NS_Appliance-1.
5. The resulting packet is sent to Cisco-IOS-Device-1.
6. Cisco-IOS-Device-1, upon receiving the packet from NS_Appliance-1, decapsulates the packet by removing the IPSec IP header, decrypts the packet, and then removes the ESP header and ESP trailer. The resulting packet is the same packet as the one received by NS_Appliance-1 in step 2. This packet has the destination IP address set to the IP address of server S1.
7. Cisco-IOS-Device-1 forwards this packet to server S1.
8. S1 processes the request packet and sends a response packet. The destination IP address in the response packet is the IP address of client CL1, and the source IP address is the IP address of server S1.
9. The response packet reaches Cisco-IOS-Device-1. The IPSec protocol in Cisco-IOS-Device-1 encrypts and encapsulates the response packet.
10. Cisco-IOS-Device-1 sends the resulting packet to NS_Appliance-1.
11. NS_Appliance-1, upon receiving the packet from Cisco-IOS-Device-1, decapsulates and decrypts the packet. The resulting packet is the same packet that was received by Cisco-IOS-Device-1 in step 9. This response packet has the destination IP address set to the IP address of server CL1.
12. NS_Appliance-1 forwards the response packet to client CL1.

Points to Consider for a CloudBridge Connector Tunnel Configuration

Before configuring a CloudBridge Connector tunnel between a NetScaler appliance and a Cisco IOS device, consider the following points:

- The following IPSec settings are supported for a CloudBridge Connector tunnel between a NetScaler appliance and a Cisco IOS device.

IPSec Properties	**Setting**
IPSec mode	Tunnel mode

© 1999-2017 Citrix Systems, Inc. All rights reserved.

IKE version	Version 1
IKE authentication method	Pre-Shared Key
IKE encryption algorithm	• AES • 3DES
IKE hash algorithm	• HMAC SHA1 • HMAC SHA256 • HMAC SHA384 • HMAC SHA512 • HMAC MD5
ESP encryption algorithm	• AES • 3DES
ESP hash algorithm	• HMAC SHA1 • HMAC SHA256 • HMAC SHA256 • HMAC SHA256 • HMAC MD5

- You must specify the same IPSec settings on the NetScaler appliance and the Cisco IOS device at the two ends of the CloudBridge Connector.
- NetScaler provides a common parameter (in IPSec profiles) for specifying an IKE hash algorithm and an ESP hash algorithm. It also provides another, and a common parameter for specifying an IKE encryption algorithm and an ESP encryption algorithm. Therefore on the Cisco device, you must specify the same hash algorithm and same encryption algorithm for IKE (while creating IKE policy) and ESP (while creating IPSec transform set).
- You must configure the firewall at the NetScaler end and Cisco device end to allow the following.
  - Any UDP packets for port 500
  - Any UDP packets for port 4500
  - Any ESP (IP protocol number 50) packets

Configuring Cisco IOS device for the CloudBridge Connector Tunnel

To configure a CloudBridge Connector tunnel on a Cisco IOS device, use the Cisco IOS command line interface, which is the primary user interface for configuring, monitoring, and maintaining Cisco devices.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Before you begin the CloudBridge Connector tunnel configuration on a Cisco IOS device, make sure that:

- You have a user account with administrator credentials on the Cisco IOS device.
- You are familiar with the Cisco IOS command line interface.
- The Cisco IOS device is UP and running, is connected to the Internet, and is also connected to the private subnets whose traffic is to be protected over the CloudBridge Connector tunnel.

Note: The procedures for configuring CloudBridge Connector tunnel on a Cisco IOS device might change over time, depending on the Cisco release cycle. Citrix recommends that you follow the official Cisco product documention for Configuring IPSec VPN tunnels, at:

- http://www.cisco.com

To configure a CloudBridge connector tunnel between a NetScaler appliance and a Cisco IOS device, perform the following tasks on the Cisco device's IOS command line:

- **Create an IKE Policy**. An IKE policy defines a combination of security parameters to be used during the IKE negotiation. For example, parameters such as hash algorithm, encryption algorithm, Diffie-Hellman group, and authentication method to be used in the IKE negotiation are set in this task.
- **Configure a Pre-shared key for IKE authentication**. A pre-shared key is a text string, which the peers of a CloudBridge Connector tunnel use to mutually authenticate with each other. The pre-shared keys are matched against each other for IKE authentication. Therefore, for the authentication to be successful, you must configure the same pre-shared key on the Cisco device and the NetScaler appliance.
- **Define a transform set and configure IPSec in tunnel mode**. A transform set defines a combination of security parameters to be used in the exchange of data over the CloudBridge Connector tunnel after the IKE negotiation is successful. Parameters such as hash algorithm and encryption algorithm are set in this task. This task also specifies that IPSec in tunnel mode be used for the CloudBridge tunnel.
- **Create a crypto access List**. Crypto access lists are used to define the subnets whose IP traffic will be protected over the CloudBridge tunnel. The source and destination parameters in the access list specify the Cisco device side and NetScaler side subnets that are to be protected over the CloudBridge Connector Tunnel. The access list must be set to permit. Any request packet that originates from a device in the Cisco device side subnet and is destined to a device in the NetScaler side subnet, and that matches the source and destination parameters of the access list, is sent across the CloudBridge Connector tunnel.
- **Create a crypto map**. Crypto maps define the IPSec parameters to be negotiated with peer. They include the following: Crypto access list to identify the subnets whose traffic is to be protected over the CloudBridge tunnel, peer (NetScaler) identification by IP address, and transform set to match the peer security settings.
- **Apply the crypto Map to an interface**. In this task, you apply a crypto map to an interface through which CloudBridge Connector tunnel traffic will flow. Applying the crypto map to an interface instructs the Cisco IOS device to evaluate all the interface traffic against the crypto map set, and to use the specified policy during connection or SA negotiation on behalf of subnet's traffic to be protected.

The examples in the following procedures create settings of Cisco IOS device Cisco-IOS-Device-1 used in "Example of CloudBridge Connector Configuration and Data Flow."

To create an IKE policy by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following commands, starting in global configuration mode, in the order shown:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command	Example	Command Description
crypto isakmp policy *priority*	Cisco-ios-device-1(config)# crypto isakmp policy 1	Enter config-isakmp command mode and identify the policy to create. (Each policy is uniquely identified by the priority number you assign.) This example configures policy 1.
encryption (**3des** | **aes**)	Cisco-ios-device-1 (config-isakmp)# **encryption 3des**	Specify the encryption algorithm. This example configures the 3DES algorithm.
hash (**sha** | **sha256** | **sha384** | **md5**)	Cisco-ios-device-1 (config-isakmp)# **hash sha256**	Specify the hash algorithm. This example configures SHA256.
authentication pre-share	Cisco-ios-device-1 (config-isakmp)# **authentication pre-share**	Specify the pre-share authentication method. **Note**: RSA encrypted nonces (**rsa-encr**), RSA signatures (**rsa-slg**), and digital certificate authentication methods are not supported.
group 2	Cisco-ios-device-1 (config-isakmp)# **group 1**	Specify 1024-bit Diffie-Hellman group identifier (**2**).
lifetime seconds	Cisco-ios-device-1 (config-isakmp)# **lifetime 86400**	Specify the security association's lifetime in seconds. This example configures 86400 seconds (one day).
exit	Cisco-ios-device-1 (config-isakmp)# **exit** Cisco-ios-device-1 (config)#	Exit back to global configuration mode.

To configure a pre-shared key by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following commands, starting in global configuration mode, in the order shown:

Command	Example	Command Description

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crypto isakmp identity address	Cisco-ios-device-1(config)# **crypto isakmp identity address**	Specify the ISAKMP identity (**address**) for the Cisco IOS device to use when communicating with the peer (NetScaler appliance) during IKE negotiations. This example specifies the **address** keyword, which uses IP address 203.0.113.200 (Gigabit Ethernet interface **0/1** of Cisco-IOS-Device-1) as the identity for the device.
crypto isakmp key *keystring* **address** *peer-address*	Cisco-ios-device-1 (config)# **crypto isakmp key examplepresharedkey address 198.51.100.100**	Specify a pre-shared key for the IKE authentication. This example configures shared key examplepresharedkey to be used with the NetScaler appliance NS_Appliance-1 (198.51.100.100). The same pre-shared key must be configured on the NetScaler appliance for IKE authentication to be successful between the Cisco IOS device and the NetScaler appliance.

To create a crypto access list by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following command in global configuration mode, in the order shown:

Command	Example	Command Description
access-list *access-list-number* **permit IP** *source source-wildcard destination destination-wildcard*	Cisco-ios-device-1(config)# **access-list 111 permit ip 10.20.20.0 0.0.0.255 10.102.147.0 0.0.0.255**	Specify conditions to determine the subnets whose IP traffic is to be protected over the CloudBridge Connector tunnel. This example configures access list 111 to protect traffic from subnets 10.20.20.0/24 (at the Cisco-IOS-Device-1 side) and 10.102.147.0/24 (at the NS_Appliance-1 side).

To define a transform and configure IPSec tunnel mode by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following commands, starting in global configuration mode, in the order shown:

Command	Example	Command Description

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crypto ipsec transform-set *name ESP_Authentication_Transform ESP_Encryption_Transform* **Note**: *ESP_Authentication_Transform* can take the following values: ● esp-sha-hmac ● esp-sha256-hmac ● esp-sha384-hmac ● esp-sha512-hmac ● esp-md5-hmac *ESP_Encryption_Transform* can take the following values: ● esp-aes ● esp-3des	Cisco-ios-device-1(config)# **crypto ipsec transform-set NS-CISCO-TS esp-sha256-hmac esp-3des**	Define a transform set and specify the ESP hash algorithm (for authentication) and the ESP encryption algorithm to be used during exchange of data between the CloudBridge Connector tunnel peers. This example defines transform set NS-CISCO-TS and specifies ESP authentication algorithm as esp-sha256-hmac, and ESP encryption algorithm as esp-3des.
mode tunnel	Cisco-ios-device-1 (config-crypto-trans)# **mode tunnel**	Set IPSec in tunnel mode.
exit	Cisco-ios-device-1 (config-crypto-trans)# **exit** Cisco-ios-device-1 (config)#	Exit back to global configuration mode.

To create a crypto map by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following commands starting in global configuration mode, in the order shown:

Command	Example	Command Description
crypto map *map-name seq-num* **ipsec-isakmp**	Cisco-ios-device-1 (config)# **crypto map NS-CISCO-CM 2 ipsec-isakmp**	Enter crypto map configuration mode, specify a sequence number for the crypto map, and configure the crypto map to use IKE to establish security associations (SAs). This example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

		configures sequence number 2 and IKE for crypto map NS-CISCO-CM.
set peer *ip-address*	Cisco-ios-device-1 (config-crypto-map)# **set peer 172.23.2.7**	Specify the peer (NetScaler appliance) by its IP address. This example specifies 198.51.100.100, which is the CloudBridge Connector endpoint IP address on the NetScaler appliance.
match address *access-list-id*	Cisco-ios-device-1 (config-crypto-map)# **match address 111**	Specify an extended access list. This access list specifies conditions to determine the subnets whose IP traffic is to be protected over the CloudBridge Connector tunnel. This example specifies access list 111.
set transform-set *transform-set-name*	Cisco-ios-device-1 (config-crypto-map)# **set transform-set NS-CISCO-TS**	Specify which transform sets are allowed for this crypto map entry. This example specifies transform set NS-CISCO-TS.
exit	Cisco-ios-device-1 (config-crypto-map)# **exit** Cisco-ios-device-1 (config)#	Exit back to global configuration mode.

To apply a crypto map to an interface by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following commands starting in global configuration mode, in the order shown:

Command	Example	Command Description
interface *interface-ID*	Cisco-ios-device-1(config)# **interface GigabitEthernet 0/1**	Specify a physical interface to which to apply the crypto map and enter interface configuration mode. This example specifies Gigabit Ethernet interface 0/1 of the Cisco device Cisco-IOS-Device-1. IP address 203.0.113.200 is

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

		already set to this interface.
crypto map *map-name*	Cisco-ios-device-1 (config-if)# **crypto map NS-CISCO-CM**	Apply the crypto map to the physical interface. This example applies crypto map NS-CISCO-CM.
exit	Cisco-ios-device-1 (config-if)# **exit** Cisco-ios-device-1 (config)#	Exit back to global configuration mode.

Configuring the NetScaler Appliance for the CloudBridge Connector Tunnel

To configure a CloudBridge Connector tunnel between a NetScaler appliance and a Cisco IOS device, perform the following tasks on the NetScaler appliance. You can use either the NetScaler command line or the NetScaler graphical user interface (GUI):

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and authentication method (Pre-Shared key) to be used by the IPSec protocol in the CloudBridge Connector tunnel.
- **Create an IP tunnel that uses IPSec protocol, and associate the IPSec profile with it**. An IP tunnel specifies the local IP address (CloudBridge Connector tunnel end point IP address (of type SNIP) configured on the NetScaler appliance), remote IP address (CloudBridge Connector tunnel endpoint IP address configured on the Cisco IOS device), protocol (IPSec) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.
- **Create a PBR rule and associate it with the IP tunnel**. A PBR entity specifies a set of rules and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP address range are the conditions for the PBR entity. Set the source IP address range to specify the NetScaler-side subnet whose traffic is to be protected over the tunnel, and set the destination IP address range to specify the Cisco IOS device side subnet whose traffic is to be protected over the tunnel. Any request packet that originates from a client in the subnet on the NetScaler side and is destined to a server in the Cisco IOS device side subnet, and matches the source and destination IP range of the PBR entity, is sent across the CloudBridge Connector tunnel associated with the PBR entity. Apply the PBR rule to make it functional.

To create an IPSEC profile by using the NetScaler command line

At the Command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **add ipsec profile** <name> -**psk** <string> -**ikeVersion** v1

To create an IPSEC tunnel and bind the IPSEC profile to it by using the NetScaler command line

At the Command prompt, type:

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> -**protocol** IPSEC –**ipsecProfileName** <string>

To create a PBR rule and bind the IPSEC tunnel to it by using the NetScaler command line

At the Command prompt, type:

- **add pbr** <pbrName> ALLOW –**srcIP** <subnet-range> -**destIP** <subnet-range> -**ipTunnel** <tunnelName>
- **apply pbrs**

Sample Configuration COPY

```
The following commands create settings of NetScaler appliance NS_Appliance-1 in "Example of CloudBridge Connector Configuration an

> add ipsec profile NS_Cisco_IPSec_Profile -psk  examplepresharedkey -ikeVersion v1 –lifetime 315360 –encAlgo 3DES

Done

> add iptunnel NS_Cisco_Tunnel 203.0.113.200 255.255.255.255 198.51.100.100 –protocol IPSEC –ipsecProfileName NS_Cisco_IPSec

Done

> add pbr NS_Cisco_Pbr -srcIP 10.102.147.0-10.102.147.255 –destIP 10.20.0.0-10.20.255.255 –ipTunnel NS_Cisco_Tunnel

Done

> apply pbrs

Done
```

To create an IPSEC profile by using the configuration utility

1. Navigate to **System** > **CloudBridge Connector** > **IPSec Profile**.
2. In the details pane, click **Add**.
3. In the **Add IPSec Profile** dialog box, set the following parameters:
   - Name
   - Encryption Algorithm
   - Hash Algorithm
   - IKE Protocol Version
4. Configure the IPSec authentication method to be used by the two CloudBridge Connector tunnel peers to mutually authenticate: Select the **Pre-shared key** authentication method and set the **Pre-Shared Key Exists** parameter.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Click **Create**, and then click **Close**.

To create an IP tunnel and bind the IPSEC profile to it by using the NetScaler GUI

1. Navigate to **System** > **CloudBridge Connector** > **IP Tunnels**.
2. On the **IPv4 Tunnels** tab, click **Add**.
3. In the **Add IP Tunnel** dialog box, set the following parameters:

   - Name
   - Remote IP
   - Remote Mask
   - Local IP Type (In the Local IP Type drop down list, select Subnet IP).
   - Local IP (All the configured IPs of the selected IP type are in the Local IP drop down list. Select the desired IP from the list.)
   - Protocol
   - IPSec Profile
4. Click **Create**, and then click **Close**.

To create a PBR rule and bind the IPSEC tunnel to it by using the NetScaler GUI

1. Navigate to **System** > **Network** > **PBR**.
2. On the **PBR** tab, click **Add**.
3. In the **Create PBR** dialog box, set the following parameters:
   - Name
   - Action
   - Next Hop Type (Select IP Tunnel)
   - IP Tunnel Name
   - Source IP Low
   - Source IP High
   - Destination IP Low
   - Destination IP High

4. Click **Create**, and then click **Close**.

To apply a PBR by using the NetScaler GUI

1. Navigate to **System** > **Network** > **PBRs**.
2. On the **PBRs** tab, select the **PBR**, in the **Action** list, select **Apply**.

The corresponding new CloudBridge Connector tunnel configuration on the NetScaler appliance appears in the configuration utility. The current status of the CloudBridge connector tunnel is shown in the Configured CloudBridge Connector pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

Monitoring the CloudBridge Connector Tunnel

You can view statistics for monitoring the performance of a CloudBridge Connector tunnel between a NetScaler appliance and a Cisco IOS device. To view CloudBridge Connector tunnel statistics on the NetScaler appliance, use the NetScaler command line.

The following table lists the statistical counters available for monitoring CloudBridge Connector tunnels on a NetScaler appliance.

Statistical counter	Specifies

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Bytes Received	Total number of bytes received by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Bytes Sent	Total number of bytes sent by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Packets Received	Total number of packets received by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Packets Sent	Total number of packets sent by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.

All these counters are reset to 0 when the NetScaler appliance is restarted. They do not increment during the following phases:

- Internet Key Exchange (IKE) authentication (pre-shared key) phase on any configured CloudBridge Connector tunnel.
- IKE Security Association (SA) establishment phase on any configured CloudBridge Connector tunnel.

To display CloudBridge Connector tunnel statistics by using the command line

At the command prompt, type:

- **stat ipsec counters**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

High Availability

Mar 19, 2012

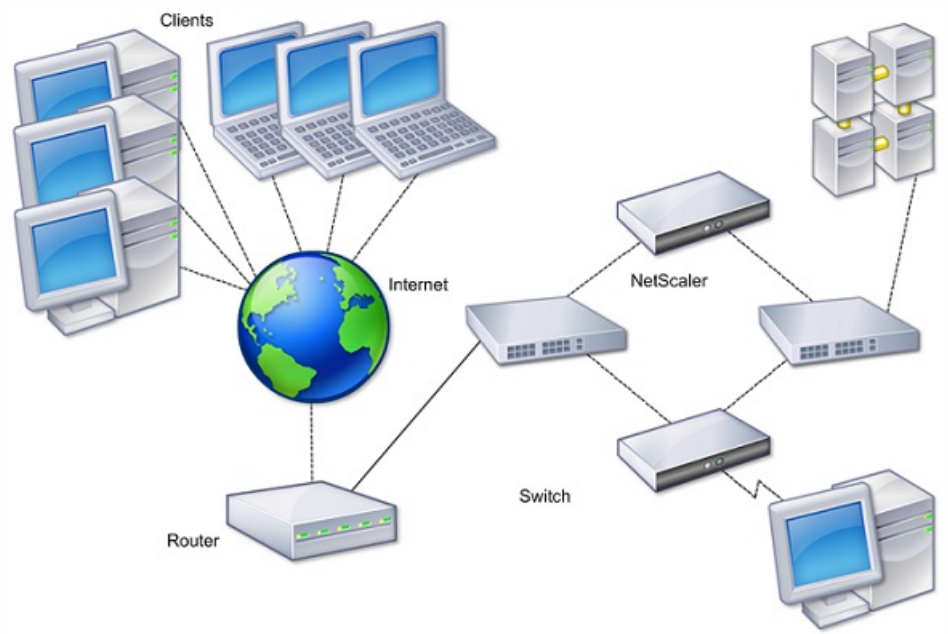
A high availability (HA) deployment of two Citrix® NetScaler® appliances can provide uninterrupted operation in any transaction. With one appliance configured as the primary node and the other as the secondary node, the primary node accepts connections and manages servers while the secondary node monitors the primary. If, for any reason, the primary node is unable to accept connections, the secondary node takes over.

The secondary node monitors the primary by sending periodic messages (often called heartbeat messages or health checks) to determine whether the primary node is accepting connections. If a health check fails, the secondary node retries the connection for a specified period, after which it determines that the primary node is not functioning normally. The secondary node then takes over for the primary (a process called failover).

After a failover, all clients must reestablish their connections to the managed servers, but the session persistence rules are maintained as they were before the failover.

With Web server logging persistence enabled, no log data is lost due to the failover. For logging persistence to be enabled, the log server configuration must carry entries for both systems in the log.conf file.

The following figure shows a network configuration with an HA pair.
Figure 1. NetScaler Appliances in a High Availability Configuration



To configure HA, you might want to begin by creating a basic setup, with both nodes in the same subnet. You can then customize the intervals at which the nodes communicate health-check information, the process by which nodes maintain synchronization, and the propagation of commands from the primary to the secondary. You can configure fail-safe mode to prevent a situation in which neither node is primary. If your environment includes devices that do not accept NetScaler gratuitous ARP messages, you should configure virtual MAC addresses. When you are ready for a more complex configuration, you can configure HA nodes in different subnets.

To improve the reliability of your HA setup, you can configure route monitors and create redundant links. In some situations, such as when troubleshooting or performing maintenance tasks, you might want to force a node to fail over (assign primary

© 1999-2017 Citrix Systems, Inc. All rights reserved.

status to the other node), or you might want to force the secondary node to stay secondary or the primary node to stay primary.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Considerations for a High Availability Setup

Mar 19, 2012

Note the following requirements for configuring systems in an HA setup:

- In an HA configuration, the primary and secondary NetScaler appliances should be of the same model. Different NetScaler models are not supported in an HA pair (for example, you cannot configure a 10010 model and a 7000 model as an HA pair).
- In an HA setup, both nodes must run the same version of NetScaler, for example, nCore/nCore or classic/classic. If the nodes are running NetScaler classic and you want to migrate to NetScaler nCore of the same NetScaler release, prop and sync are not supported during the migration process. Once migration is complete, prop and sync are auto-enabled. The same applies if you migrate from NetScaler nCore to NetScaler classic.
- Entries in the configuration file (ns.conf) on both the primary and the secondary system must match, with the following exceptions:
  - The primary and the secondary systems must each be configured with their own unique NetScaler IP addresses (NSIPs.)
  - In an HA pair, the node ID and associated IP address of one node must point to the other node. For example, if you have nodes NS1 and NS2, you must configure NS1 with a unique node ID and the IP address of NS2, and you must configure NS2 with a unique node ID and the IP address of NS1.
- If you create a configuration file on either node by using a method that does not go directly through the GUI or the CLI (for example, importing SSL certificates, or changing to startup scripts), you must copy the configuration file to the other node or create an identical file on that node.
- Initially, all NetScaler appliances are configured with the same RPC node password. RPC nodes are internal system entities used for system-to-system communication of configuration and session information. For security, you should change the default RPC node passwords.

  One RPC node exists on each NetScaler. This node stores the password, which is checked against the password provided by the contacting system. To communicate with other systems, each NetScaler requires knowledge of those systems, including how to authenticate on those systems. RPC nodes maintain this information, which includes the IP addresses of the other systems, and the passwords they require for authentication.

  RPC nodes are implicitly created when adding a node or adding a Global Server Load Balancing (GSLB) site. You cannot create or delete RPC nodes manually.

  Note: If the NetScaler appliances in a high availability setup are configured in one-arm mode, you must disable all system interfaces except the one connected to the switch or hub.
- For an IPv6 HA configuration, the following considerations apply:
  - You must install the IPv6PT license on both NetScaler appliances.
  - After installing the IPv6PT license, enable the IPv6 feature by using the configuration utility or the command line interface.
  - Both NetScaler appliances require a global NSIP IPv6 address. In addition, network entities (for example, switches and routers) between the two nodes must support IPv6.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring High Availability

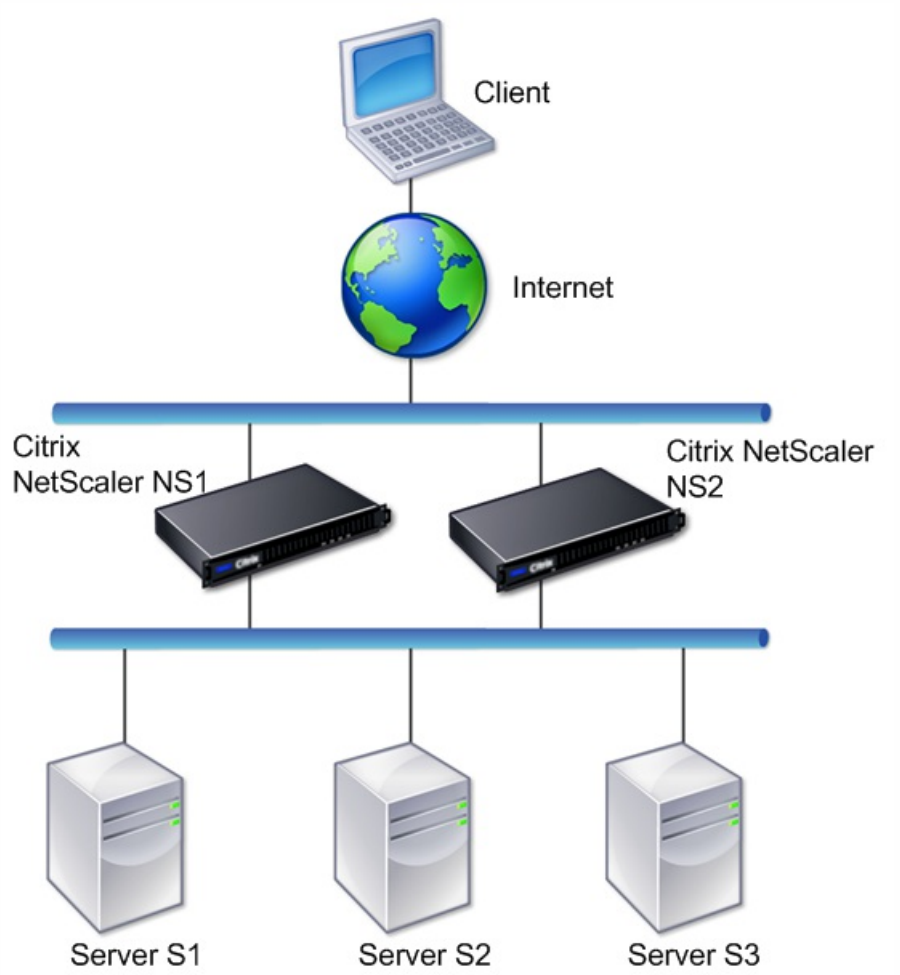
Sep 07, 2016

To set up a high availability configuration, you create two nodes, each of which defines the other's NetScaler IP (NSIP) address as a remote node. Begin by logging on to one of the two NetScaler appliances that you want to configure for high availability, and add a node. Specify the other appliance's NetScaler IP (NSIP) address as the address of the new node. Then, log on to the other appliance and add a node that has the NSIP address of the first appliance. An algorithm determines which node becomes primary and which becomes secondary.

Note: The configuration utility provides an option that avoids having to log on to the second appliance.

The following figure shows a simple HA setup, in which both nodes are in same subnet.

Figure 1. Two NetScaler Appliances Connected in a High Availability Configuration



Adding a Remote Node

To add a remote NetScaler appliance as a node in a high availability setup, you specify a unique node ID and the appliance's NSIP. The maximum number of node IDs in an HA setup is 64. When you add an HA node, you must disable the HA monitor for each interface that is not connected or not being used for traffic. For CLI users, this is a separate procedure.

Note: To ensure that each node in the high availability configuration has the same settings, you should synchronize your SSL certificates, startup scripts, and other configuration files with those on the primary node.

To add a node by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

- add ha node <id> <IPAddress>
- show ha node

Example

```
> add ha node 10 203.0.113.32
Done
```

To disable an HA monitor by using the command line interface

At the command prompt, type:

- set interface <ifNum> [-haMonitor (**ON** | **OFF**)]
- show interface <ifNum>

Example

```
> set interface 1/3 -haMonitor OFF
 Done
```

To add a remote node by using the configuration utility

Navigate to System > High Availability and, on the Nodes tab, add a new remote node, or edit an existing node.

Disabling or Enabling a Node

Updated: 2013-08-28

You can disable or enable only a secondary node. When you disable a secondary node, it stops sending heartbeat messages to the primary node, and therefore the primary node can no longer check the status of the secondary. When you enable a node, the node takes part in the high availability configuration.

To disable or enable a node by using the command line interface

At the command prompt, type one of the following commands:

- set ha node -hastatus DISABLED
- set ha node -hastatus ENABLED

To disable or enable a node by using the configuration utility

1. Navigate to System > High Availability and, on the Nodes tab, open the node.
2. In the High Availability Status list, select ENABLED (Actively Participate in HA) or DISABLED (Do not participate in HA).

Removing a Node

Updated: 2013-08-28

If you remove a node, the nodes are no longer in high availability configuration.

To remove a node by using the command line interface

At the command prompt, type:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

rm ha node <id>

Example

```
> rm ha node 10
  Done
```

To remove a node by using the configuration utility

Navigate to System > High Availability and, on the Nodes tab, delete the node.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Communication Intervals

Oct 28, 2016

The hello interval is the interval at which the heartbeat messages are sent to the peer node. The dead interval is the time interval after which the peer node is marked DOWN if heartbeat packets are not received. The heartbeat messages are UDP packets sent to port 3003 of the other node in an HA pair.

To set the hello and dead intervals by using the command line interface

At the command prompt, type:

- set HA node [-helloInterval <msecs>] [-deadInterval <secs>]
- show HA node <id>

To set the hello and dead intervals by using the configuration utility

1. Navigate to System > High Availability and, on the Nodes tab, open the node.
2. Set the following parameters:
   - Hello Interval (msecs)
   - Dead Interval (secs)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Synchronization

Mar 19, 2012

Synchronization is a process of duplicating the configuration of the primary node on the secondary node. The purpose of synchronization is to ensure that there is no loss of configuration information between the primary and the secondary nodes, regardless of the number of failovers that occur. Synchronization uses port 3010.

Synchronization is triggered by either of the following circumstances:

- The secondary node in an HA setup comes up after a restart.
- The primary node becomes secondary after a failover.

Automatic synchronization is enabled by default. You can also force synchronization.

Disabling or Enabling Synchronization

Updated: 2013-08-28

Automatic HA synchronization is enabled by default on each node in an HA pair. You can enable or disable it on either node.

To disable or enable automatic synchronization by using the command line interface

At the command prompt, type:

- set HA node -haSync DISABLED
- set HA node -haSync ENABLED

To disable or enable synchronization by using the configuration utility

1. Navigate to System > High Availability and, on the Nodes tab, open the node.
2. Under HA Synchronization, clear or select the Secondary node will fetch the configuration from Primary option.

Forcing the Secondary Node to Synchronize with the Primary Node

Updated: 2013-08-28

In addition to automatic synchronization, the NetScaler supports forced synchronization. You can force the synchronization from either the primary or the secondary node. When you force synchronization from the secondary node, it starts synchronizing its configuration with the primary node.

However, if synchronization is already in progress, forced synchronization fails and the system displays a warning. Forced synchronization also fails in any of the following circumstances:

- You force synchronization on a standalone system.
- The secondary node is disabled.
- HA synchronization is disabled on the secondary node.

To force synchronization by using the command line interface

At the command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

force HA sync

To force synchronization by using the configuration utility

1. Navigate to System > High Availability.
2. On the Nodes tab, in the Action list, click Force Synchronization.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Synchronizing Configuration Files in a High Availability Setup

Apr 15, 2016

In a high availability setup, all configuration files are synchronised automatically from the primary node to the secondary node at an interval of one minute. Synchronizing configuration files can be performed manually by using the command line interface or the configuration utility at either the primary or the secondary node.

Files located on the secondary that are specific to the secondary (not present on the primary) are not deleted during the synchronization.

To synchronize files in a high availability setup by using the command line interface

At the command prompt, type:

sync HA files <mode>

Example

```
> sync HA files all
 Done
> sync HA files ssl
 Done
```

Parameter Descriptions (of the command listed in the CLI procedure)

sync ha files <mode>

mode

Specify one of the following modes of synchronization.

- **all** - Synchronize files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, and Application Firewall XML objects.
- **bookmarks** - Synchronize all Access Gateway bookmarks.
- **ssl** - Synchronize all certificates, keys, and CRLs for the SSL feature.
- **htmlinjection** - Synchronize all scripts configured for the HTML injection feature.
- **imports** - Synchronizeall XML objects (for example, WSDLs, schemas, error pages) configured for the application firewall.
- **misc** - Synchronize all license files and the rc.conf file.
- **all_plus_misc** - Synchronize files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, application firewall XML objects, licenses, and the rc.conf file.

To synchronize files in a high availability setup by using the configuration utility

Navigate to System > Diagnostics and, in the Utilities group, click Start HA files synchronization.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Command Propagation

Aug 28, 2013

In an HA setup, any command issued on the primary node propagates automatically to, and is executed on, the secondary before it is executed on the primary. If command propagation fails, or if command execution fails on the secondary, the primary node executes the command and logs an error. Command propagation uses port 3010.

In an HA pair configuration, command propagation is enabled by default on both the primary and secondary nodes. You can enable or disable command propagation on either node in an HA pair. If you disable command propagation on the primary node, commands are not propagated to the secondary node. If you disable command propagation on the secondary node, commands propagated from the primary are not executed on the secondary node.

Note: After reenabling propagation, remember to force synchronization.
If synchronization occurs while you are disabling propagation, any configuration-related changes that you make before the disabling of propagation takes effect are synchronized with the secondary node. This is also true for cases where propagation is disabled while synchronization is in progress.

To disable or enable command propagation by using the command line interface

At the command prompt, type:

- set HA node -haProp DISABLED
- set HA node -haProp ENABLED

To disable or enable command propagation by using the configuration utility

1. Navigate to System > High Availability and, on the Nodes tab, open the node.
2. Clear or select the Primary node will propagate configuration to the Secondary option.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Restricting High-Availability Synchronization Traffic to a VLAN

Dec 18, 2015

In a high availability (HA) deployment, traffic related to maintaining the HA configurationflows between the two HA nodes. This traffic is of the following types:

- Config synchronization
- Config propagation
- Connection mirroring
- Load balancing persistency config synchronization
- Persistent session synchronization
- Session state synchronization

Proper flow of this HA related traffic between the two nodes is critical for the functioning of the HA deployment. Typically, the HA related traffic is small in volume but can become very high during a failover. It becomes very high if stateful connection failover is enabled and the node that was primary before the failover was handling a large number of connections.

By default, the HA related traffic flows through the VLANs to which the NSIP address is bound. To accommodate a potential surge in this traffic, you can separate the HA related traffic from the management traffic and restrict its flow to a separate VLAN. This VLAN is called the HA SYNC VLAN.

Points to Consider before Configuring an HA SYNC VLAN

- The configuration of an HA SYNC VLAN is neither propagated nor synchronized. In other words, the HA SYNC VLAN is node specific and is configured independently on each nodes.
- HA SYNC VLAN configuration is removed when you clear the configuration in only FULL mode.
- HA MON must be set to OFF for interfaces that are part of the HA SYNC VLAN, to avoid a situation in which both nodes function as the primary node.
- Management interfaces (for example, 0/1 and 0/2) must not be part of the HA SYNC VLAN, so that HA related traffic does not flow through management interfaces.

To configure an HA SYNC VLAN on a NetScaler node, specify a configured VLAN with the HA SYNC VLAN parameter of the local node entity.

To configure an HA SYNC VLAN on a local node by using the command line

At the command prompt, type:

- **set node –hasyncvlan** <VLANID>
- **show node**

Parameter Description

Hasyncvlan (Sync VLAN)

VLAN on which HA related traffic is sent. This includes traffic for synchronization, propagation, connection mirroring, load balancing persistency, configuration synchonization, persistent session synchr

To configure an HA SYNC VLAN on a node by using the configuration utility

1. Navigate to **System** > **High Availability**.
2. Set the **Sync VLAN** parameter while modifying the local node.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Fail-Safe Mode

Aug 28, 2013

In an HA configuration, fail-safe mode ensures that one node is always primary when both nodes fail the health check. This is to ensure that when a node is only partially available, backup methods are enabled to handle traffic as best as possible. The HA fail-safe mode is configured independently on each node.

The following table shows some of the fail-safe cases. The NOT_UP state means that the node failed the health check yet it is partially available. The UP state means that the node passed the health check.

Table 1. Fail-Safe Mode Cases

Node A (Primary) Health State	Node B (Secondary) Health State	Default HA Behavior	Fail-Safe Enabled HA Behavior	Description
NOT_UP(failed last)	NOT_UP (failed first)	A (Secondary), B (Secondary)	A (Primary), B (Secondary)	If both nodes fail, one after the other, the node that was the last primary remains primary.
NOT_UP (failed first)	NOT_UP(failed last)	A (Secondary), B (Secondary)	A(Secondary), B(Primary)	If both nodes fail, one after the other, the node that was the last primary remains primary.
UP	UP	A (Primary), B (Secondary)	A (Primary), B (Secondary)	If both nodes pass the health check, no change in behavior with fail-safe enabled.
UP	NOT_UP	A(Primary), B(Secondary)	A (Primary), B (Secondary)	If only the secondary node fails, no change in behavior with fail-safe enabled.
NOT_UP	UP	A(Secondary), B(Primary)	A(Secondary), B(Primary)	If only the primary fails, no change in behavior with fail-safe enabled.
NOT_UP	UP (STAYSECONDARY)	A (Secondary), B (Secondary)	A (Primary), B (Secondary)	If the secondary is configured as STAYSECONDARY, the primary remains primary even if it fails.

To enable fail-safe mode by using the command line interface

At the command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

set HA node [-failSafe (**ON** | **OFF**)]

Example

```
set ha node -failsafe ON
```

To enable fail-safe mode by using the configuration utility

1. Navigate to System > High Availability and, on the Nodes tab, open the node.
2. Under Fail-Safe Mode, select the Maintain one Primary node even when both nodes are unhealthy option.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Virtual MAC Addresses

Mar 19, 2012

A Virtual MAC address (VMAC) is a floating entity shared by the primary and the secondary nodes in an HA setup.

In an HA setup, the primary node owns all of the floating IP addresses, such as the MIPs, SNIPs, and VIPs. The primary node responds to Address Resolution Protocol (ARP) requests for these IP addresses with its own MAC address. As a result, the ARP table of an external device (for example, an upstream router) is updated with the floating IP address and the primary node's MAC address.

When a failover occurs, the secondary node takes over as the new primary node. It then uses Gratuitous ARP (GARP) to advertise the floating IP addresses that it acquired from the primary. However, the MAC address that the new primary advertises is the MAC address of its own interface.

Some devices (notably a few routers) do not accept the GARP messages generated by the NetScaler appliance. As a result, some external devices retain the old IP to MAC mapping advertised by the old primary node. This can result in a site going down.

You can overcome this problem by configuring a VMAC on both nodes of an HA pair. Both nodes then possess identical MAC addresses. Therefore, when failover occurs, the MAC address of the secondary node remains unchanged, and the ARP tables on the external devices do not need to be updated.

To create a VMAC, you need to first create a Virtual Router ID (VRID) and bind it to an interface. (In an HA setup, you need to bind the VRID to the interfaces on both nodes.) Once the VRID is bound to an interface, the system generates a VMAC with the VRID as the last octet.

This section includes the following details:

- [Configuring IPv4 VMACs]
- [Configuring IPv6 VMAC6s]

Configuring IPv4 VMACs

When you create a IPv4 VMAC address and bind it to a interface, any IPv4 packet sent from the interface uses the VMAC address that is bound to the interface. If there is no IPv4 VMAC bound to an interface, the interface's physical MAC address is used.

The generic VMAC is of the form 00:00:5e:00:01:<VRID>. For example, if you create a VRID with a value of 60 and bind it to an interface, the resulting VMAC is 00:00:5e:00:01:3c, where 3c is the hex representation of the VRID. You can create 255 VRIDs with values from 1 to 255.

Creating or Modifying an IPv4 VMAC

Updated: 2013-08-28

You create an IPv4 virtual MAC by assigning it a virtual router ID. You can then you bind the VMAC to an interface. You cannot bind multiple VRIDs to the same interface. To verify the VMAC configuration, you should display and examine the VMACs and the interfaces bound to the VMACs.
To add a VMAC by using the command line interface

At the command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- add vrID <id>
- bind vrid <id> -ifnum <interface_name>
- show vrID

Example

```
> add vrID 100
 Done
> bind vrid 100 -ifnum 1/1 1/2 1/3
 Done
```

To unbind interfaces from a VMAC by using the command line interface

At the command prompt, type:

- unbind vrid <id> -ifnum <interface_name>
- show vrID

To configure a VMAC by using the configuration utility

Navigate to System > Network > VMAC and, on the VMAC tab, add a new VMAC, or edit an existing VMAC.

Removing an IPv4 VMAC

Updated: 2013-08-28

To remove an IPv4 virtual MAC, you delete its virtual router ID.
To remove an IPv4 VMAC by using the command line interface

At the command prompt, type:
rm vrid <id>
Example

```
rm vrid 100s
```

To remove an IPv4 VMAC by using the configuration utility

Navigate to System > Network > VMAC and, on the VMAC tab, delete the IPv4 VMAC.

Configuring IPv6 VMAC6s

The NetScaler supports VMAC6 for IPv6 packets. You can bind any interface to a VMAC6, even if an IPv4 VMAC is bound to the interface. Any IPv6 packet sent from the interface uses the VMAC6 bound to that interface. If there is no VMAC6 bound to an interface, an IPv6 packet uses the physical MAC.

Creating or Modifying a VMAC6

Updated: 2013-08-28

You create an IPv6 virtual MAC by assigning it an IPv6 virtual router ID. You can then you bind the VMAC to an interface. You cannot bind multiple IPv6 VRIDs to an interface. To verify the VMAC6 configuration, you should display and examine the VMAC6s and the interfaces bound to the VMAC6s.
To add a VMAC6 by using the command line interface

At the command prompt, type:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- add vrID6 <id>
- bind vrID6 <id> -ifnum <interface_name>
- show vrID6

Example

```
> add vrID6 100
Done
> bind vrID6 100 -ifnum 1/1 1/2 1/3
 Done
```

To unbind interfaces from a VMAC6 by using the command line interface

At the command prompt, type:

- unbind vrID6 <id> -ifnum <interface_name>
- show vrID6

To configure a VMAC6 by using the configuration utility

Navigate to System > Network > VMAC and, on the VMAC6 tab, add a new VMAC6, or edit an existing VMAC6.

Removing a VMAC6

Updated: 2013-08-28

To remove an IPv4 virtual MAC, you delete its virtual router ID.
To remove a VMAC6 by using the command line interface

At the command prompt, type:
rm vrid6 <id>
Example

```
rm vrid6 100s
```

To remove a VMAC6 by using the configuration utility

Navigate to System > Network > VMAC and, on the VMAC6 tab, delete the virtual router ID.

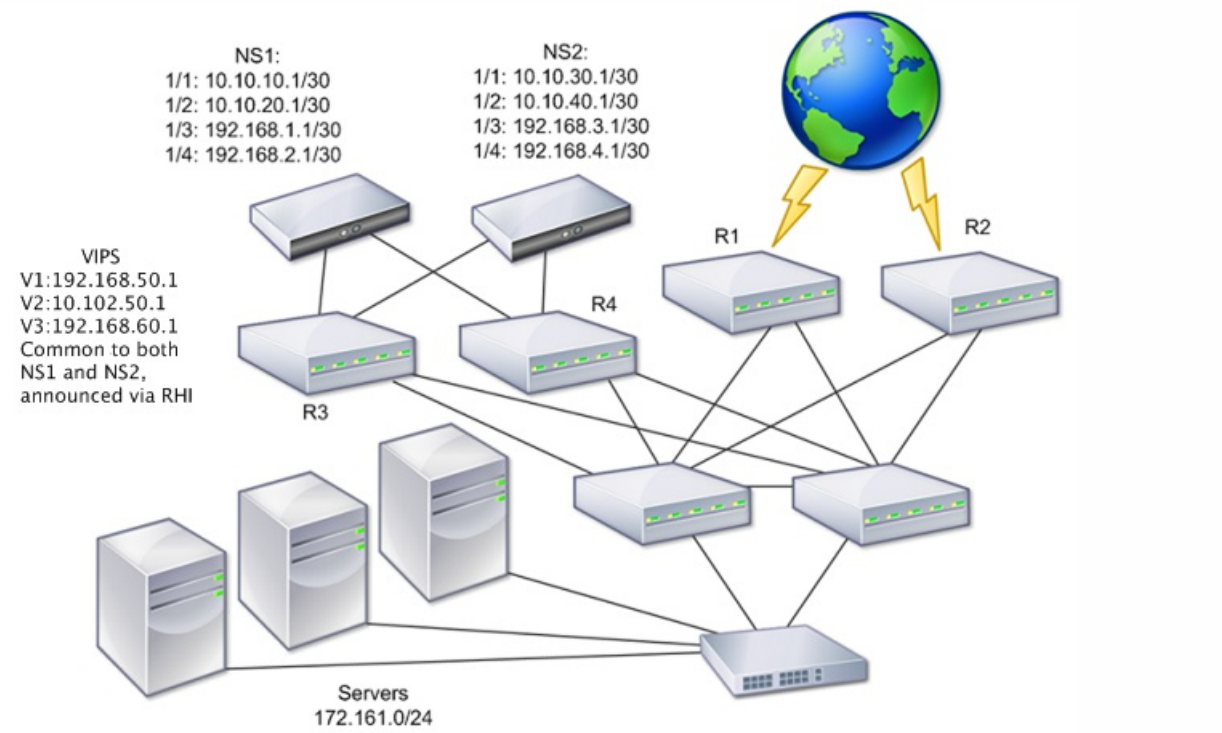
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring High Availability Nodes in Different Subnets

Sep 07, 2016

The following figure shows an HA deployment with the two systems located in different subnets:

Figure 1. High Availability over a Routed Network



In the figure, the systems NS1 and NS2 are connected to two separate routers, R3 and R4, on two different subnets. The NetScaler appliances exchange heartbeat packets through the routers. This configuration could be expanded to accommodate deployments involving any number of interfaces.

Note: If you use static routing on your network, you must add static routes between all the systems to ensure that heartbeat packets are sent and received successfully. (If you use dynamic routing on your systems, static routes are unnecessary.)

If the nodes in an HA pair reside on two separate networks, the primary and secondary node must have independent network configurations. This means that nodes on different networks cannot share entities such as SNIPs, VLANs, and routes. This type of configuration, where the nodes in an HA pair have different configurable parameters, is known as Independent Network Configuration (INC) or Symmetric Network Configuration (SNC).

The following table summarizes the configurable entities and options for an INC, and shows how they must be set on each node.

Table 1. Behavior of NetScaler Entities and Options in an Independent Network Configuration

NetScaler entities	Options

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler entities	Options
IPs (NSIP/SNIPs)	Node-specific. Active only on that node.
VIPs	Floating.
VLANs	Node-specific. Active only on that node.
Routes	Node-specific. Active only on that node. Link load balancing routes are floating.
ACLs	Floating (Common). Active on both nodes.
Dynamic routing	Node-specific. Active only on that node. The secondary node should also run the routing protocols and peer with upstream routers.
L2 mode	Floating (Common). Active on both nodes.
L3 mode	Floating (Common). Active on both nodes.
Reverse NAT (RNAT)	Node-specific. RNAT with VIP, because NATIP is floating.

As in configuring HA nodes in the same subnet, to configure HA nodes in different subnets, you log on to each of the two NetScaler appliances and add a remote node representing the other appliance.

Adding a Remote Node

When two nodes of an HA pair reside on different subnets, each node must have a different network configuration. Therefore, to configure two independent systems to function as an HA pair, you must specify INC mode during the configuration process.

When you add an HA node, you must disable the HA monitor for each interface that is not connected or not being used for traffic. For CLI users, this is a separate procedure.

To add a node by using the command line interface

At the command prompt, type:

- add ha node <id> <IPAddress> -inc ENABLED
- show ha node

Example

```
> add ha node 3 10.102.29.170 -inc ENABLED
 Done
> add ha node 3 1000:0000:0000:0000:0005:0600:700a:888b
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done
```

To disable an HA monitor by using the command line interface

At the command prompt, type:

- set interface <ifNum> [-haMonitor (**ON** | **OFF**)]
- show interface <ifNum>

Example

```
> set interface 1/3 -haMonitor OFF
Done
```

To add a remote node by using the configuration utility

1. Navigate to System > High Availability and, on the Nodes tab, add a new remote node.
2. Make sure to select the Turn off HA monitor on interfaces/channels that are down and Turn on INC (Independent Network Configuration) mode on self mode options.

Removing a Node

Updated: 2013-08-28

If you remove a node, the nodes are no longer in high availability configuration.

To remove a node by using the command line interface

At the command prompt, type:

rm ha node <id>

Example

```
> rm ha node 2
Done
```

To remove a node by using the configuration utility

Navigate to System > High Availability and, on the Nodes tab, delete the node.

Note: You can use the Network Visualizer to view the NetScaler appliances that are configured as a high availability (HA) pair and perform high availability configuration tasks. For more information, see "."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Route Monitors

Sep 06, 2013

You can use route monitors to make the HA state dependent on the internal routing table, whether or not the table contains any dynamically learned or static routes. In an HA configuration, a route monitor on each node watches the internal routing table to make sure that a route entry for reaching a particular network is always present. If the route entry is not present, the state of the route monitor changes to DOWN.

When a NetScaler appliance has only static routes for reaching a network, and you want to create a route monitor for the network, you must enable monitored static routes (MSR) for the static routes. MSR removes unreachable static routes from the internal routing table. If MSR is disabled on static routes, an unreachable static route can remain in the internal routing table, defeating the purpose of having the route monitor.

Route Monitors are supported both in non-INC and INC mode.

Route Monitors in HA in non-INC mode	**Route Monitors in HA in INC mode**
Route monitors are propagated by nodes and exchanged during synchronization.	Route monitors are neither propagated by nodes nor exchanged during synchronization.
Route monitors are active only in the current primary node.	Route monitors are active on both the primary and the secondary node.
The NetScaler appliance always displays the state of a route monitor as UP irrespective of the whether the route entry is present or not in the internal routing table.	The NetScaler appliance displays the state of the route monitor as DOWN if the corresponding route entry is not present in the internal routing table.
A route monitor starts monitoring its route after 180 seconds in the following cases [This is done to allow dynamic routes to get learnt, which may take 180 secs]: • reboot • failover • set route6 command for v6 routes • set route msr enable/disable command for v4 routes. • adding a new route monitor	-

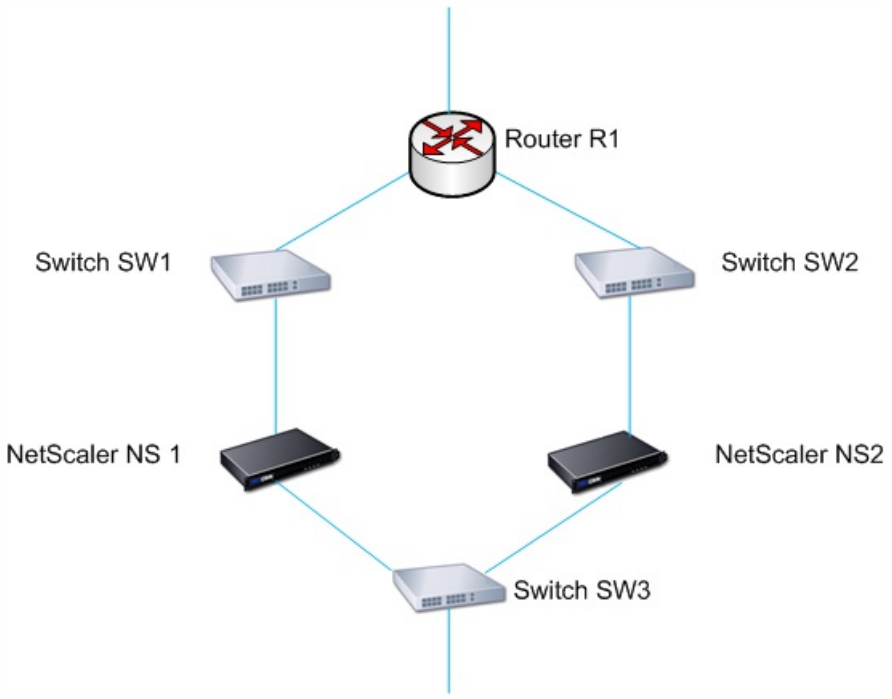
Route monitors are useful in a non-INC mode HA configuration where you want the non-reachability of a gateway from a primary node to be one of the conditions for HA failover.

Consider an example of a non-Inc mode HA setup in a two-arm topology that has NetScaler appliances NS1 and NS2 in the same subnet, with router R1 and switches SW1, SW2, and SW3.

Because R1 is the only router in this setup, you want the HA setup to failover whenever R1 is not reachable from the current primary node. You can configure a route monitor (say, RM1 and RM2, respectively) on each of the nodes to monitor the reachability of R1 from that node.

Figure 1.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



With NS1 as the current primary node, the execution flow is as follows:

1. Route monitor RM1 on NS1 monitors NS1's internal routing table for the presence of a route entry for router R1. NS1 and NS2 exchange heartbeat messages through switch SW1 or SW3 at regular intervals.
2. If switch SW1 goes down, the routing protocol on NS1 detects that R1 is not reachable and therefore removes the route entry for R1 from the internal routing table. NS1 and NS2 exchanges heartbeat messages through switch SW3 at regular intervals.
3. Detecting that the route entry for R1 is not present in the internal routing table, RM1 initiates a failover. If route to R1 is down from both NS1 and NS2, failover happens every 180 seconds till one of the appliances is able to reach R1 and restore the connectivity.

Adding a Route Monitor to a High Availability Node

A single procedure creates a route monitor and binds it to an HA node.

To add a route monitor by using the command line interface

At the command prompt, type:

- bind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])
- show HA node

Example

```
> bind HA node 0 -routeMonitor 10.102.71.0 255.255.255.0
 Done
> bind HA node 0 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
 Done
```

To add a route monitor by using the configuration utility

Navigate to System > High Availability and, on the Route Monitors tab, click Configure.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Removing Route Monitors

Updated: 2013-08-28

To remove a route monitor by using the command line interface

At the command prompt, type:

- unbind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])
- show ha node

Example

```
unbind HA node 3 -routeMonitor 10.102.71.0 255.255.255.0
unbind HA node 3 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
```

To remove a route monitor by using the configuration utility

Navigate to System > High Availability and, on the Route Monitors tab, delete the route monitor.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Limiting Failovers Caused by Route Monitors in non-INC mode

May 16, 2016

In an HA configuration in non-INC mode, if route monitors fail on both nodes, failover happens every 180 seconds until one of the nodes is able to reach all of the routes monitored by the respective route monitors.

However, for a node, you can limit the number of failovers for a given interval by setting the Maximum Number of Flips and Maximum Flip Time parameters on the nodes. When either limit is reached, no more failovers occur, and the node is assigned as primary (but node state as NOT UP) even if any route monitor fails on that node. This combination of HA state as Primary and Node state as NOT UP is called Stick Primary state.

If the node is then able to reach all of the monitored routes, the next monitor failure triggers resetting of the Maximum Number of Flips and Maximum Flip Time parameters on the node and starting the time specified in the Maximum Flip Time parameter.

These parameters are set independently on each node and therefore are neither propagated nor synchronized.

Parameters for limiting the number of failovers

Maximum Number of Flips (maxFlips)

Maximum number of failovers allowed, within the Maximum Flip Time interval, for the node in HA in non INC mode, if the failovers are caused by route-monitor failure.

Maximum Flip Time (maxFlipTime)

Amount of time, in seconds, during which failovers resulting from route-monitor failure are allowed for the node in HA in non INC mode.

To limit the number of failovers by using the command line interface

At the command prompt, type:

- **set HA node** [-**maxFlips** < positive_integer>] [-**maxFlipTime** <positive_integer>]
- **show HA node** [< id>]

To limit the number of failovers by using the configuration utility

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the local node.
2. Set the following parameters:

- Maximum Number of Flips
- Maximum Flip Time

Sample Configuration COPY

```
> set ha node -maxFlips 30 -maxFlipTime 60
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done

> sh ha node

1) Node ID: 0

IP: 10.102.169.82 (NS)

Node State: UP

Master State: Primary

Fail-Safe Mode: OFF

INC State: DISABLED

Sync State: ENABLED

Propagation: ENABLED

Enabled Interfaces : 1/1

Disabled Interfaces : None

HA MON ON Interfaces : 1/1

Interfaces on which heartbeats are not seen :None

Interfaces causing Partial Failure:None

SSL Card Status: NOT PRESENT

Hello Interval: 200 msecs

Dead Interval: 3 secs

Node in this Master State for: 0:4:24:1 (days:hrs:min:sec)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
2) Node ID: 1

IP: 10.102.169.81

Node State: UP

Master State: Secondary

Fail-Safe Mode: OFF

INC State: DISABLED

Sync State: SUCCESS

Propagation: ENABLED

Enabled Interfaces : 1/1

Disabled Interfaces : None

HA MON ON Interfaces : 1/1

Interfaces on which heartbeats are not seen : None

Interfaces causing Partial Failure: None

SSL Card Status: NOT PRESENT



Local node information:

Configured/Completed Flips: 30/0

Configured Flip Time: 60

Critical Interfaces: 1/1
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Done

SNMP Alarm for Sticky Primary State

Enable HA-STICKY-PRIMARY SNMP alarm in a node of a high availability set up if you want to be alerted of the node becoming sticky primary. When the node becomes sticky primary, it alerts by generating a trap message (stickyPrimary (1.3.6.1.4.1.5951.1.1.0.138)) and sends it to all the configured SNMP trap destinations. For more information about configuring SNMP alarms and trap destinations, see Configuring the NetScaler to Generate SNMPv1 and SNMPv2 Traps.

Frequently Asked Questions

Consider an example of a high availability setup of two NetScaler appliances NS-1 and NS-2 in non-INC mode. Maximum numbers of flips and maximum flip time in both the nodes have been set with the same values.

The following table lists the settings used in this example:

Entity	Detail
IP address of NS-1	10.102.173.211
IP address of NS-2	10.102.173.212
Maximum number of flips	2
Maximum flip time	200

The following table lists some FAQs and answers about maximum number of flips and maximum flip time settings:

Question	Answer
What must be the next plan of action after one of the node become sticky primary?	Rectify the routes, which are being monitored. After the maximum flip time is elapsed, any route monitor failure triggers resetting of the Maximum number of flips and maximum flip time, then starting the time specified in maximum flip time. The following example shows that NS-1 (10.102.173.211) becomes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	sticky primary. > show ha node 1) Node ID: 0 IP: 10.102.173.211 Node State: NOT UP Master State: Primary . . 2) Node ID: 1 IP: 10.102.173.212 Node State: UP Master State: Secondary . . Local node information: Route Monitor - Network: 10.102.173.216 Netmask: 255.255.255.255 State: DOWN Critical Interfaces: 1/1 1/2 **Configured/Completed Flips: 2/2** **Configured/Remaining Flip Time: 200/0** Done
What happens if a node recovers from sticky primary state before the maximum flip time is elapsed?	Nothing happens. Maximum number of flips and maximum flip time are not reset.
What happens if a node recovers from sticky primary state after the maximum flip time is elapsed?	Nothing happens. Maximum number of flips and maximum flip time are not reset.
What happens if a node recovers from sticky primary state and then the route that is being monitored goes down again before the maximum flip time is elapsed?	The node will again become sticky primary without a failover. Maximum number of flips and maximum flip time are not reset.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following example shows that NS-1 (10.102.173.211) recovers from sticky primary state. NS-1 again becomes sticky primary when the route that is being monitored goes down again before the maximum flip time is elapsed.

```
> show ha node

1)    Node ID:     0

      IP:  10.102.173.211

      Node State: UP

      Master State: Primary

      .

      .

2)    Node ID:     1

      IP:  10.102.173.212

      Node State: UP

      Master State: Secondary

      .

      .

Local node information:

Route Monitor -  Network: 10.102.173.216   Netmask:
255.255.255.255   State: UP

Critical Interfaces: 1/1 1/2
Configured/Completed Flips: 2/2
Configured/Remaining Flip Time: 200/113

Done

> show ha node

1)    Node ID:     0

      IP:  10.102.173.211

      Node State: NOT UP

      Master State: Primary

      .

      .
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

	2) Node ID: 1 IP: 10.102.173.212 Node State: UP Master State: Secondary . . Local node information: Route Monitor - Network: 10.102.173.216 Netmask: 255.255.255.255 State: DOWN Critical Interfaces: 1/1 1/2 **Configured/Completed Flips: 2/2** **Configured/Remaining Flip Time: 200/83** Done
What happens if a node recovers from sticky primary state and then the route that is being monitored goes down again after the maximum flip time is elapsed?	Maximum number of flips and maximum flip time are reset to the configured values. Then, Maximum flip time starts. Also, failover happens until either of the following condition is achieved: ● one of the nodes is able to reach all of the routes monitored by the respective route monitors. ● number of failover equals the maximum number of flips The following example shows that NS-1 (10.102.173.211) recovers from sticky primary state. When the route (10.102.173.216) that is being monitored goes down again before the maximum flip time is elapsed, maximum number of flips and maximum flip time are reset, and maximum flip time starts. The second output of show ha node shows NS-1 becomes secondary after a failover. > show ha node 1) Node ID: 0 IP: 10.102.173.211 Node State: UP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Master State: Primary

.

.

2) Node ID: 1

IP: 10.102.173.212

Node State: UP

Master State: Secondary

.

.

Local node information:

Route Monitor - Network: 10.102.173.216 Netmask: 255.255.255.255 State: UP

Critical Interfaces: 1/1 1/2

Configured/Completed Flips: 2/2

Configured/Remaining Flip Time: 200/0

Done

> show ha node

1) Node ID: 0

IP: 10.102.173.211

Node State: UP

Master State: Primary

.

.

2) Node ID: 1

IP: 10.102.173.212

Node State: UP

Master State: Secondary

.

.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	Local node information: Route Monitor - Network: 10.102.173.216 Netmask: 255.255.255.255 State: UP Critical Interfaces: 1/1 1/2 **Configured/Completed Flips: 2/1** **Configured/Remaining Flip Time: 200/196** Done
What happens when maximum number of flips and maximum flip time are unset?	After the maximum number of flips and maximum flip time, the setup falls to the failover cycle of180 seconds until the route monitor state become UP.
What happens when maximum flip time is over but not the maximum number of flips and there is a route down event?	The setup goes to continuous flip cycle. If maximum flip time is over before the maximum flips are completed, both these parameters are reset to the configured values. As a result, the flip cycle continues forever. The maximum flip time must be configured in such a way that the maximum number of flips can be completed in this configured time.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Failover Interface Set

Mar 19, 2012

A Failover Interface Set (FIS) is a logical group of interfaces. In an HA configuration, using a FIS is a way to prevent failover by grouping interfaces so that, when one interface fails, other functioning interfaces are still available. A FIS can also be configured for the nodes of a NetScaler cluster.

HA MON interfaces that are not bound to an FIS are known as critical interfaces (CI) because if any of them fails, failover is triggered.

Note: An FIS does not create an active and standby configuration. It also does not prevent bridging loops when connecting to links to the same VLAN.

> **Note**
>
> An FIS does not create an active and standby Interfaces or channels. It also does not prevent bridging loops when connecting to links to the same VLAN

Creating or Modifying an FIS

To add an FIS and bind interfaces to it by using the command line interface

At the command prompt, type:

- add fis <name>
- bind fis <name> <ifnum> ...
- show fis <name>

Example

```
> add fis fis1
 Done
> bind fis fis1 1/3 1/5
 Done
```

An unbound interface becomes a critical interface (CI) if it is enabled and HA MON is on.

To unbind an interface from an FIS by using the command line interface

At the command prompt, type:

- unbind fis <name> <ifnum> ...
- show fis <name>

Example

```
> unbind fis fis1 1/3
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure an FIS by using the configuration utility

Navigate to System > High Availability and, on the Failover Interface Set tab, add a new FIS, or edit an existing FIS.

Removing an FIS

Updated: 2013-08-28

When the FIS is removed, its interfaces are marked as critical interfaces.

To remove an FIS by using the command line interface

At the command prompt, type:

rm fis <name>

Example

```
> rm fis fis1
 Done
```

To remove an FIS by using the configuration utility

Navigate to System > High Availability and, on the Failover Interface Set tab, delete the FIS.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding the Causes of Failover

Sep 30, 2013

The following events can cause failover in an HA configuration:

1. If the secondary node does not receive a heartbeat packet from the primary for a period of time that exceeds the dead interval set on the secondary. (See Note: 1.)
2. The primary node experiences a hardware failure of its SSL card.
3. The primary node does not receive any heartbeat packets on its network interfaces for three seconds.
4. On the primary node, a network interface that is not part of a Failover Interface Set (FIS) or a Link Aggregation (LA) channel and has the HA Monitor (HAMON) enabled, fails. (See Note: 2.)
5. On the primary node, all interfaces in an FIS fail. (See Note: 2.)
6. On the primary node, an LA channel with HAMON enabled fails. (See Note: 2.)
7. On the primary node, all interfaces fail (see Note: 2). In this case, failover occurs regardless of the HAMON configuration.
8. On the primary node, all interfaces are manually disabled. In this case, failover occurs regardless of the HAMON configuration.
9. You force a failover by issuing the force failover command on either node.
10. A route monitor that is bound to the primary node goes DOWN.

    Note: 1. For more information about setting the dead interval, see [Configuring the Communication Intervals]. Possible causes for a node not receiving heartbeat packets from a peer node include:

    - A network configuration problem prevents heartbeats from traversing the network between the HA nodes.
    - The peer node experiences a hardware or software failure that causes it to freeze (hang), reboot, or otherwise stop processing and forwarding heartbeat packets.

    Note: 2. In this case, fail means that the interface was enabled but goes to the DOWN state, as can be seen from the show interface command or from the configuration utility. Possible causes for an enabled interface to be in the DOWN state are LINK DOWN and TXSTALL.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Forcing a Node to Fail Over

Mar 19, 2012

You might want to force a failover if, for example, you need to replace or upgrade the primary node. You can force failover from either the primary or the secondary node. A forced failover is not propagated or synchronized. To view the synchronization status after a forced failover, you can view the status of the node.

A forced failover fails in any of the following circumstances:

- You force failover on a standalone system.
- The secondary node is disabled.
- The secondary node is configured to remain secondary.

The NetScaler appliance displays a warning message if it detects a potential issue when you run the force failover command. The message includes the information that triggered the warning, and requests confirmation before proceeding.

You can force a failover on a primary node, secondary node, and when nodes are in listen mode.

- **Forcing Failover on the Primary Node**.
  If you force failover on the primary node, the primary becomes the secondary and the secondary becomes the primary. Forced failover is possible only when the primary node can determine that the secondary node is UP.

  If the secondary node is DOWN, the force failover command returns the following error message: "Operation not possible due to invalid peer state. Rectify and retry."

  If the secondary system is in the claiming state or inactive, it returns the following error message: "Operation not possible now. Please wait for system to stabilize before retrying."

- **Forcing Failover on the Secondary Node**.
  If you run the force failover command from the secondary node, the secondary node becomes primary and the primary node becomes secondary. A force failover can occur only if the secondary node's health is good and it is not configured to stay secondary.

  If the secondary node cannot become the primary node, or if secondary node was configured to stay secondary (using the STAYSECONDARY option), the node displays the following error message: "Operation not possible as my state is invalid. View the node for more information."

- **Forcing Failover When Nodes Are in Listen Mode**.
  When the two nodes of an HA pair are running different versions of the system software, the node running the higher version switches to the listen mode. In this mode, neither command propagation nor synchronization works.

  Before upgrading the system software on both nodes, you should test the new version on one of the nodes. To do this, you need to force a failover on the system that has already been upgraded. The upgraded system then takes over as the primary node, but neither command propagation or synchronization occurs. Also, all connections need to be re-established.

To force failover on a node by using the command line interface

At the command prompt, type:

force HA failover

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To force failover on a node by using the configuration utility

Navigate to System > High Availability and, on the Nodes tab, select the node, in the Action list, select Force Failover.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Forcing the Secondary Node to Stay Secondary

Aug 28, 2013

In an HA setup, the secondary node can be forced to stay secondary regardless of the state of the primary node.

For example, suppose the primary node needs to be upgraded and the process will take a few seconds. During the upgrade, the primary node may go down for a few seconds, but you do not want the secondary node to take over; you want it to remain the secondary node even if it detects a failure in the primary node.

When you force the secondary node to stay secondary, it will remain secondary even if the primary node goes down. Also, when you force the status of a node in an HA pair to stay secondary, it does not participate in HA state machine transitions. The status of the node is displayed as STAYSECONDARY.

Forcing the node to stay secondary works on both standalone and secondary nodes. On a standalone node, you must use this option before you can add a node to create an HA pair. When you add the new node, the existing node continues to function as the primary node, and the new node becomes the secondary node.

Note: When you force a system to remain secondary, the forcing process is not propagated or synchronized. It affects only the node on which you run the command.

To force the secondary node to stay secondary by using the command line interface

At the command prompt, type:

```
set ha node -hastatus STAYSECONDARY
```

To force the secondary node to stay secondary by using the configuration utility

Navigate to System > High Availability, on the Nodes tab, open the local node, and select STAY SECONDARY.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Forcing the Primary Node to Stay Primary

Aug 28, 2013

In an HA setup, you can force the primary node to remain primary even after a failover. You can enable this option either on a primary node in an HA pair or on a standalone system.

On a standalone system, you must run this command before you can add a node to create an HA pair. When you add the new node, it becomes the primary node. The existing node stops processing traffic and becomes the secondary node in the HA pair.

To force the primary node to stay primary by using the command line interface

At the command prompt, type:

```
set ha node -hastatus STAYPRIMARY
```

To force the primary node to stay primary by using the configuration utility

Navigate to System > High Availability, on the Nodes tab, open the local node, and select STAY PRIMARY.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding the High Availability Health Check Computation

Mar 19, 2012

The following table summarizes the factors examined in a health check computation:

- State of the CIs
- State of the FISs
- State of the route monitors

The following table summarizes the health check computation.

Table 1. High Availability Health Check Computation

FIS	CI	Route monitor	Condition
N	Y	N	If the system has any CIs, all of those CIs must be UP.
Y	Y	N	If the system has any FISs, all of those FISs must be UP.
Y	Y	Y	If the system has any route monitors configured, all monitored routes must be present in the FIS.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

High Availability FAQs

Jul 16, 2013

What are the various ports used to exchange the HA-related information between the nodes in an HA configuration?

In an HA configuration, both nodes use the following ports to exchange HA related information:

- UDP Port 3003, to exchange heartbeat packets.
- Port 3010, for synchronization and command propagation.

What are the conditions that trigger synchronization?

Synchronization is triggered by any of the following conditions:

- The incarnation number of the primary node, received by the secondary, does not match that of the secondary node. Note: Both nodes in an HA configuration maintain a counter called *incarnation number*, which counts the number of configurations in the node's configuration file. Each node sends its incarnation number to each other node in the heartbeat messages. The incarnation number is not incremented for the following commands:
  1. All HA configuration related commands. For example, add ha node, set ha node, and bind ha node.
  2. All Interface related commands. For example, set interface and unset interface.
  3. All channel-related commands. For example, add channel, set channel, and bind channel.
- The secondary node comes up after a restart.
- The primary node becomes secondary after a failover.

What configurations are not synced or propagated in an HA configuration in INC or non-INC mode?

The following commands are neither propagated nor synced to the secondary node:

- All node specific HA configuration commands. For example, add ha node, set ha node, and bind ha node.
- All Interface related configuration commands. For example, set interface and unset interface.
- All channel related configuration commands. For example, add channel, set channel, and bind channel.

What configurations are not synced nor propagated in an HA configuration in INC mode?

The following configurations are not synced or propagated. Each node has its own.

- MIPs
- SNIPs
- VLANs
- Routes (except LLB routes)
- Route monitors
- RNAT rules (except any RNAT rule with VIP as the NAT IP)
- Dynamic routing configurations.

Does a configuration added to the secondary node get synchronized on the primary?

No, a configuration added to the secondary node is not synchronized to the primary.

What could be the reason for both nodes claiming to be the primary in an HA configuration?

The most likely reason is that the primary and secondary nodes are both healthy but the secondary does not receive the heartbeat packets from the primary. The problem could be with the network between the nodes.

Does an HA configuration run into any issues if you deploy the two nodes with different system clock settings?

Different system-clock settings on the two nodes can cause the following issues:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- The time stamps in the log file entries do not match. This situation makes it difficult to analyze the log entries for any issues.
- After a failover, you might have problems with any type of cookie based persistence for load balancing. A significant difference between the times can cause a cookie to expire sooner than expected, resulting in termination of the persistence session.
- Similar considerations apply to any time related decisions on the nodes.

What are the conditions for failure of the *force HA sync* command?

Forced synchronization fails in any of the following circumstances:

- You force synchronization when synchronization is already in progress.
- You force synchronization on a standalone NetScaler appliance.
- The secondary node is disabled.
- HA synchronization is disabled on the current secondary node.
- HA propagation is disabled on the current primary node and you force synchronization from the primary.

What are the conditions for failure of the *sync HA files* command?

Synchronizing configuration files fail in either of the following circumstances:

- On a standalone system.
- With the secondary node disabled.

In an HA configuration, if the secondary node takes over as the primary, does it switch back to secondary status if the original primary comes back online?

No. After the secondary node takes over as the primary, it remains as primary even if the original primary node comes back online again. To interchange the primary and secondary status of the nodes, run the *force failover* command.

What are the conditions for failure of the *force failover* command?

A forced failover fails in any of the following circumstances:

- You force failover on a standalone system.
- The secondary node is disabled.
- The secondary node is configured to remain secondary.
- The primary node is configured to remain primary.
- The state of the peer node is unknown.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting High Availability Issues

Jul 29, 2013

The most common high availability issues involve the high availability feature not working at all, or working only intermittently. Following are common high availability issues, and probable causes and resolutions.

- **Issue**
  The inability of the NetScaler appliances to pair the NetScaler appliances in a high availability setup.
  - **Cause**
    Network connectivity

    Resolution

    Verify that both the appliances are connected to the switch and the interfaces are enabled.

  - **Cause**
    Mismatch in the Password for the default Administrator account

    Resolution

    Verify that the password on both the appliances is the same.

  - **Cause**
    IP conflict

    Resolution

    Verify that both the appliances have unique NetScaler IP (NSIP) address. The appliances should not have the same NSIP address.

  - **Cause**
    Node ID mismatch

    Resolution

    Verify that the Node ID Configuration on both the appliances is unique. The appliances should not have the same Node ID configuration. Additionally, you must assign value for a Node ID between 1 and 64.

  - **Cause**
    Mismatch in the password of the RPC node

    Resolution

    Verify that both the nodes have the same RPC node password.

  - **Cause**
    An administrator has disabled the remote node

    Resolution

    Enable the remote node.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Cause**
  The Firewall application has blocked the heartbeat packets

  Resolution

  Verify that the UDP port 3003 is allowed.

- **Issue**
  Both the appliances claim to be the primary appliance.
  - **Cause**
    Missing heartbeat packets between the appliances

    Resolution

    Verify that the UDP port 3003 is not blocked for communication between the appliances.

- **Issue**
  The NetScaler appliance is not able to synchronize the configuration.
  - **Cause**
    A Firewall application is blocking the required port.

    Resolution

    Verify that the UDP port 3010 (or UDP port 3008 with secure synchronization) is not blocked for communication between the appliances.

  - **Cause**
    An administrator has disabled synchronization.

    Resolution

    Enable synchronization on the appliance that has the issue.

  - **Cause**
    Different NetScaler releases or builds are installed on appliances.

    Resolution

    Upgrade the appliances to the same NetScaler release or build.

- **Issue**
  Command propagation fails between the appliances.
  - **Cause**
    A Firewall application is blocking the port.

    Resolution

    Verify that the UDP port 3011 (or UDP port 3009 with secure propagation) is not blocked for communication between the appliances.

  - **Cause**
    An administrator has disabled command propagation.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Resolution

Enable command propagation on the appliance that has the issue.

- **Cause**
  Different NetScaler releases or builds are installed on appliances.

  Resolution

  Upgrade the appliances to the same NetScaler release or build.

- **Issue**
  The NetScaler appliances in the high availability pair are unable to run the force failover process.
  - **Cause**
    The Secondary node is disabled.

    Resolution

    Enable the secondary node.

  - **Cause**
    The Secondary node is configured to stay secondary.

    Resolution

    Set the secondary high availability status of the secondary node to Enable from Stay Secondary.

- **Issue**
  The secondary appliance does not receive any traffic after the failover process.
  - **Cause**
    The upstream router does not understand GARP messages of NetScaler appliance.

    Resolution

    Configure Virtual MAC (VMAC) address on the secondary appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Networking

May 13, 2016

The following topics provide a conceptual reference and instructions for configuring the various networking components on the NetScaler appliance.

IP Addressing	Learn the various types of NetScaler-owned IP addresses and how to create, customize, and remove them.
Interfaces	Configure some of the basic network configurations that must be done to get started.
Access Control Lists (ACLs)	Configure the different types of Access Control Lists and how to create, customize, and remove them.
IP Routing	Learn and configure the routing functionality of the NetScaler appliance, both static and dynamic.
Internet Protocol version 6 (IPv6)	Learn how the NetScaler appliance supports IPv6.
Traffic Domains	Learn and configure traffic domains to segment network traffic for different applications.
VXLAN	Learn and configure VXLANs to meet the scalability needs in your datacenter.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

IP Addressing

Aug 28, 2013

Before you can configure the NetScaler appliance, you must assign the NetScaler IP Address (NSIP), also known as the Management IP address. You can also create other NetScaler-owned IP addresses for abstracting servers and establishing connections with the servers. In this type of configuration, the appliance serves as a proxy for the abstracted servers. You can also proxy connections by using network address translations (INAT and RNAT). When proxying connections, the appliance can behave either as a bridging (Layer 2) device or as a packet forwarding (Layer 3) device. To make packet forwarding more efficient, you can configure static ARP entries. For IPv6, you can configure neighbor discovery (ND).

This document includes the following information:

- Configuring NetScaler-Owned IP Addresses
- How the NetScaler Proxies Connections
- Enabling Use Source IP Mode
- Configuring Network Address Translation
- Configuring Static ARP
- Setting the Timeout for Dynamic ARP Entries
- Configuring Neighbor Discovery
- Configuring IP Tunnels

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring NetScaler-Owned IP Addresses

Mar 20, 2012

The NetScaler-owned IP Addresses—NetScaler IP Address (NSIP), Virtual IP Addresses (VIPs), Subnet IP Addresses (SNIPs), Mapped IP Addresses (MIPs), and Global Server Load Balancing Site IP Addresses (GSLBIPs)—exist only on the NetScaler appliance. The NSIP uniquely identifies the NetScaler on your network, and it provides access to the appliance. A VIP is a public IP address to which a client sends requests. The NetScaler terminates the client connection at the VIP and initiates a connection with a server. This new connection uses a SNIP or a MIP as the source IP address for packets forwarded to the server. If you have multiple data centers that are geographically distributed, each data center can be identified by a unique GSLBIP.

You can configure some NetScaler-owned IP addresses to provide access for management applications.

This document includes the following information:

- Configuring the NetScaler IP Address (NSIP)
- Configuring and Managing Virtual IP (VIP) Addresses
- Configuring ARP response Suppression for Virtual IP addresses (VIPs)
- Configuring Subnet IP Addresses (SNIPs)
- Configuring Mapped IP Addresses (MIPs)
- Configuring GSLB Site IP Addresses (GSLBIP)
- Removing a NetScaler-Owned IP Address
- Configuring Application Access Controls

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NetScaler IP Address (NSIP)

Aug 28, 2013

The NetScaler IP (NSIP) address is the IP address at which you access the NetScaler for management purposes. The NetScaler can have only one NSIP, which is also called the Management IP address. You must add this IP address when you configure the NetScaler for the first time. If you modify this address, you must reboot the NetScaler. You cannot remove an NSIP address. For security reasons, NSIP should be a non-routable IP address on your organization's LAN.

Note: Configuring the NetScaler IP address is mandatory.

To create the NetScaler IP address by using the command line interface

At the command prompt, type:

- set ns config [-IPAddress <ip_addr> -netmask <netmask>]
- show ns config

Example

```
> set ns config -ipaddress 10.102.29.170 -netmask 255.255.255.0
 Done
```

To configure the NetScaler IP address by using the configuration utility

1. In the navigation pane, click System.
2. On the System Information tab, click Setup Wizard.
3. In the Setup Wizard dialog box, click Next.
4. Under System Configuration, set the following parameters:
   - IP Address
   - Netmask
5. Follow the instructions in the Setup Wizard to complete the configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring and Managing Virtual IP (VIP) Addresses

Nov 23, 2015

Configuration of a virtual server IP (VIP) address is not mandatory during initial configuration of the NetScaler ADC. When you configure load balancing, you assign VIP addresses to virtual servers.

For more information about configuring a load balancing setup, see "Load Balancing."

In some situations, you need to customize VIP attributes or enable or disable a VIP address. A VIP address is usually associated with a virtual server, and some of the VIP attributes are customized to meet the requirements of the virtual server. You can host the same virtual server on multiple NetScaler appliances residing on the same broadcast domain, by using ARP and ICMP attributes. After you add a VIP (or any IP address), the NetScaler sends, and then responds to, ARP requests. VIPs are the only NetScaler-owned IP addresses that can be disabled. When a VIP address is disabled, the virtual server using it goes down and does not respond to ARP, ICMP, or L4 service requests.

As an alternative to creating VIP addresses one at a time, you can specify a consecutive range of VIP addresses.

To create a VIP address by using the command line interface

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.59 255.255.255.0 -type VIP
 Done
```

To create a range of VIP addresses by using the command line interface

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.[60-64] 255.255.255.0 -type VIP
ip "10.102.29.60" added
ip "10.102.29.61" added
ip "10.102.29.62" added
ip "10.102.29.63" added
ip "10.102.29.64" added
 Done
```

To configure a VIP address by using the configuration utility

Navigate to System > Network > IPs > IPV4s, and add a new IP address or edit an existing address.

To create a range of VIP addresses by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to System > Network > IPs > IPV4s.
2. In the Action list, select Add Range.

To enable or disable an IPv4 VIP address by using the command line interface

At the command prompt, type one of the following sets of commands to enable or disable a VIP and verify the configuration:

- enable ns ip <IPAddress>
- show ns ip <IPAddress>
- disable ns ip <IPAddress>
- show ns ip <IPAddress>

Example

```
> enable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

      IP: 10.102.29.79
      Netmask: 255.255.255.255
      Type: VIP
      state: Enabled
      arp: Enabled
      icmp: Enabled
      vserver: Enabled
      management access: Disabled
        telnet: Disabled
        ftp: Disabled
        ssh: Disabled
        gui: Disabled
        snmp: Disabled
      Restrict access: Disabled
      dynamic routing: Disabled
      hostroute: Disabled
 Done
> disable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

      IP: 10.102.29.79
      Netmask: 255.255.255.255
      Type: VIP
      state: Disabled
      arp: Enabled
      icmp: Enabled
      vserver: Enabled
      management access: Disabled
        telnet: Disabled
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    ftp: Disabled
    ssh: Disabled
    gui: Disabled
    snmp: Disabled
  Restrict access: Disabled
  dynamic routing: Disabled
  hostroute: Disabled

Done
```

To enable or disable a VIP address by using the configuration utility

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. Do one of the following:
   - Select a VIP address.
   - Hold down the **Ctrl** key and select multiple server address entries.
   - Hold down the **Shift** key and select a range of server address entries.
   - Select all the addresses by selecting the checkbox on the left side of the header row.
3. From the **Action** list, select **Disable** or **Enable**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring ARP response Suppression for Virtual IP addresses (VIPs)

Aug 28, 2013

You can configure the NetScaler appliance to respond or not respond to ARP requests for a Virtual IP (VIP) address on the basis of the state of the virtual servers associated with that VIP.

For example, if virtual servers V1, of type HTTP, and V2, of type HTTPs, share VIP address 10.102.29.45 on a NetScaler appliance, you can configure the appliance to not respond to any ARP request for VIP 10.102.29.45 if both V1 and V2 are in the DOWN state.

The following three options are available for configuring ARP-response suppression for a virtual IP address.

- **NONE**. The NetScaler appliance responds to any ARP request for the VIP address, irrespective of the state of the virtual servers associated with the address.
- **ONE VSERVER**. The NetScaler appliance responds to any ARP request for the VIP address if at least one of the associated virtual servers is in UP state.
- **ALL VSERVER**. The NetScaler appliance responds to any ARP request for the VIP address if all of the associated virtual servers are in UP state.

Following table shows the sample behavior of NetScaler appliance for a VIP configured with two virtual servers:

Associated virtual servers for a VIP	STATE 1	STATE 2	STATE 3	STATE 4
NONE				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Respond to an ARP request for this VIP?	Yes	Yes	Yes	Yes
ONE VSERVER				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Respond to an ARP request for this VIP?	Yes	Yes	Yes	No
ALL VSERVER				
V1	UP	UP	DOWN	DOWN

© 1999-2017 Citrix Systems, Inc. All rights reserved.

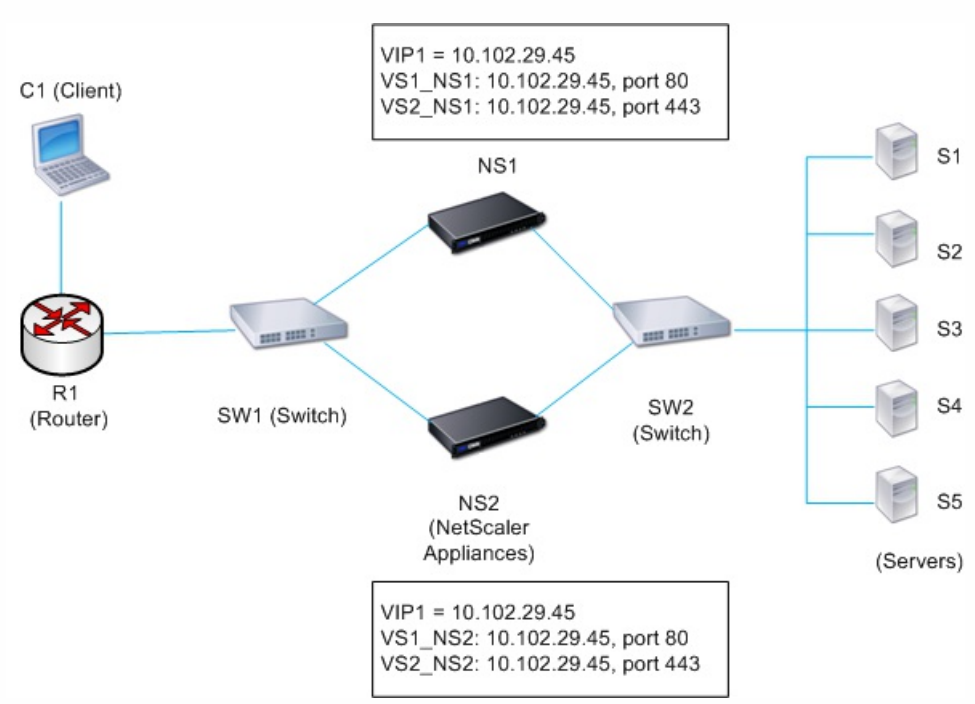
Associated virtual servers for a VIP	STATE 1	STATE 2	STATE 3	STATE 4
Respond to an ARP request for this VIP?	Yes	No	No	No

Consider an example where you want to test the performance of two virtual servers, V1 and V2, which have the same VIP address but are of different types and are each configured on NetScaler appliances NS1 and NS2. Let's call the shared VIP address *VIP1*.

V1 load balances servers S1, S2, and S3. V2 load balances servers S4 and S5.

On both NS1 and NS2, for VIP1, the ARP suppression parameter is set to ALL_VSERVER. If you want to test the performance of V1 and V2 on NS1, you must manually disable V1 and V2 on NS2, so that NS2 does not respond to any ARP request for VIP1.

Figure 1.



The execution flow is as follows:

1. Client C1 sends a request to V1. The request reaches R1.
2. R1 does not have an APR entry for the IP address (VIP1) of V1, so R1 broadcasts an ARP request for VIP1.
3. NS1 replies with source MAC address MAC1 and source IP address VIP1. NS2 does not reply to the ARP request.
4. SW1 learns the port for VIP1 from the ARP reply and updates its bridge table, and R1 updates the ARP entry with MAC1 and VIP1.
5. R1 forwards the packet to address VIP1 on NS1.
6. NS1's load balancing algorithm selects server S2, and NS1 opens a connection between one of its SNIP or MIP addresses and S2. When S2 sends a response to the client, the response returns by the same path.
7. Now you want to test the performance of V1 and V2 on NS2, so you enable V1 and V2 on NS2 and disable them on NS1. NS2 now broadcasts an ARP message for VIP1. In the message, MAC2 is the source MAC address and VIP1 is the source IP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

8. SW1 learns the port number for reaching MAC2 from the ARP broadcast and updates its bridge table to send subsequent client requests for VIP1 to NS2. R1 updates its ARP table.
9. Now suppose the ARP entry for VIP1 times out in the ARP table of R1, and client C1 sends a request for V1. Because R1 does not have an APR entry for VIP1, it broadcasts an ARP request for VIP1.
10. NS2 replies with a source MAC address and VIP1 as the source IP address. NS1 does not reply to the ARP request.

To configure ARP response suppression by using the command line interface

At the command prompt, type:

- set ns ip -arpResponse <arpResponse>]
- show ns ip <IPAddress>

Example

```
> set ns ip 10.102.29.96 -arpResponse ALL_VSERVERS
 Done
```

To configure ARP response suppression by using the configuration utility

1. Navigate to System > Network > IPs > IPV4s.
2. Open an IP address entry and select the type of ARP Response.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Subnet IP Addresses (SNIPs)

May 23, 2014

A subnet IP address (SNIP) is a NetScaler owned IP address that is used by the NetScaler ADC to communicate with the servers.

The NetScaler ADC uses the subnet IP address as a source IP address to proxy client connections to servers. It also uses the subnet IP address when generating its own packets, such as packets related to dynamic routing protocols, or to send monitor probes to check the health of the servers.

Depending on your network topology, you might have to configure one or more SNIPs for different scenarios. Following are three typical scenarios in which you have to configure SNIPs:

- Using SNIPs for a Directly Connected Server Subnet
- Using SNIPs for Server Subnets Connected through a Router
- Using SNIPs for Multiple Server Subnets (VLANs) on an L2 Switch

To configure a SNIP address on a NetScaler ADC, you add the SNIP address and then enable global Use Subnet IP (USNIP) mode.

As an alternative to creating SNIPs one at a time, you can specify a consecutive range of SNIPs.

To configure a SNIP address by using the command line interface

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.203 255.255.255.0 -type SNIP
 Done
```

To create a range of SNIP addresses by using the command line interface

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.[205-209] 255.255.255.0 -type SNIP
ip "10.102.29.205" added
ip "10.102.29.206" added
ip "10.102.29.207" added
ip "10.102.29.208" added
ip "10.102.29.209" added
 Done
```

To enable or disable USNIP mode by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type one of the following commands:

- enable ns modeUSNIP
- disable ns modeUSNIP

To configure a SNIP address by using the configuration utility

Navigate to System > Network > IPs > IPV4s, and add a new SNIP address or edit an existing address.

To create a range of SNIP addresses by using the configuration utility

1. Navigate to System > Network > IPs > IPV4s.
2. In the Action list, select Add Range.

To enable or disable USNIP mode by using the command line interface

At the command prompt, type one of the following commands:

- enable ns mode USNIP
- disable ns mode USNIP

To enable or disable USNIP mode by using the configuration utility

1. Navigate to System > Settings, in Modes and Features group, click Change modes.
2. Select or clear the Use Subnet IP option.

Using SNIPs for a Directly Connected Server Subnet

Updated: 2014-05-23

To enable communication between the NetScaler and a server that is either connected directly to the NetScaler or connected through only an L2 switch, you must configure a subnet IP address that belongs to the subnet of the server. You must configure at least one subnet IP address for each directly connected subnet, except for the directly connected management subnet that is connected through NSIP.

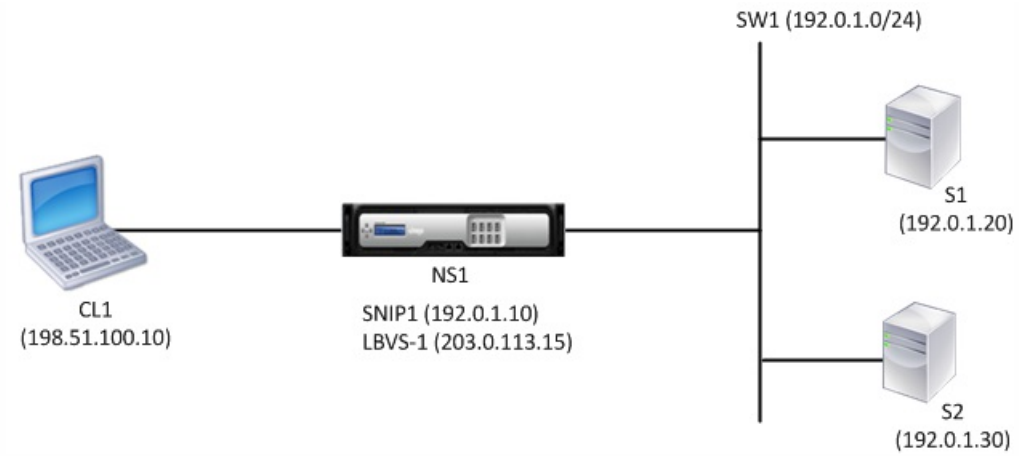
Consider an example of a load balancing setup in which load balancing virtual server LBVS1 on NetScaler ADC NS1 is used to load balance servers S1 and S2, which are connected to NS1 through L2 switch SW1. S1 and S2 belong to the same subnet.

SNIP address SNIP1, which belongs to the same subnet as S1 and S2, is configured on NS1. As soon as SNIP1 is configured, NS1 broadcasts ARP packets for SNIP1.

Services SVC-S1 and SVC-S2 on NS1 represent S1 and S2. As soon as these services are configured, NS1 broadcasts ARP requests for S1 and S2 to resolve IP-to-MAC mapping. After S1 and S2 respond, NS1 sends them monitoring probes at regular intervals, from address SNIP1, to check their health.

For more information about configuring load balancing on a NetScaler ADC, see Load Balancing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Following is the traffic flow in this example:

1. Client C1 sends a request packet to LBVS-1. The request packet has:
   - Source IP = IP address of the client (198.51.100.10)
   - Destination IP = IP address of LBVS-1 (203.0.113.15)
2. LBVS1 of NS1 receives the request packet.
3. LBVS1's load balancing algorithm selects server S2.
4. Because S2 is directly connected to NS1, and SNIP1 (192.0.1.10) is the only IP address on NS1 that belongs to the same subnet as S2, NS1 opens a connection between SNIP1 and S2.
5. NS1 sends the request packet to S2 from SNIP1. The request packet has:
   - Source IP = SNIP1 (192.0.1.10)
   - Destination IP = IP address of S2 (192.0.1.30)
6. S2's response returns by the same path.

Using SNIPs for Server Subnets Connected through a Router

Updated: 2014-05-23

To enable communication between the NetScaler ADC and servers in subnets connected through a router, you must configure at least one subnet IP address that belongs to the subnet of the directly connected interface to the router. The ADC uses this subnet IP address to communicate with servers in subnets that can be reached through the router.

Consider an example of a load balancing setup in which load balancing virtual server LBVS1 on NetScaler ADC NS1 is used to load balance servers S1, S2, S3, and S4, which are connected to NS1 through router R1.

S1 and S2 belong to same subnet, 192.0.2.0/24, and are connected to R1 through L2 switch SW1. S3 and S4 belong to a different subnet, 192.0.3.0/24, and are connected to R1 through L2 switch SW2.

NetScaler ADC NS1 is connected to router R1 through subnet 192.0.1.0/24. SNIP address SNIP1, which belongs to the same subnet as the directly connected interface to the router (192.0.1.0/24), is configured on NS1. NS1 uses this address to communicate with servers S1 and S2, and with servers S3 and S4.

For more information about configuring load balancing on a NetScaler ADC, see Load Balancing.

As soon as address SNIP1 is configured, NS1 broadcasts ARP announcement packets for SNIP1.

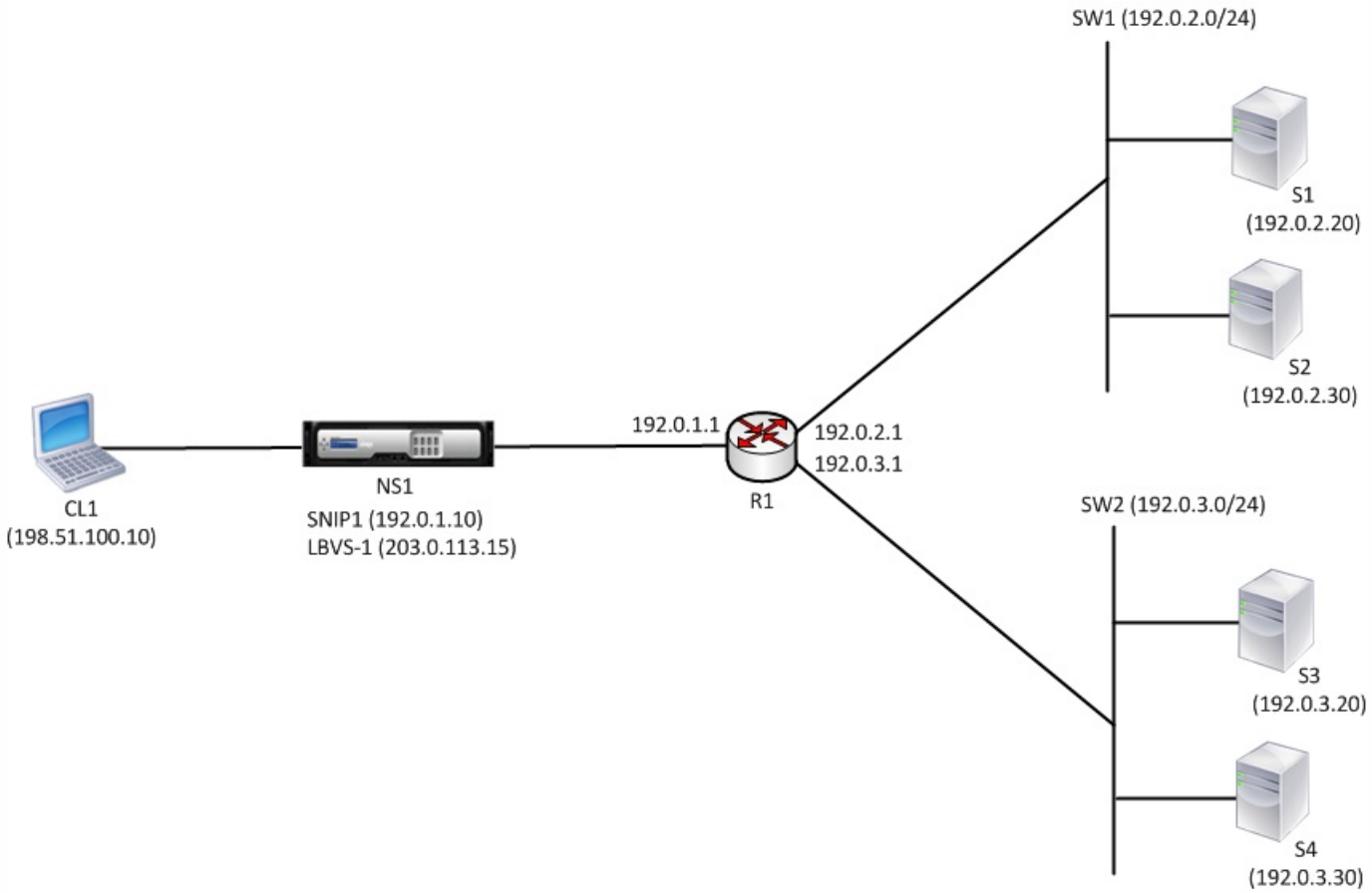
NS1's routing table consists of route entries for S1, S2, S3, and S4 through R1. These route entries are either static route entries or advertised by R1 to NS1, using dynamic routing protocols.

Services SVC-S1, SVC-S2, SVC-S3, and SVC-S4 on NS1 represent servers S1, S2, S3, and S4. NS1 finds, in its routing tables,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

that these servers are reachable through R1. NS1 sends them monitoring probes at regular intervals, from address SNIP1, to check their health.

For more information about IP routing on a NetScaler ADC, see IP Routing.



Following is the traffic flow in this example:

1. Client C1 sends a request packet to LBVS-1. The request packet has:
   - Source IP = IP address of the client (198.51.100.10)
   - Destination IP = IP address of LBVS-1 (203.0.113.15)
2. LBVS1 of NS1 receives the request packet.
3. LBVS1's load balancing algorithm selects server S3.
4. NS1 checks its routing table and finds that S3 is reachable through R1. SNIP1 (192.0.1.10) is the only IP address on NS1 that belongs to the same subnet as router R1, NS1 opens a connection between SNIP1 and S3 through R1.
5. NS1 sends the request packet to R1 from SNIP1. The request packet has:
   - Source IP address = SNIP1 (192.0.1.10)
   - Destination IP address = IP address of S3 (192.0.3.20)
6. The request reaches R1, which checks its routing table and forwards the request packet to S3.
7. S3's response returns by the same path.

Using SNIPs for Multiple Server Subnets (VLANs) on an L2 Switch

Updated: 2014-05-23

When you have multiple server subnets (VLANs) on an L2 switch that is connected to a NetScaler ADC, you must configure at least one SNIP address for each of the server subnets, so that the NetScaler ADC can communicate with these server subnets.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

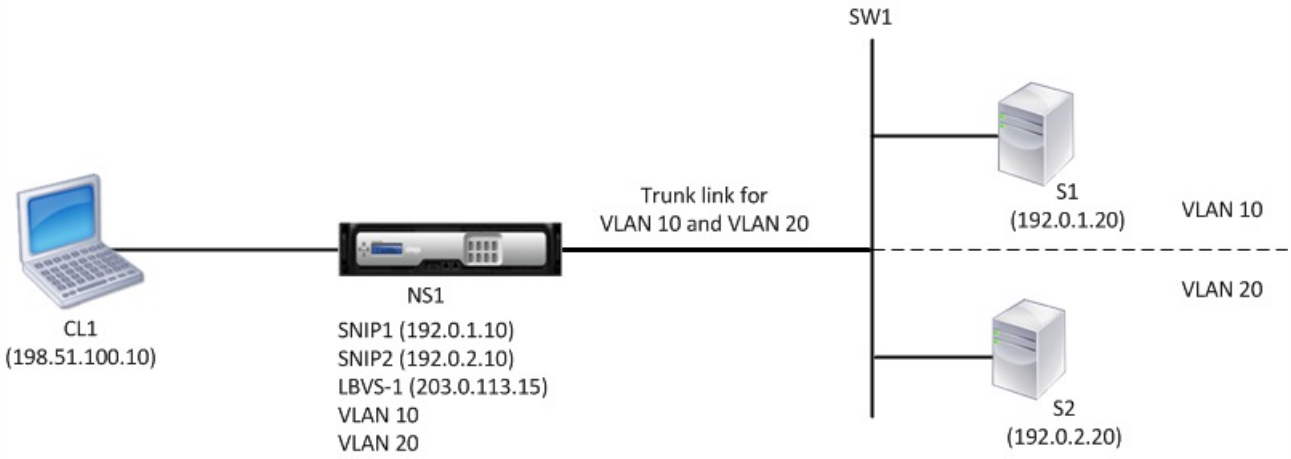
Consider an example of a load balancing setup in which load balancing virtual server LBVS1 on NetScaler ADC NS1 is used to load balance servers S1 and S2, which are connected to NS1 through L2 switch SW1. S1 and S2 belong to different subnets and are part of VLAN 10 and VLAN20, respectively. The link between NS1 and SW1 is a trunk link and is shared by VLAN10 and VLAN20.

For more information about configuring load balancing on a NetScaler ADC, see Load Balancing.

Subnet IP addresses SNIP1 (for reference purposes only) and SNIP2 (for reference purposes only) are configured on NS1. NS1 uses SNIP1 (on VLAN 10) to communicate with server S1, and SNIP2 (on VLAN 20) to communicate with S2. As soon as SNIP1 and SNIP2 are configured, NS1 broadcasts ARP announcement packets for SNIP1 and SNIP2.

For more information about configuring VLANs on a NetScaler ADC, see Configuring a VLAN.

Services SVC-S1 and SVC-S2 on NS1 represent servers S1 and S2. As soon as these services are configured, NS1 broadcasts ARP requests for them. After S1 and S2 respond, NS1 sends them monitoring probes at regular intervals to check their health. NS1 sends monitoring probes to S1 from address SNIP1, and to S2 from address SNIP2.



Following is the traffic flow in this example:

1. Client C1 sends a request packet to LBVS-1. The request packet has:
   - Source IP = IP address of the client (198.51.100.10)
   - Destination IP = IP address of LBVS-1 (203.0.113.15)
2. LBVS1 of NS1 receives the request packet.
3. LBVS1's load balancing algorithm selects server S2.
4. Because S2 is directly connected to NS1, and SNIP2 (192.0.2.10) is the only IP address on NS1 that belongs to the same subnet as S2, NS1 opens a connection between SNIP2 and S2.
   Note: If S1 is selected, NS1 opens a connection between SNIP1 and S1.
5. NS1 sends the request packet to S2 from SNIP2. The request packet has:
   - Source IP = SNIP1 (192.0.2.10)
   - Destination IP = IP address of S2 (192.0.2.20)
6. S2's response returns by the same path.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Mapped IP Addresses (MIPs)

Sep 30, 2013

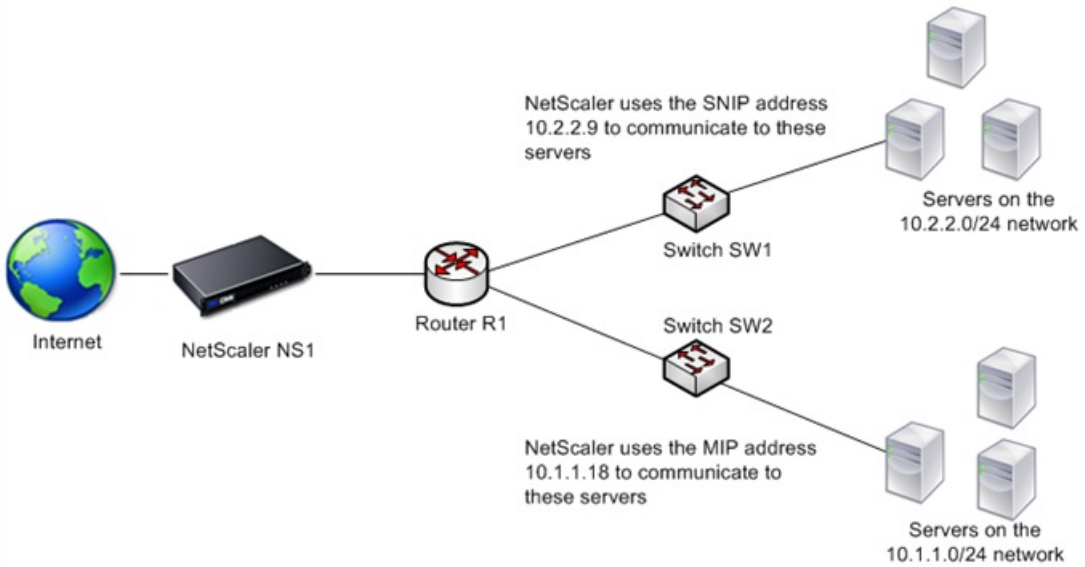
Mapped IP addresses (MIP) are used for server-side connections. A MIP can be considered a default Subnet IP (SNIP) address, because MIPs are used when a SNIP is not available or Use SNIP (USNIP) mode is disabled.

If the mapped IP address is the first in the subnet, the NetScaler appliance adds a route entry, with this IP address as the gateway to reach the subnet. You can create or delete a MIP during run time without rebooting the appliance.

As an alternative to creating MIPs one at a time, you can specify a consecutive range of MIPs.

The following diagram shows the use of the MIP and SNIP addresses in a NetScaler appliance that connects to the backend servers across the subnets.

Figure 1. MIP and SNIP addresses



In the setup, if the NetScaler appliance and the backend servers are in the 10.1.1.0/24 subnet, then the appliance uses the MIP address to communicate to the servers. However, if the setup has backend servers on additional subnets, such as 10.2.2.0/24, and there is no router between the NetScaler appliance and the subnet, then you can configure a SNIP address that has a range of 10.2.2.x/24, such as 10.2.2.9 in this case, to communicate to the additional subnet.

You can enable to NetScaler appliance to use MIP to communicate the additional subnet. However, if the setup has a Firewall application between the appliance and the server, then the Firewall might prevent the traffic other than 10.2.2.0/24. In such cases, you need a SNIP address to communicate to the servers.

To create a MIP address by using the command line interface

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> add ns ip 10.102.29.171 255.255.255.0 -type MIP
 Done
```

To create a range of MIP addresses by using the command line interface

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

Example

```
> add ns ip 10.102.29.[173-175] 255.255.255.0 -type MIP
ip "10.102.29.173" added
ip "10.102.29.174" added
ip "10.102.29.175" added
Done
```

To configure a MIP address by using the configuration utility

Navigate to System > Network > IPs > IPV4s, and add a new MIP address or edit an existing address.

To create a range of MIP addresses by using the configuration utility

1. Navigate to System > Network > IPs > IPV4s.
2. In the Action list, select Add Range.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring GSLB Site IP Addresses (GSLBIP)

Aug 28, 2013

A GSLB site IP (GSLBIP) address is an IP address associated with a GSLB site. It is not mandatory to specify a GSLBIP address when you initially configure the NetScaler appliance. A GSLBIP address is used only when you create a GSLB site.

For more information about creating a GSLB site IP address, see "Global Server Load Balancing."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Removing a NetScaler-Owned IP Address

Aug 28, 2013

You can remove any IP address except the NSIP. The following table provides information about the processes you must follow to remove the various types of IP addresses. Before removing a VIP, remove the associated virtual server.

Table 1. Implications of Removing a NetScaler-Owned IP Address

IP address type	Implications
Subnet IP address (SNIP)	If IP address being removed is the last IP address in the subnet, the associated route is deleted from the route table. If the IP address being removed is the gateway in the corresponding route entry, the gateway for that subnet route is changed to another NetScaler-owned IP address.
Mapped IP address (MIP)	If a SNIP exists, you can remove the MIPs. The NetScaler uses NSIP and SNIPs to communicate with the servers when the MIP is removed. Therefore, you must also enable use SNIP (USNIP) mode. For information about enabling and disabling USNIP mode, see "Configuring Subnet IP Addresses (SNIPs)."
Virtual Server IP address (VIP)	Before removing a VIP, you must first remove the vserver associated with it. For information about removing the vserver, see "Load Balancing."
GSLB-Site-IP address	Before removing a GSLB site IP address, you must remove the site associated with it. For information about removing the site, see "Global Server Load Balancing."

To remove an IP address by using the command line interface

At the command prompt, type:

rm ns ip <IPaddress>

Example

```
rm ns ip 10.102.29.54
```

To remove an IP address by using the configuration utility

Navigate to System > Network > IPs > IPV4s, delete the IP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Application Access Controls

Sep 30, 2015

Application access controls, also known as management access controls, form a unified mechanism for managing user authentication and implementing rules that determine user access to applications and data. You can configure MIPs and SNIPs to provide access for management applications. Management access for the NSIP is enabled by default and cannot be disabled. You can, however, control it by using ACLs.

For information about using ACLs, see Access Control Lists (ACLs).

The NetScaler appliance does not support management access to VIPs.

The following table provides a summary of the interaction between management access and specific service settings for Telnet.

Management Access	Telnet (State Configured on the NetScaler)	Telnet (Effective State at the IP Level)
Enable	Enable	Enable
Enable	Disable	Disable
Disable	Enable	Disable
Disable	Disable	Disable

The following table provides an overview of the IP addresses used as source IP addresses in outbound traffic.

Application/ IP	NSIP	MIP	SNIP	VIP
ARP	Yes	Yes	Yes	No
Server side traffic	No	Yes	Yes	No
RNAT	No	Yes	Yes	Yes
ICMP PING	Yes	Yes	Yes	No
Dynamic routing	Yes	No	Yes	Yes

The following table provides an overview of the applications available on these IP addresses.

Application/ IP	NSIP	MIP	SNIP	VIP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application/ IP	NSIP	MIP	SNIP	VIP
SNMP	Yes	Yes	Yes	Yes
System access	Yes	Yes	Yes	No

You can access and manage the NetScaler by using applications such as Telnet, SSH, GUI, and FTP.

Note: Telnet and FTP are disabled on the NetScaler for security reasons. To enable them, contact the customer support. After the applications are enabled, you can apply the controls at the IP level.
To configure the NetScaler to respond to these applications, you need to enable the specific management applications. If you disable management access for an IP address, existing connections that use the IP address are not terminated, but no new connections can be initiated.

Also, the non-management applications running on the underlying FreeBSD operating system are open to protocol attacks, and these applications do not take advantage of the NetScaler appliance's attack prevention capabilities.

You can block access to these non-management applications on a MIP, SNIP, or NSIP. When access is blocked, a user connecting to a NetScaler by using the MIP, SNIP, or NSIP is not be able to access the non-management applications running on the underlying operating system.

To configure management access for an IP address by using the command line interface

At the command prompt, type:

set ns ip <IPAddress> -mgmtAccess <value> -telnet <value> -ftp <value> -gui <value> -ssh <value> -snmp <value> -restrictAccess (ENABLED | DISABLED)

Example

```
> set ns ip 10.102.29.54 -mgmtAccess enabled -restrictAccess ENABLED
Done
```

To enable management access for an IP address by using the configuration utility

1. Navigate to System > Network > IPs > IPV4s.
2. Open an IP address entry, and select the Enable Management Access control to support the below listed applications option.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How the NetScaler Proxies Connections

Aug 28, 2013

When a client initiates a connection, the NetScaler appliance terminates the client connection, initiates a connection to an appropriate server, and sends the packet to the server. The appliance does not perform this action for service type UDP or ANY.

For more information about service types, see "Load Balancing."

You can configure the NetScaler to process the packet before initiating the connection with a server. The default behavior is to change the source and destination IP addresses of a packet before sending the packet to the server. You can configure the NetScaler to retain the source IP address of the packets by enabling Use Source IP mode.

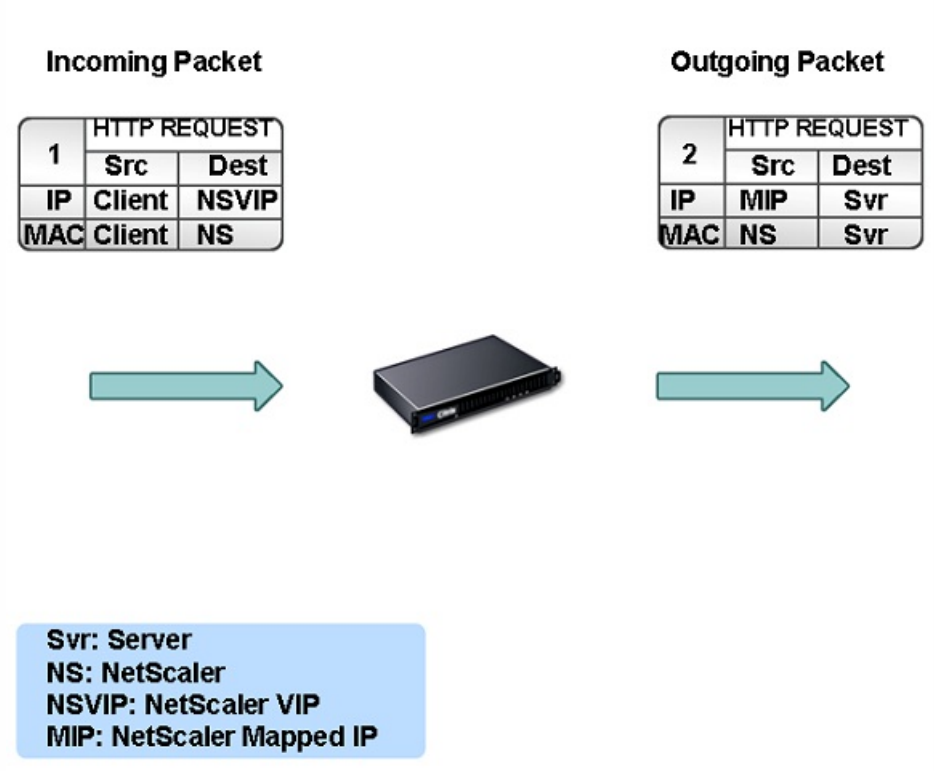
This section includes the following details:

- How the Destination IP Address Is Selected
- How the Source IP Address Is Selected

How the Destination IP Address Is Selected

Traffic sent to the NetScaler appliance can be sent to a virtual server or to a service. The appliance handles traffic to virtual servers and services differently. The NetScaler terminates traffic received at a virtual server IP (VIP) address and changes the destination IP address to the IP address of the server before forwarding the traffic to the server, as shown in the following diagram.

Figure 1. Proxying Connections to VIPs



Packets destined for a service are sent directly to the appropriate server, and the NetScaler does not modify the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

destination IP addresses. In this case, the NetScaler functions as a proxy.

How the Source IP Address Is Selected

When the NetScaler appliance communicates with the physical servers or peer devices, by default, it does not use the IP address of the client. NetScaler maintains a pool of mapped IP addresses (MIPs) and subnet IP addresses (SNIPs), and selects an IP address from this pool to use as the source IP address of a connection to the physical server. Depending on the subnet in which the physical server is placed, NetScaler decides whether a MIP should be used or SNIP.

Note: If the Use Source IP (USIP) option is enabled, NetScaler uses the IP address of the client.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Use Source IP Mode

Dec 16, 2013

When the NetScaler appliance communicates with the physical servers or peer devices, by default, it uses one of its own IP addresses as the source IP. The appliance maintains a pool of mapped IP addresses (MIPs) and subnet IP addresses (SNIPs), and selects an IP address from this pool to use as the source IP address for a connection to the physical server. The decision of whether to select a MIP or a SNIP depends on the subnet in which the physical server resides.

If necessary, you can configure the NetScaler appliance to use the client's IP address as source IP. Some applications need the actual IP address of the client. The following use cases are a few examples:

- Client's IP address in the web access log is used for billing purposes or usage analysis.
- Client's IP address is used to determine the country of origin of the client or the originating ISP of the client. For example, many search engines such as Goggle provide content relevant to the location to which the user belongs.
- The application must know the client's IP address to verify that the request is from a trustworthy source.
- Sometimes, even though an application server does not need the client's IP address, a firewall placed between the application server and the NetScaler may need the client's IP address for filtering the traffic.

Enable Use Source IP mode (USIP) mode if you want NetScaler to use the client's IP address for communication with the servers. By default, USIP mode is disabled. USIP mode can be enabled globally on the NetScaler or on a specific service. If you enable it globally, USIP is enabled by default for all subsequently created services. If you enable USIP for a specific service, the client's IP address is used only for the traffic directed to that service.

As an alternative to USIP mode, you have the option of inserting the client's IP address (CIP) in the request header of the server-side connection for an application server that needs the client's IP address.

In earlier NetScaler releases, USIP mode had the following source-port options for server-side connections:

- Use the client's port. With this option, connections cannot be reused. For every request from the client, a new connection is made with the physical server.
- Use proxy port. With this option, connection reuse is possible for all requests from the same client. Before NetScaler release 8.1 this option imposed a limit of 64000 concurrent connections for all server-side connections.

In the later NetScaler releases , if USIP is enabled, the default is to use a proxy port for server-side connections and not reuse connections. Not reusing connections may not affect the speed of establishing connections.

By default, the Use Proxy Port option is enabled if the USIP mode is enabled.

For more information about the Use Proxy Port option, see "Using the Client Port When Connecting to the Server."

Note: If you enable the USIP mode, it is recommended to enable the Use Proxy Port option.
The following figure shows how the NetScaler uses IP addresses in USIP mode.

Figure 1. IP Addressing in USIP Mode

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Recommended Usage

Enable USIP in the following situations:

- Load balancing of Intrusion Detection System (IDS) servers
- SMTP load balancing
- Stateless connection failover
- Sessionless load balancing
- If you use the Direct Server Return (DSR) mode

Note: When USIP is enabled, you must set server's gateway to one of the NetScaler owned IP addresses (either of type Subnet IP (SNIP) or mapped IP (MIP)) so that server's response always go through the NetScaler appliance. For more information about NetScaler owned IP addresses, see "Configuring NetScaler owned IP addresses."

- If you enable USIP, set the idle timeout for server connections to a value lower than the default value, so that idle connections are cleared quickly on the server side.
- For transparent cache redirection, if you enable USIP, enable L2CONN also.
- Because HTTP connections are not reused when USIP is enabled, a large number of server-side connections may accumulate. Idle server connections can block connections for other clients. Therefore, set limits on maximum number of connections to a service. Citrix also recommends setting the HTTP server time-out value, for a service on which USIP is enabled, to a value lower than the default, so that idle connections are cleared quickly on the server side.

To globally enable or disable USIP mode by using the command line interface

At the command prompt, type one of the following commands:

- enable ns mode USIP
- disable ns mode USIP

To enable USIP mode for a service by using the command line interface

At the command prompt, type:

set service <name>@ -usip (**YES** | **NO**)

Example

```
set service Service-HTTP-1 -usip YES
```

To globally enable or disable USIP mode by using the configuration utility

1. Navigate to System > Settings, in Modes and Features group, click Change modes.
2. Select or clear the Use Source IP option.

To enable USIP mode for a service by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Use Source IP Address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Network Address Translation

Jul 05, 2013

Network address translation (NAT) involves modification of the source and/or destination IP addresses and/or the TCP/UDP port numbers of IP packets that pass through the NetScaler appliance. Enabling NAT on the appliance enhances the security of your private network, and protects it from a public network such as the Internet, by modifying your networks source IP addresses when data passes through the NetScaler. Also, with the help of NAT entries, your entire private network can be represented by a few shared public IP addresses. The NetScaler supports the following types of network address translation:

- Inbound NAT (INAT), in which the NetScaler replaces the destination IP address in the packets generated by the client with the private IP address of the server.
- Reverse NAT (RNAT), in which the NetScaler replaces the source IP address in the packets generated by the servers with the public NAT IP addresses.

This document includes the following information:

- Configuring INAT
- Coexistence of INAT and Virtual Servers
- Stateless NAT46 Translation
- DNS64
- Stateful NAT64 Translation
- Configuring RNAT
- Configuring Prefix-Based IPv6-IPv4 Translation

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring INAT

Aug 28, 2013

When a client sends a packet to a NetScaler appliance that is configured for Inbound Network Address Translation (INAT), the appliance translates the packet's public destination IP address to a private destination IP address and forwards the packet to the server at that address.

The following configurations are supported:

- **IPv4-IPv4 Mapping**: A public IPv4 address on the NetScaler appliance listens to connection requests on behalf of a private IPv4 server. The NetScaler appliance translates the packet's public destination IP address to the destination IP address of the server and forwards the packet to the server at that address.
- **IPv4-IPv6 Mapping**: A public IPv4 address on the NetScaler appliance listens to connection requests on behalf of a private IPv6 server. The NetScaler appliance creates an IPv6 request packet with the IP address of the IPv6 server as the destination IP address.
- **IPv6-IPv4 Mapping**: A public IPv6 address on the NetScaler appliance listens to connection requests on behalf of a private IPv4 server. The NetScaler appliance creates an IPv4 request packet with the IP address of the IPv4 server as the destination IP address.
- **IPv6-IPv6 Mapping**: A public IPv6 address on the NetScaler appliance listens to connection requests on behalf of a private IPv6 server. The NetScaler appliance translates the packet's public destination IP address to the destination IP address of the server and forwards the packet to the server at that address.

When the appliance forwards a packet to a server, the source IP address assigned to the packet is determined as follows:

- If use subnet IP (USNIP) mode is enabled and use source IP (USIP) mode is disabled, the NetScaler uses a subnet IP address (SNIP) as the source IP address.
- If USNIP mode is disabled and USIP mode is disabled, the NetScaler uses a mapped IP address (MIP) as the source IP address.
- If USIP mode is enabled, and USNIP mode is disabled the NetScaler uses the client IP (CIP) address as the source IP address.
- If both USIP and USNIP modes are enabled, USIP mode takes precedence.
- You can also configure the NetScaler to use a unique IP address as the source IP address, by setting the proxyIP parameter.
- If none of the above modes are enabled and a unique IP address has not been specified, the NetScaler attempts to use a MIP as the source IP address.
- If both USIP and USNIP modes are enabled and a unique IP address has been specified, the order of precedence is as follows: USIP-unique IP-USNIP-MIP-Error.

To protect the NetScaler from DoS attacks, you can enable TCP proxy. However, if other protection mechanisms are used in your network, you may want to disable them.

You can create, modify, or remove an INAT entry.

To create an INAT entry by using the command line interface

At the command prompt, type the following commands to create an INAT entry and verify its configuration:

- add inat <name> <publicIP> <privateIP> [-tcpproxy (ENABLED | DISABLED)] [-ftp (ENABLED | DISABLED)] [-usip (ON | OFF)] [-usnip (ON | OFF)] [-proxyIP <ip_addr|ipv6_addr>]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- show inat [<name>]

Example

```
> add inat ip4-ip4 172.16.1.2 192.168.1.1 -proxyip 10.102.29.171
 Done
```

To modify an INAT entry by using the command line interface

To modify an INAT entry, type the set inat command, the name of the entry, and the parameters to be changed, with their new values.

To remove an INAT configuration by using the command line interface

At the command prompt, type:

rm inat <name>

Example

```
> rm inat ip4-ip4
 Done
```

To configure an INAT entry by using the configuration utility

Navigate to System > Network > Routes > INAT, and add a new INAT entry or edit an existing INAT entry.

To remove an INAT configuration by using the configuration utility

Navigate to System > Network > Routes > INAT, delete the INAT configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Coexistence of INAT and Virtual Servers

Mar 20, 2012

If both INAT and RNAT are configured, the INAT rule takes precedence over the RNAT rule. If RNAT is configured with a network address translation IP (NAT IP) address, the NAT IP address is selected as the source IP address for that RNAT client.

The default public destination IP in an INAT configuration is the virtual IP (VIP) address of the NetScaler device. Virtual servers also use VIPs. When both INAT and a virtual server use the same IP address, the Vserver configuration overrides the INAT configuration.

Following are a few sample configuration setup scenarios and their effects.

Case	Result
You have configured a virtual server and a service to send all data packets received on a specific NetScaler port to the server directly. You have also configured INAT and enabled TCP. Configuring INAT in this manner sends all data packets received through a TCP engine before sending them to the server.	All packets received on the NetScaler, except those received on the specified port, pass through the TCP engine.
You have configured a virtual server and a service to send all data packets of service type TCP, that are received on a specific port on the NetScaler, to the server after passing through the TCP engine. You have also configured INAT and disabled TCP. Configuring INAT in this manner sends the data packets received directly to the server.	Only packets received on the specified port pass through the TCP engine.
You have configured a virtual server and a service to send all data packets received to either of two servers. You are attempting to configure INAT to send all data packets received to a different server.	The INAT configuration is not allowed.
You have configured INAT to send all received data packets directly to a server. You are attempting to configure a virtual server and a service to send all data packets received to two different servers.	The vserver configuration is not allowed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Stateless NAT46 Translation

Aug 28, 2013

The stateless NAT46 feature enables communication between IPv4 and IPv6 networks through IPv4 to IPv6 packet translation, and vice versa, without maintaining any session information on the NetScaler appliance.

For a stateless NAT46 configuration, the appliance translates an IPv4 packet to IPv6 or an IPv6 packet to IPv4 as defined in RFCs 6145 and 2765.

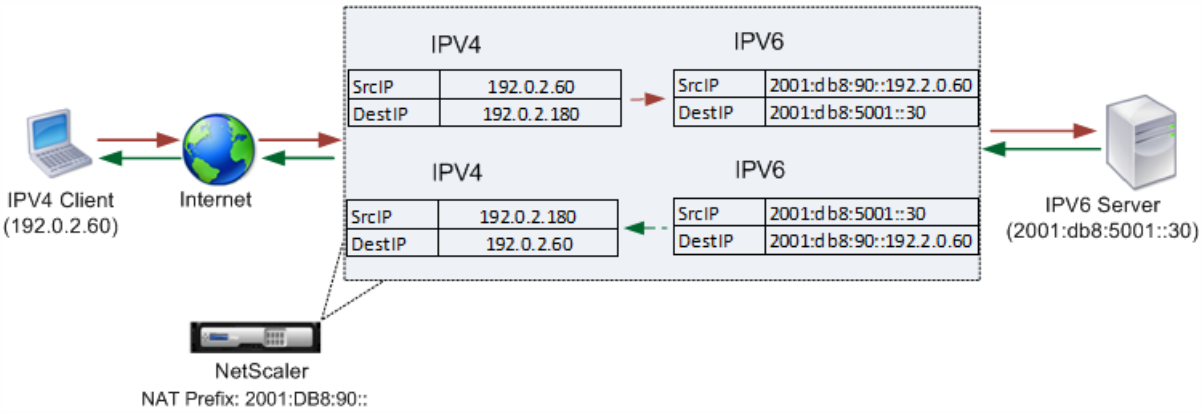
Note: This feature is supported only on NetScaler 10.e and later.

A stateless NAT46 configuration on the NetScaler appliance has the following components:

- **IPv4-IPv6 INAT entry**—An INAT entry defining a 1:1 relationship between an IPv4 address and an IPv6 address. In other words, an IPv4 address on the appliance listens to connection requests on behalf of an IPv6 server. An IPv4 request packet for this IPv4 address is translated into an IPv6 packet, and then the IPv6 packet is sent to the IPv6 server.
  The appliance translates an IPv6 response packet into an IPv4 response packet with its source IP address field set as the IPv4 address specified in the INAT entry. The translated packet is then sent to the client.

- **NAT46 IPv6 prefix**—A global IPv6 prefix of length 96 bits (128-32=96) configured on the appliance. During IPv4 packet to IPv6 packet translation, the appliance sets the source IP address of the translated IPv6 packet to a concatenation of the NAT46 IPv6 prefix [96 bits] and the IPv4 source address [32 bits] that was received in the request packet.
  During IPv6 packet to IPv4 packet translation, the appliance sets the destination IP address of the translated IPv4 packet to the last 32 bits of the destination IP address of the IPv6 packet.

Consider an example in which an enterprise hosts site www.example.com on server S1, which has an IPv6 address. To enable communication between IPv4 clients and IPv6 server S1, NetScaler appliance NS1 is deployed with a stateless NAT46 configuration that includes an IPv4-IPv6 INAT entry for server S1, and a NAT46 Prefix. The INAT entry includes an IPv4 address at which the appliance listens to connection requests from IPv4 clients on behalf of the IPv6 server S1.



The following table lists the settings used in this example:

Entities	Name	Value
IP address of the client	Client_IPv4 (for reference purposes only)	192.0.2.60
IPv6 address of the server	Sevr_IPv6 (for reference purposes only)	2001:DB8:5001::30

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entities	Name	Value
IPv4 address defined in the INAT entry for IPv6 server S1	Map-Sevr-IPv4 (for reference purposes only)	192.0.2.180
IPv6 prefix for NAT 46 translation	NAT46_Prefix (for reference purposes only)	2001:DB8:90::

Following is the traffic flow in this example:

1. IPv4 Client CL1 sends a request packet to the Map-Sevr-IPv4 (192.0.2.180) address on the NetScaler appliance.
2. The appliance receives the request packet and searches the NAT46 INAT entries for the IPv6 address mapped to the Map-sevr-IPv4 (192.0.2.180) address. It finds the Sevr-IPv6 (2001:DB8:5001::30) address.
3. The appliance creates a translated IPv6 request packet with:
   - Destination IP address field = Sevr-IPv6 = 2001:DB8:5001::30
   - Source IP address field = Concatenation of NAT Prefix (First 96 bits) and Client_IPv4 (last 32 bits) = 2001:DB8:90::192.0.2.60
4. The appliance sends the translated IPv6 request to Sevr-IPv6.
5. The IPv6 server S1 responds by sending an IPv6 packet to the NetScaler appliance with:
   - Destination IP address field = Concatenation of NAT Prefix (First 96 bits) and Client_IPv4 (last 32 bits)= 2001:DB8:90::192.0.2.60
   - Source IP address field = Sevr-IPv6 = 2001:DB8:5001::30
6. The appliance receives the IPv6 response packet and verifies that its destination IP address matches the NAT46 prefix configured on the appliance. Because the destination address matches the NAT46 prefix, the appliance searches the NAT46 INAT entries for the IPv4 address associated with the Sevr-IPv6 address (2001:DB8:5001::30). It finds the Map-Sevr-IPv4 address (192.0.2.180).
7. The appliance creates an IPv4 response packet with:
   - Destination IP address field = The NAT46 prefix stripped from the destination address of the IPv6 response = Client_IPv4 (192.0.2.60)
   - Source IP address field = Map-Sevr-IPv4 address (192.0.2.180)
8. The appliance sends the translated IPv4 response to client CL1.

Configuring Stateless NAT46

Updated: 2013-09-04

Creating the required entities for stateless NAT46 configuration on the NetScaler appliance involves the following procedures:

1. Create an IPv4-IPv6 mapping INAT entry with stateless mode enabled.
2. Add a NAT46 IPv6 prefix.

To configure an INAT mapping entry by using the command line interface

At the command prompt, type:

- add inat <name> <publicIPv4> <privateIPv6> -mode STATELESS
- show inat <name>

To add an NAT46 prefix by using the command line interface

At the command prompt, type:

- set inatparam -nat46v6Prefix <ipv6_addr|*>
- show inatparam

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> add inat exmpl-com-stls-nat46 192.0.2.180
2001:DB8:5001::30 -mode stateless
Done

> set inatparam -nat46v6Prefix 2001:DB8:90::/96
Done
```

To configure an INAT mapping entry by using the configuration utility

1. Navigate to System > Network > Routes > INAT.
2. Add a new INAT entry, or edit an existing INAT entry.
3. Set the following parameters:
   - Name*
   - Public IP Address*
   - Private IP Address* (Select the IPv6 check box and enter the address in IPv6 format.)
   - Mode (Select Stateless from the drop down list.)

   * A required parameter

To add a NAT46 prefix by using the configuration utility

Navigate to System > Network, in the Settings group, click Configure INAT Parameters, and set the Prefix parameter.

Setting Global Parameters for Stateless NAT46

The appliance provides some optional global parameters for stateless NAT46 configurations.

To set global parameters for stateless NAT46 by using the command line interface

At the command prompt, type:

- set inatparam [-nat46IgnoreTOS (YES | NO)] [-nat46ZeroCheckSum (ENABLED | DISABLED)] [-nat46v6Mtu <positive_integer>] [-nat46FragHeader (ENABLED | DISABLED)]
- show inatparam

Example

```
> set inatparam -nat46IgnoreTOS YES -nat46ZeroCheckSum DISABLED -nat46v6Mtu 1400 -nat46FragHeader DISABLED
 Done
```

To set global parameters for stateless NAT46 by using the configuration utility

Navigate to System > Network, in the Settings group, click Configure INAT Parameters.

Limitations of Stateless NAT46

The following limitations apply to stateless NAT46:

- Translation of IPv4 options is not supported.
- Translation of IPv6 routing headers is not supported.
- Translation of hop-by-hop extension headers of IPv6 packets is not supported.
- Translation of ESP and EH headers of IPv4 packets is not supported.
- Translation of multicast packets is not supported.
- Translation of destination option headers and source routing headers is not supported.
- Translation of fragmented IPv4 UDP packets that do not contain UDP checksum is not supported.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

DNS64

Feb 23, 2014

The NetScaler DNS64 feature responds with a synthesized DNS AAAA record to an IPv6 client sending an AAAA request for an IPv4-only domain. The DNS64 feature is used with the NAT64 feature to enable seamless communication between IPv6-only clients and IPv4-only servers. DNS64 enables discovery of the IPv4domain by the IPV6 only clients, and NAT64 enables communication between the clients and servers.

For synthesizing an AAAA record, the NetScaler appliance fetches a DNS A record from a DNS server. The DNS64 prefix is a 96-bit IPv6 prefix configured on the NetScaler appliance. The NetScaler appliance synthesizes the AAAA record by concatenation of the DNS64 Prefix (96 bits) and the IPv4 address (32 bits).

For enabling communication between IPv6 clients and IPv4 servers, a NetScaler appliance with DNS64 and NAT64 configuration can deployed either on the IPv6 client side or on the IPv4 server side. In both cases, the DNS64 configuration on the NetScaler appliance is similar and includes a load balancing virtual server acting as a proxy server for DNS servers. If the NetScaler appliance is deployed on the client side, the load balancing virtual server must be specified, on the IPv6 client, as the nameserver for a domain.

Consider an example where a NetScaler appliance with DNS64 and NAT64 configuration is configured on the IPv4 side. In this example, an enterprise hosts site www.example.com on server S1, which has an IPv4 address. To enable communication between IPv6 clients and IPv4 server S1, NetScaler appliance NS1 is deployed with a DNS64 and stateful NAT64 configuration.

The DNS64 configuration includes DNS load balancing virtual server LBVS-DNS64-1, on which the DNS64 option is enabled. A DNS64 policy named DNS64-Policy-1, and an associated DNS64 action named DNS64-Action-1, are also configured on NS1, andDNS64-Policy-1 is bound to LBVS-DNS64-1. LBVS-DNS64-1 acts as a DNS proxy server for DNS servers DNS-1 and DNS-2.

When traffic arriving at LBVS-DNS64-1 matches the conditions specified in DNS64-Policy-1, the traffic is processed according to the settings in DNS64-Action-1. DNS64-Action-1 specifies the DNS64 prefix used, with the A record received from a DNS server, to synthesize an AAAA record.

The global DNS parameter cacherecords is enabled on the NetScaler appliance, so the appliance caches DNS records. This setting is necessary for the DNS64 to work properly.

The following table lists the settings used in the above example:

Entity	Name	Value
IPv6 client	CL1 (for reference purposes only)	• IP address = 2001:DB8:5001::30
DNS64 Prefix		• 2001:DB8:300::
Service on NS representing DNS server DNS-1	SVC-DNS-1	• IP address = 203.0.113.50 • Port = 53

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Service on NS representing DNS server DNS-2	SVC-DNS-2	• IP address = 203.0.113.60 • Port = 53
DNS64 action	DNS64-Action-1	• DNS64 Prefix=2001:DB8:300::
DNS64 policy	DNS64-Policy-1	• DNS64 action = DNS64-Action-1 • Rule= CLIENT.IP.SRC.IN_SUBNET(2001:DB8:5001::/64)
DNS load balancing virtual server	LBVS-DNS64-1	• IP address=2001:DB8:9999::99 • Bound DNS services= SVC-DNS-1, SVC-DNS-2 • DNS64=Enabled • Bound DNS64 policy= DNS64-Policy-1

Following is the traffic flow in this example:

1. IPv6 client CL1 sends a DNS AAAA request for the IPv6 address of the site www.example.com.
2. The request is received by the DNS load balancing virtual server LBVS-DNS64-1 on NetScaler appliance NS1.
3. NS1 checks its DNS cache records for the requested AAAA record and finds that AAAA record for the site www.example.com does not exist in the DNS cache.
4. LBVS-DNS64-1's load balancing algorithm selects DNS server DNS-1 and forwards the AAAA request to it.
5. Because the site www.example.com is hosted on an IPv4 server, the DNS server DNS-1 does not have any AAAA record for the site www.example.com.
6. DNS-1 sends either an empty DNS AAAA response or an error message to LBVS-DNS64-1.
7. Because DNS64 option is enabled on LBVS-DNS64-1 and the AAAA request from CL1 matches the condition specified in DNS64-Policy-1, NS1 sends a DNS A request to DNS-1 for the IPv4 address of www.example.com.
8. DNS-1 responds by sending the DNS A record for www.example.com to LBVS-DNS64-1. The A record includes the IPv4 address for www.example.com.
9. NS1 synthesizes an AAAA record for the site www.example.com with:
   - IPv6 address for site www.example.com = Concatenation of DNS64 Prefix (96 bits) specified in the associated DNS64action, and IPv4 address of DNS A record (32 bits) = 2001:DB8:300::192.0.2.60
10. NS1 sends the synthesized AAAA record to IPv6 client CL1. NS1 also caches the A record into its memory. NS1 uses the cached A record to synthesize AAAA records for subsequent AAAA requests.

•

Points to Consider for a DNS64 Configuration

Before configuring DNS64 on a NetScaler appliance, consider the following points:

- The DNS64 feature of the NetScaler appliance is compliant with RFC 6174.
- The DNS64 feature of the NetScaler appliance does not support DNSSEC. The NetScaler appliance does not synthesize an AAAA record from a DNSSEC response received from a DNS server. A response is classified as a DNSSEC response, only if it contains RRSIG records.
- The NetScaler appliance supports DNS64 prefix of length of only 96 bits.
- Though the DNS64 feature is used with the NAT64 feature, the DNS64 and NAT64 configurations are independent on

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the NetScaler appliance. For a particular flow, you must specify the same IPv6 prefix value for the DNS64 prefix and the NAT64 prefix parameters, so that the synthesized IPv6 addresses received by the client are routed to the particular NAT64 configuration. For more information on configuring NAT64 on a NetScaler appliance, see "Stateful NAT64."

- The following are the different cases of DN64 processing by the NetScaler appliance:
  - If the AAAA response from the DNS server includes AAAA records, then each record in the response is checked for the set of exclusion rule configured on the NetScaler appliance for the particular DNS64 configuration. The NetScaler removes the IPv6 addresses, whose prefix matches the exclusion rule, from the response. If the resulting response includes at least one IPv6 record, the NetScaler appliance forwards this response to the client, else, the appliance synthesizes a AAAA response from the A record of the domain and sends it to the IPv6 client.
  - If the AAAA response from the DNS server is an empty answer response, the appliance requests for A resource records with the same domain name or searches in its own records if the appliance is an authentic domain name server for the domain. If the request results in an empty answer or error, the same is forwarded to the client.
  - If the response from the DNS server includes RCODE=1 (format error), the NetScaler appliance forwards the same to the client. If there is no response before the timeout, the NetScaler appliance sends a response with RCODE=2 (server failure) to the client.
  - If the response from the DNS server includes a CNAME, the chain is followed until the terminating A or AAAA record is reached. If the CNAME does not have any AAAA resource records, the NetScaler appliance fetches the DNS A record to be used for synthesizing AAAA record. The CNAME chain is added to the answer section along with the synthesized AAAA record and then sent to the client.
- The DNS64 feature of the NetScaler appliance also supports responding to PTR request. When a PTR request for a domain of an IPv6 address is received on the appliance and the IPv6 address matches any of the configured DNS64 prefix, the appliance creates a CNAME record mapping the IP6-ARPA domain into the corresponding IN-ADDR.ARPA domain and the newly formed IN-ADDR.ARPA domain is used for resolution. The appliance searches the local PTR records and if the records are not present, the appliance sends a PTR request for IN-ADDR.ARPA domain to the DNS server. The NetScaler appliance uses the response from the DNS server to synthesize response for the initial PTR request.

Configuration Steps

Updated: 2013-09-30

Creating the required entities for stateful NAT64 configuration on the NetScaler appliance involves the following procedures:

- **Add DNS services.** DNS services are logical representation of DNS servers for which the NetScaler appliance acts as a DNS proxy server. For more information on setting optional parameters of a service, see "Load Balancing".
- **Add DNS64 action and DNS64 policy and then bind the DNS64 action to the DNS64 policy.** A DNS64 policy specifies conditions to be matched against traffic for DNS64 processing according to the settings in the associated DNS64 action. The DNS64 action specifies the mandatory DNS64 prefix and the optional exclude rule and mapped rule settings.
- **Create a DNS load balancing virtual server and bind the DNS services and the DNS64 policy to it.** The DNS load balancing virtual server acts as a DNS proxy server for DNS servers represented by the bound DNS services. Traffic arriving at the virtual server is matched against the bound DNS64 policy for DNS64 processing. For more information on setting optional parameters of a load balancing virtual server, see "Load Balancing".
  Note: The command line interface has separate commands for these two tasks, but the configuration utility combines them in a single dialog box.
- **Enable caching of DNS records.** Enable the global parameter for the NetScaler appliance to cache DNS records, which are obtained through DNS proxy operations. For more information on enabling caching of DNS records, see "Enabling

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Caching of DNS Records".

To create a service of type DNS by using the command line interface

At the command prompt, type:

- add service <name> <IP> <serviceType> <port> ...

To create a DNS64 action by using the command line interface

At the command prompt, type:

- add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule <expression>] [-excludeRule <expression>]

To create a DNS64 policy by using the command line interface

At the command prompt, type:

- add dns policy64 <name> -rule <expression> -action <string>

To create a DNS load balancing virtual server by using the command line interface

At the command prompt, type:

- add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED) [-bypassAAAA (YES | NO)] ...

To bind the DNS services and the DNS64 policy to the DNS load balancing virtual server by using the command line interface

At the command prompt, type:

- bind lb vserver <name> <serviceName> ...
- bind lb vserver <name> -policyName <string> -priority <positive_integer> ...

Example

```
> add service SVC-DNS-1 203.0.113.50 DNS 53
 Done

> add service SVC-DNS-2 203.0.113.60 DNS 53
 Done

> add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
 Done

> add dns Policy64 DNS64-Policy-1 -rule "CLIENT.IPv6.SRC.IN_SUBNET(2001:DB8:5001::/64)"
-action DNS64-Action-1
 Done

> add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-1
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-2
 Done

> bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Done

To create a service of type DNS by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and add a new service.
2. Set the following parameters:
   - Service Name*
   - Server*
   - Protocol* (Select DNS from the drop down list.)
   - Port*

To create a DNS64 action by using the configuration utility

Navigate to Traffic Management > DNS > Actions, on the DNS Actions64 tab, add a new DNS64 action.

To create a DNS64 policy by using the configuration utility

Navigate to Traffic Management > DNS > Policies, on the DNS Policies64 tab, add a new DNS64 policy.

To create a DNS load balancing virtual server and bind the DNS services and the DNS64 policy to it by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and add a new virtual server.
2. Set the following parameters:
   - Name*
   - IP Address*
   - Protocol* (Select DNS from the drop down list.)
   - Port*
3. Select the Enable DNS64 option.
4. In the Services pane, bind the service to the virtual server.
5. In the Policies pane, bind the policy to the virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Stateful NAT64 Translation

May 11, 2012

The stateful NAT64 feature enables communication between IPv6 clients and IPv4 servers through IPv6 to IPv4 packet translation, and vice versa, while maintaining session information on the NetScaler appliance.

A stateful NAT64 configuration on the NetScaler appliance has the following components:

- **NAT64 rule**— An entry consisting of an ACL6 rule and a netprofile, which consists of a pool of NetScaler owned SNIP Addresses.
- **NAT64 IPv6 Prefix**— A global IPv6 prefix of length 96 bits (128-32=96) configured on the appliance.
  Note: Currently the NetScaler appliance supports only one prefix to be used commonly with all NAT 64 rules.

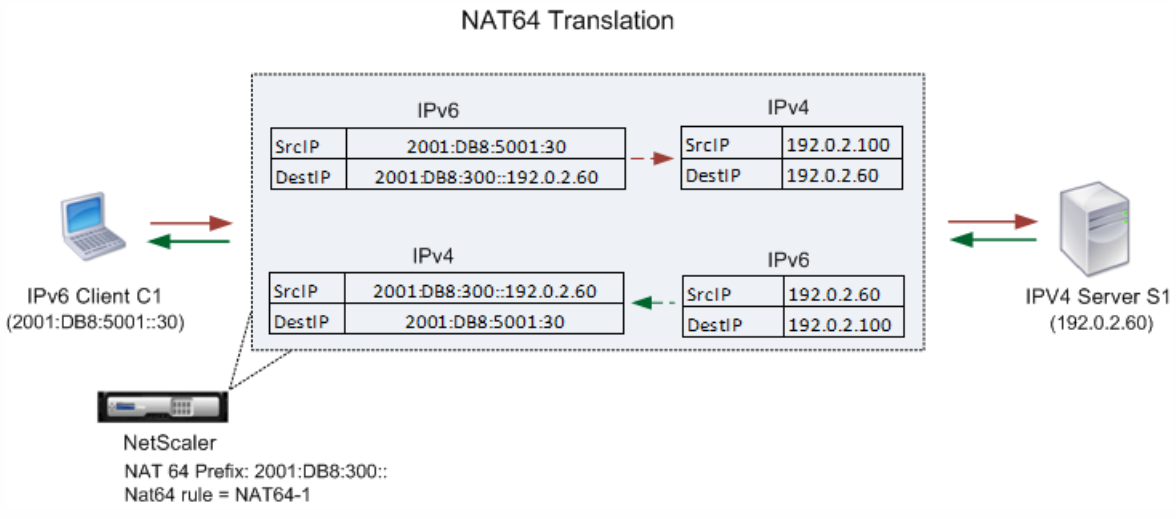
The NetScaler appliance considers an incoming IPv6 packet for NAT64 translation when all of the following conditions are met:

- The incoming IPv6 packet matches the ACL6 rule bound to a NAT64 rule.
- The destination IP address of the IPv6 packet matches the NAT64 IPv6 prefix.

When an IPv6 request packet received by the NetScaler appliance matches an ACL6 defined in a NAT64 rule and the destination IP of the packet matches the NAT64 IPv6 prefix, the NetScaler appliance considers the IPv6 packet for translation.

The appliance translates this IPv6 packet to an IPv4 packet with a source IP address matching one of the IP address bound to the netprofile defined in the NAT64 rule, and a destination IP address consisting of the last 32 bits of the destination IPv6 address of the IPv6 request packet. The NetScaler appliance creates a NAT64 session for this particular flow and forwards the packet to the IPv4 server. Subsequent responses from the IPv4 server and requests from the IPv6 client are translated accordingly by the appliance, on the basis of information in the particular NAT64 session.

Consider an example in which an enterprise hosts site www.example.com on server S1, which has an IPv4 address. To enable communication between IPv6 clients and IPv4 server S1, NetScaler appliance NS1 is deployed with a stateful NAT64 configuration that includes a NAT64 rule and a NAT64 prefix. A mapped IPv6 address of server S1 is formed by concatenating the NAT64 IPv6 prefix [96 bits] and the IPv4 source address [32 bits]. This mapped IPv6 address is then manually configured in the DNS servers. The IPv6 clients get the mapped IPv6 address from the DNS servers to communicate withIPv4 server S1.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following table lists the settings used in this example:

Entities	Name	Value
IPv6 address of client CL1	Client_IPv6 (for reference purposes only)	2001:DB8:5001::30
IPv4 address of server S1	Sevr_IPv4 (for reference purposes only)	192.0.2.60
IPv6 prefix for NAT64 translation	NAT64_Prefix (for reference purposes only)	2001:DB8:300::
Mapped IPv6 address (NAT64_Prefix + Sevr_IPv4) of server S1 for IPv6 clients to reach server S1	Map-Sevr-IPv6 (for reference purposes only)	2001:DB8:300::192.0.2.60
ACL6 rule	ACL6-1	• Action = ALLOW • Source IP address =2001:DB8:5001::30
IPset	IPset-1	IP addresses bound (of type SNIPs) = 192.0.2.100 and 192.0.2.102
Netprofile	Netprofile-1	Source IP address = IPset-1
NAT64 rule	NAT64-1	ACL6 rule = ACL6-1 Netprofile = Netprofile-1

Following is the traffic flow in this example:

1. IPv6 client CL1 sends a request packet to Map-Sevr-IPv6 (2001:DB8:300::192.0.2.60) address.
2. The NetScaler appliance receives the request packet. If the request packet matches the ACL6 defined in the NAT64 rule, and the destination IP address of the packet matches the NAT64 IPv6 prefix, the NetScaler considers the IPv6 packet for translation.
3. The appliance creates a translated IPv4 request packet with:
   - Destination IP address field containing the NAT64 prefix stripped from the destination address of the IPv6 request (Sevr_IPv4 = 192.0.2.60)
   - Source IP address field containing one of the IPv4 address bound to Netprofile-1(in this case, 192.0.2.100)
4. The NetScaler appliance creates a NAT64 session for this flow and sends the translated IPv4 request to server S1.

5. IPv64 server S1 responds by sending an IPv4 packet to the NetScaler appliance with:
   - Destination IP address field containing 192.0.2.100
   - Source IP address field containing the address ofSevr_IPv4(192.0.2.60)
6. The appliance receives the IPv4 response packet, searches all the session entries, and finds that the IPv6 response packet matches the NAT64 session entry created in step 4. The appliance considers the IPv4 packet for translation.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

7. The appliance creates a translated IPv6 response packet with:
   - Destination IP address field=Client_IPv6=2001:DB8:5001::30
   - Source IP address field = Concatenation of NAT64 Prefix (First 96 bits) and Sevr_IPv4 (last 32 bits) =2001:DB8:300::192.0.2.60
8. The appliance sends the translated IPv6 response to client CL1.

-

Limitations of Statelful NAT64

The following limitations apply to stateful NAT64 translation:

- Translation of IPv4 options is not supported.
- Translation of IPv6 routing headers is not supported.
- Translation of hop-by-hop extension headers of IPv6 packets is not supported.
- Translation of ESP and EH headers of IPv6 packets is not supported.
- Translation of multicast packets is not supported.
- Packets of Stream Control Transmission Protocol (SCTP), Datagram Congestion Control Protocol (DCCP), and IPSec, are not translated.

Configuring Stateful NAT64

Updated: 2013-10-31

Creating the required entities for stateful NAT64 configuration on the NetScaler appliance involves the following procedures:

1. Add an ACL6 rule with action ALLOW.
2. Add an ipset, which binds multiple IP addresses.
3. Add a netprofile and bind the ipset to it. If you want to bind only one IP address, you need not create an ipset entity. In that case, bind the IP address directly to the netprofile.
4. Add a NAT64 rule, which includes binding the ACl6 rule and the netprofile to the NAT 64 rule.
5. Add a NAT64 IPv6 prefix.

To add an ACL6 rule by using the command line interface

At the command prompt, type:

- add ns acl6 <acl6name> <acl6action> ...

To add an IPset and bind multiple IPs to it by using the command line interface

At the command prompt, type:

- add ipset <name>
- bind ipset <name> <IPaddress ...>

To add a netprofile by using the command line interface

At the command prompt, type:

- add netprofile <name> -srcIP <IPaddress or IPset>

To add a NAT64 rule by using the command line interface

At the command prompt, type:

- add nat64 <name> <acl6name> -netProfile <string>

To add a NAT64 prefix by using the command line interface

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

- set ipv6 -natprefix <ipv6_addr|*>

Example

```
> add acl6 ACL6-1 ALLOW -srcIPv6 2001:DB8:5001::30
 Done

> apply acls6
 Done

> add ip 192.0.2.100 255.255.255.0 –type SNIP
 Done

> add ip 192.0.2.102 255.255.255.0  –type SNIP
 Done

> add ipset IPset-1
 Done

> bind ipset IPset-1 192.0.2.100 192.0.2.102
IPAddress "192.0.2.100" bound
IPAddress "192.0.2.102" bound
 Done

> add netprofile Netprofile-1 -srcIP IPset-1
 Done

> add nat64 NAT64-1 ACL6-1 -netprofile Netprofile-1
 Done

> set ipv6 -natprefix 2001:DB8:300::/96
 Done
```

To add a NAT64 rule by using the configuration utility

Navigate to System > Network > Routes > NAT64, and a new NAT64 rule, or edit an existing rule.

To add a NAT64 prefix by using the configuration utility

Navigate to System > Network, in the Settings group, click Configure INAT Parameters, and set the Prefix parameter.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring RNAT

May 11, 2012

In Reverse Network Address Translation (RNAT), the NetScaler appliance replaces the source IP addresses in the packets generated by the servers with public NAT IP addresses. By default, the appliance uses a Mapped IP address (MIP) as the NAT IP address. You can also configure the appliance to use a unique NAT IP address for each subnet. You can also configure RNAT by using Access Control Lists (ACLs). Use Source IP (USIP), Use Subnet IP (USNIP), and Link Load Balancing (LLB) modes affect the operation of RNAT. You can display statistics to monitor RNAT.

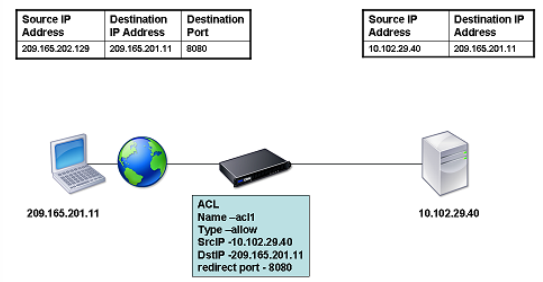
Note: The ephemeral port range for RNAT on the NetScaler appliance is 1024-65535.

You can use either a network address or an extended ACL as the condition for an RNAT entry:

- **Using a Network address.** When you use a network address, RNAT processing is performed on all of the packets coming from the specified network.
- **Using Extended ACLs.** When you use ACLs, RNAT processing is performed on all packets that match the ACLs. To configure the NetScaler appliance to use a unique IP address for traffic that matches an ACL, you must perform the following three tasks:
  1. Configure the ACL.
  2. Configure RNAT to change the source IP address and Destination Port.
  3. Apply the ACL.

  The following diagram illustrates RNAT configured with an ACL.

  Figure 1. RNAT with an ACL

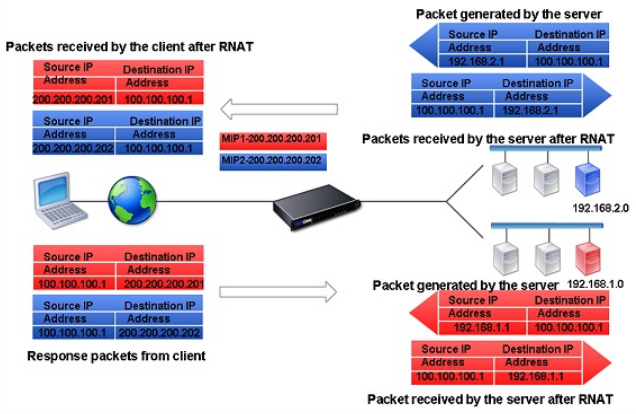


You have the following basic choices for the type of NAT IP address:

- **Using a MIP or SNIP as the NAT IP Address.** When using a MIP as the NAT IP address, the NetScaler appliance replaces the source IP addresses of server-generated packets with the a MIP. Therefore, the MIP address must be a public IP address. If Use Subnet IP (USNIP) mode is enabled, the NetScaler can use a subnet IP address (SNIP) as the NAT IP address.
- **Using a Unique IP Address as the NAT IP Address.** When using a unique IP address as the NAT IP address, the NetScaler appliance replaces the source IP addresses of server-generated packets with the unique IP address specified. The unique IP address must be a public NetScaler-owned IP address. If multiple NAT IP addresses are configured for a subnet, NAT IP selection uses the round robin algorithm.

  This configuration is illustrated in the following diagram.

  Figure 2. Using a Unique IP Address as the NAT IP Address



This section includes the following details:

- Creating an RNAT Entry
- Monitoring RNAT
- RNAT in USIP, USNIP, and LLB Modes
- Configuring RNAT for IPv6 Traffic

Creating an RNAT Entry

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2013-08-28

The following instructions provide separate command-line procedures for creating RNAT entries that use different conditions and different types of NAT IP addresses. In the configuration utility, all of the variations can be configured in the same dialog box, so there is only one procedure for configuration utility users.

To create an RNAT entry by using the command line interface

At the command prompt, type one the following commands to create, respectively, an RNAT entry that uses a network address as the condition and a MIP or SNIP as the NAT IP address, an RNAT entry that uses a network address as the condition and a unique IP address as the NAT IP address, an RNAT entry that uses an ACL as the condition and a MIP or SNIP as the NAT IP address, or an RNAT entry that uses an ACL as a condition and a unique IP address as the NAT IP address:

- set rnat <IPAddress> <netmask>
- set rnat IPAddress <netMask> -natip <NATIPAddress>
- set rnat <aclname> [-redirectPort <port>]
- set rnat <aclname> [-redirectPort <port>] -natIP <NATIPAddress>

Use the following command to verify the configuration:

- show rnat

Examples

A network address as the condition and a MIP or SNIP as the NAT IP address:

```
> set rnat 192.168.1.0 255.255.255.0
 Done
```

A network address as the condition and a unique IP address as the NAT IP address:

```
> set rnat 192.168.1.0 255.255.255.0 -natip 10.102.29.50
 Done
```

If instead of a single NAT IP address you specify a range, RNAT entries are created with all the NetScaler-owned IP addresses, except the NSIP, that fall within the range specified:

```
> set rnat 192.168.1.0 255.255.255.0  -natIP 10.102.29.[50-110]
 Done
```

An ACL as the condition and a MIP or SNIP as the NAT IP address:

```
> set rnat acl1
 Done
```

An ACL as a condition and a unique IP address as the NAT IP address:

```
> set rnat acl1  -natIP 209.165.202.129
Done
```

If instead of a single NAT IP address you specify a range, RNAT entries are created with all the NetScaler-owned IP addresses, except the NSIP, that fall within the range specified:

```
> set rnat acl1  -natIP 10.102.29.[50-70]
 Done
```

To create an RNAT entry by using the configuration utility

1. Navigate to System > Network > Routes > RNAT.
2. In the Action list, select Configure RNAT.

Monitoring RNAT

Updated: 2013-09-27

You can display RNAT statistics to troubleshoot issues related to IP address translation.

To view RNAT statistics by using the command line interface

At the command prompt, type:

stat rnat

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> stat rnat

RNAT summary
                    Rate (/s)        Total
Bytes Received          0              0
Bytes Sent              0              0
Packets Received        0              0
Packets Sent            0              0
Syn Sent                0              0
Current RNAT sessions   --             0
 Done
>
```

The following tables describes the statistics associated with RNAT and RNAT IP.

Table 1. RNAT Statistics

Statistic	Description
Bytes received	Bytes received during RNAT sessions
Bytes sent	Bytes sent during RNAT sessions
Packets received	Packets received during RNAT sessions
Packets sent	Packets sent during RNAT sessions
Syn sent	Requests for connections sent during RNAT sessions
Current sessions	Currently active RNAT sessions

To monitor RNAT by using the configuration utility

Navigate to System > Network > Routes > RNAT, and click Statistics.

RNAT in USIP, USNIP, and LLB Modes

Updated: 2013-12-18

Before configuring a RNAT rule, consider the following points:

- When RNAT and Use Source IP (USIP) are both configured on the NetScaler appliance, RNAT takes precedence. In other words, the source IP address of the packets, which matches a RNAT rule, is replaced according to the setting in the RNAT rule.
- When RNAT and Use SNIP (USNIP) are configured on the NetScaler appliance, selection of the source IP address is based on the state of USNIP, as follows:
  - If USNIP is off, the NetScaler appliance uses the mapped IP addresses.
  - If USNIP is on, the NetScaler uses a SNIP address as the NAT IP address.

This behavior does not apply when a unique NAT IP address is used.

In a topology where the NetScaler appliance performs both Link Load Balancing (LLB) and RNAT for traffic originating from the server, the appliance selects the source IP address based on the router. The LLB configuration determines selection of the router. For more information about LLB, see "Link Load Balancing."

Configuring RNAT for IPv6 Traffic

Updated: 2013-10-31

Reverse Network Address Translation (RNAT) rules for IPv6 packets are called RNAT6s. When an IPv6 packet generated by a server matches the conditions specified in the RNAT6 rule, the appliance replaces the source IPv6 address of the IPv6 packet with a configured NAT IPv6 address before forwarding it to the destination. The NAT IPv6 address is one of the NetScaler owned SNIP6 or VIP6 addresses.

When configuring an RNAT6 rule, you can specify either an IPv6 prefix or an ACL6 as the condition:

- **Using a IPv6 network address**. When you use an IPv6 prefix, the appliance performs RNAT processing on those IPv6 packets whose IPv6 address matches the prefix.
- **Using ACL6s**. When you use an ACL6, the appliance performs RNAT processing on those IPv6 packets that match the conditions specified in the ACL6.

You have one of the following options to set the NAT IP address:

- Specify a set of NetScaler owned SNIP6 and VIP6 addresses for an RNAT6 rule. The NetScaler appliance uses any one of the IPv6 addresses from this set as a NAT IP address for each session. The selection is based on the round robin algorithm and is done for each session.
- Do not specify any NetScaler owned SNIP6 or VIP6 address for an RNAT6 rule. The NetScaler appliance uses any one of the NetScaler owned SNIP6 or VIP6 addresses as a NAT IP address. The selection is based on the next hop network to which an IPv6 packet that matches the RNAT rule is destined.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To create an RNAT6 rule by using the command line interface

At the command prompt, to create the rule and verify the configuration, type:

- add rnat6 <name> (<network> | (<acl6name> [-redirectPort <port>]))
- bind rnat6 <name> <natIP6>@ ...
- show rnat6

To modify or remove an RNAT6 rule by using the command line interface

- To modify an RNAT6 rule whose condition is an ACL6, type the set rnat6 <name> command, followed by a new value for the redirectPort parameter.
- To remove an RNAT6 rule, type the clear rnat6 <name> command.
- show rnat6

To configure an RNAT6 rule by using the configuration utility

Navigate to System > Network > Routes > RNAT6, and add a new RNAT6 rule, or edit an existing rule.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Prefix-Based IPv6-IPv4 Translation

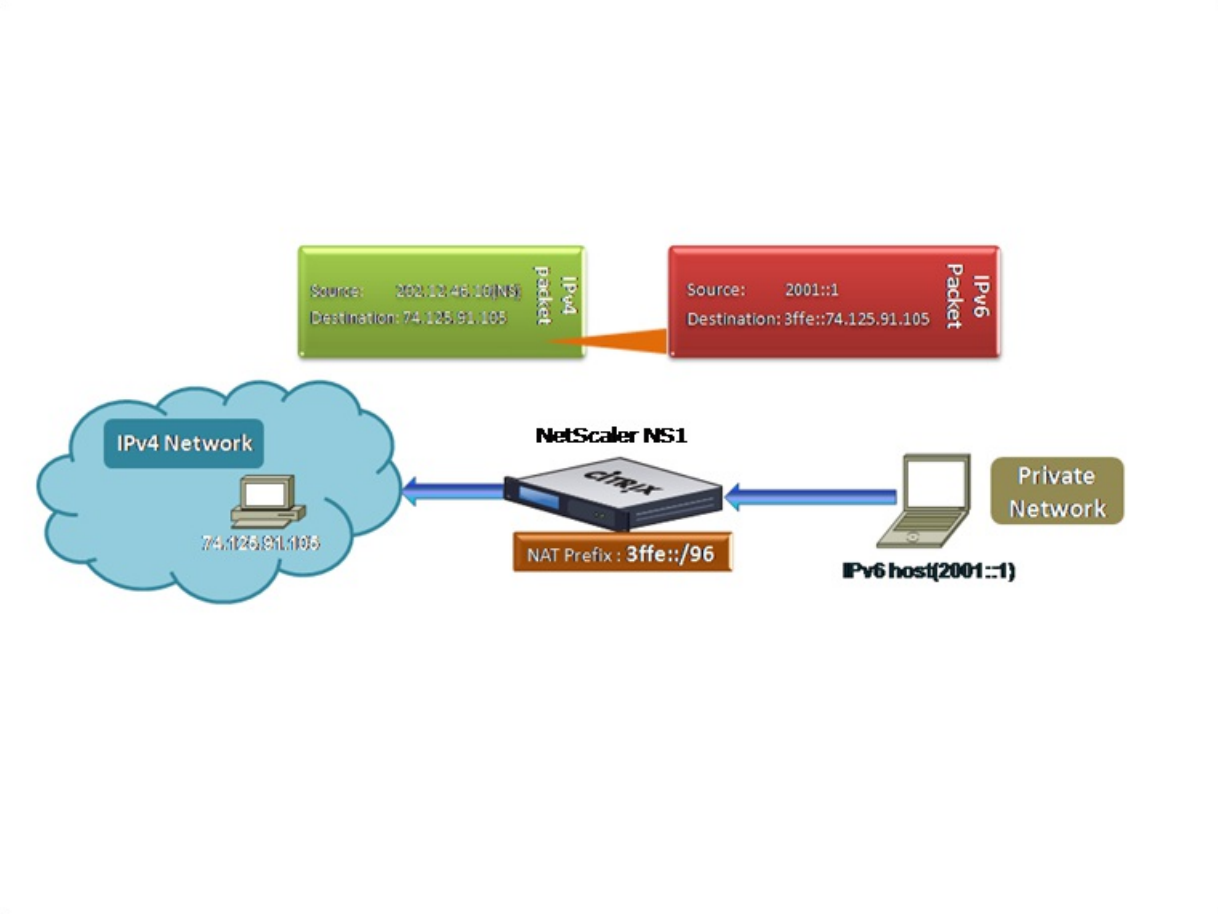
Aug 28, 2013

Prefix-based translation is a process of translating packets sent from private IPv6 servers into IPv4 packets, using an IPv6 prefix configured in the NetScaler appliance. This prefix has a length of 96 bits (128-32=96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix.

The NetScaler appliance compares the first 96 bits of the destination IP address of all the incoming IPv6 packets to the configured prefix. If there is a match, the NetScaler appliance generates an IPv4 packet and sets the destination IP address as the last 32 bits of the destination IP address of the matched IPv6 packet. IPv6 packets addressed to this prefix have to be routed to the NetScaler so that the IPv6-IPv4 translation is done by the NetScaler.

In the following diagram, 3ffe::/96 is configured as the IPv6 NAT prefix on NetScaler NS1. The IPv6 host sends an IPv6 packet with destination IP address 3ffe::74.125.91.105. NS1 compares the first 96 bits of the destination IP address of all the incoming IPv6 packets to the configured prefix, and they match. NS1 then generates an IPv4 packet and sets the destination IP address as 74.125.91.105.

Figure 1. IPv6-IPv4 Prefix-Based Translation



To configure prefix-based IPv6-IPv4 translation by using the command line interface

At the command prompt, type the following commands to set a NAT prefix and verify its configuration:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- set ipv6 [-natprefix <ipv6_addr|*>]
- show ipv6

Example

```
> set ipv6 -natprefix 3ffe::/96
 Done
```

To configure prefix-based IPv6-IPv4 translation by using the configuration utility

Navigate to System > Network, in the Settings group, click Configure INAT Parameters, and set the Prefix parameter.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

IP Prefix NAT

Jan 04, 2016

The NetScaler appliance supports translating a part of the source IP address instead of the complete address of packets received on the appliance. IP prefix NAT includes changing one or more octets or bits of the source IP address.

The NetScaler appliance supports IP prefix NAT for traffic related to virtual servers and services for which the NetScaler does not maintain any session information. For example, virtual servers and services of type ANY, UDP, and DNS.

Use Case: Zonification of Clients for a Deployment of a NetScaler appliance and an Optimization Device

IP prefix NAT is very useful in a deployment that includes a NetScaler appliance and an optimization device (for example, Citrix ByteMobile). This type of deployment has different geographically located client networks, which share the same network address. The NetScaler appliance must send the traffic received from each of the client networks to the optimization device before forwarding to the destination.

The device sends the optimized traffic back to the NetScaler appliance. Because the optimization requirement is different for traffic from each client network, the optimization device must recognize the client network of each packet that it receives. The solution is to segregate traffic from each client network into a different zone by using VLANs. IP prefix NAT with a different setting is configured for each zone. The NetScaler appliance translates the last octet of the source IP address of every packet, and the translated octet value is different for each zone.

Consider an example of two zones, Z1 and Z2, sharing network address 192.0.2.0/24. On the NetScaler appliance, IP prefix NAT entities named natrule-1 and natrule-2 are configured for these two zones. Before the appliance forwards a packet from Z1, natrule-1 translates the last octet of the packet's source IP address to 100. Similarly, for packets from Z2, natrule-2 translates the last octet of the source IP address to 200. For two clients, CL1-Z1 in zone Z1 and CL1-Z2 in zone Z2, each with IP address 192.0.2.30, the NetScaler appliance translates the source IP address of CL1-Z1's packets to 100.0.2.30 and of CL1-Z2's packets to 200.0.2.30. The optimization device to which the NetScaler appliance sends the translated packets is configured to use a packet's source IP address to recognize the zone, so it applies the appropriate optimization configured for the zone from which the packet originated.

Configuration Steps

Configuring IP prefix NAT consists of the following steps:

Create a net profile and set the NAT Rule parameter of a net profile. A NAT rule specifies two IP addresses and a net mask. The first IP address (specified by IP Address parameter) is the source IP address that is to be translated with the second one (specified by IP Rewrite parameter). The net mask specifies the part of the source IP address that is to be translated with the same part of the second IP address.

Bind the net profile to load balancing virtual servers or services. A net profile with NAT rule setting can be bound to a virtual server or service of only type ANY, UDP, or DNS. After binding a net profile to a virtual server or service, the NetScaler appliance matches the source IP address of the incoming packets related to the virtual server or service with the NAT rule setting. The NetScaler then performs IP prefix NAT for packets that match the NAT rule.

To configure IP prefix NAT translation by using the command line

At the command prompt, type:

- **bind netProfile** <name> (-**natRule** <ip_addr> <netmask> <rewriteIp>)
- **show netprofile** <name>

In the following sample configuration, net profile PARTIAL-NAT-1 has IP prefix NAT settings and is bound to load balancing virtual server LBVS-1, which is of type ANY. For packets received on LBVS-1 from 192.0.0.0/8, the NetScaler appliance translates the last octet of the packet's source IP address to 100. For example, a packet with source IP address 192.0.2.30 received on LBVS-1, the NetScaler appliance translates the source IP address to 100.0.2.30 before sending it one of the bound servers.

```
> add netprofile PARTIAL-NAT-1

Done

> bind netprofile PARTIAL-NAT-1 –natrule 192.0.0.0 255.0.0.0 100.0.0.0

Done

> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done
```

To configure IP prefix NAT by using the configuration utility

1. Navigate to **System** > **Network** > **Net Profiles**.

2. Set the following parameters under NAT Rules while adding or modifying NetProfiles.

- IP Address
- Netmask
- Rewrite IP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Static ARP

Aug 28, 2013

You can add static ARP entries to and remove static ARP entries from the ARP table. After adding an entry, you should verify the configuration. If the IP address, port, or MAC address changes after you create a static ARP entry, you must remove or manually adjust the static entry. Therefore, creating static ARP entries is not recommended unless necessary.

To add a static ARP entry by using the command line interface

At the command prompt, type:

- add arp -IPAddress <ip_addr> -mac<mac_addr> -ifnum <interface_name>
- show arp <IPAddress>

Example

```
> add arp -ip 10.102.29.6 -mac 00:24:e8:73:ca:ec -ifnum 1/1
 Done
```

To remove a static ARP entry by using the command line interface

At the command prompt, type the rm arp command and the IP address.

To add a static ARP entry by using the configuration utility

Navigate to System > Network > ARP Table, and a new ARP entry.

Specifying a VLAN in a Static ARP Entry

In a static ARP entry, you can specify the VLAN through which the destination device is accessible. This feature is useful when the interface specified in the static ARP entry is part of multiple tagged VLANs and the destination is accessible through one of the VLANs. The NetScaler appliance includes the specified VLAN ID in the outgoing packets matching the static ARP entry. If you don't specify a VLAN ID in an ARP entry, and the specified interface is part of multiple tagged VLANs, the appliance assigns the interface's native VLAN to the ARP entry.

For example, say NetScaler interface 1/2 is part of native VLAN 2 and of tagged VLANs 3 and 4, and you add a static ARP entry for network device A, which is part of VLAN 3 and is accessible through interface 1/2. You must specify VLAN 3 in the ARP entry for network device A. The NetScaler appliance then includes tagged VLAN 3 in all the packets destined to network device A, and sends them from interface 1/2.

If you don't specify a VLAN ID, the NetScaler appliance assigns native VLAN 2 for the ARP entry. Packets destined to device A are dropped in the network path, because they do not specify tagged VLAN 3, which is the VLAN for device A.

To specify a VLAN in a Static ARP entry by using the command line interface

At the command prompt, type:

1. **add arp -IPAddress** <ip_addr> **-mac**<mac_addr> **-ifnum** <interface_name> [-**vlan** <positive_integer>]
2. **show arp** <IPAddress>

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> add arp -ip 198.51.100.91 -mac 36:db:4b:f6:12:15 -ifnum 1/2 -vlan 3
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting the Timeout for Dynamic ARP Entries

Aug 28, 2013

You can globally set an aging time (time-out value) for dynamically learned ARP entries. The new value applies only to ARP entries that are dynamically learned after the new value is set. Previously existing ARP entries expire after the previously configured aging time.

You can specify an ARP time-out value of from 1 through 1200 seconds.

To set the time-out for dynamic ARP entries by using the command line interface

At the command prompt, type the following commands to set the time-out for dynamic ARP entries and verify its configuration:

- set arpparam -timeout <positive_integer>]
- show arpparam

Example

```
> set arpparam -timeout 500
 Done
```

To set the time-out for dynamic ARP entries to its default value by using the command line interface

At the command prompt, type the following commands to set the time-out for dynamic ARP entries to its default value and verify its configuration:

- unset arpparam
- show arpparam

Example

```
> unset arpparam
 Done
```

To set the time-out for dynamic ARP entries by using the configuration utility

Navigate to System > Network, in the Settings group, click Configure ARP Global Parameters, and set the ARP Table Entry Timeout parameter.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Neighbor Discovery

May 10, 2012

Neighbor discovery (ND) is one of the most important protocols of IPv6. It is a message-based protocol that combines the functionality of the Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), and Router Discovery. ND allows nodes to advertise their link layer addresses and obtain the MAC addresses or link layer addresses of the neighboring nodes. This process is performed by the Neighbor Discovery protocol (ND6).

Neighbor discovery can perform the following functions:

Router Discovery

Enables a host to discover the local routers on an attached link and automatically configure a default router.

Prefix Discovery

Enables the host to discover the network prefixes for local destinations.

Note: Currently, the NetScaler does not support Prefix Discovery.

Parameter Discovery

Enables a host to discover additional operating parameters, such as MTU and the default hop limit for outbound traffic.

Address Autoconfiguration

Enables hosts to automatically configure IP addresses for interfaces both with and without stateful address configuration services such as DHCPv6. The NetScaler does not support Address Autoconfiguration for Global IPv6 addresses.

Address Resolution

Equivalent to ARP in IPv4, enables a node to resolve a neighboring node's IPv6 address to its link-layer address.

Neighbor Unreachability Detection

Enables a node to determine the reachability state of a neighbor.

Duplicate Address Detection

Enables a node to determine whether an NSIP address is already in use by a neighboring node.

Redirect

Equivalent to the IPv4 ICMP Redirect message, enables a router to redirect the host to a better first-hop IPv6 address to reach a destination.

Note: The NetScaler does not support IPv6 Redirect.

To enable neighbor discovery, you create entries for the neighbors.

This section includes the following details:

- Adding IPv6 Neighbors
- Removing IPv6 Neighbors

Adding IPv6 Neighbors

Updated: 2013-08-28

Adding IPv6 neighbors enables neighbor discovery.

To add an IPv6 neighbor by using the command line interface

At the command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- add nd6 <neighbor> <mac> <ifnum> [-vlan <integer>]
- show nd6

Example

```
> add nd6 2001::1 00:04:23:be:3c:06 1/1 –vlan 1
 Done
> show nd6
  Neighbor              MAC-Address(Vlan, Interface)      State       TIME
  --------              ----------------------------      -----      --------
1) ::1                   00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE     PERMANENT
2) fe80::2d0:68ff:fe0b:58da 00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE     PERMANENT
3) 2001::1               00:04:23:be:3c:06(   1,   1/1)   REACHABLE      STATIC
 Done
```

To add an IPv6 neighbor by using the configuration utility

Navigate to System > Network > IPv6 Neighbors, and add a new IPv6 neighbor.

Removing IPv6 Neighbors

Updated: 2013-08-28

To remove a neighbor discovery entry by using the command line interface

At the command prompt, type:

rm nd6 <Neighbor> -vlan <VLANID>

Example

```
 rm nd6 3ffe:100:100::1 -vlan 1
```

To remove all neighbor discovery entries by using the command line interface

At the command prompt, type:

clear nd6

To remove a neighbor discovery entry by using the configuration utility

Navigate to System > Network > IPv6 Neighbors, delete the IPv6 neighbor.

To remove all neighbor discovery entries by using the configuration utility

Navigate to System > Network > IPv6 Neighbors, and click Clear.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring IP Tunnels

Mar 20, 2012

An IP Tunnel is a communication channel, that can be created by using encapsulation technologies, between two networks that do not have a routing path. Every IP packet that is shared between the two networks is encapsulated within another packet and then sent via the tunnel.

The NetScaler appliance implements IP Tunneling in the following ways:

- NetScaler as an Encapsulator (Load Balancing with DSR mode)
- NetScaler as a Decapsulator

NetScaler as an Encapsulator (Load Balancing with DSR Mode)

Consider an organization that has multiple data centers across different countries, where the NetScaler maybe at one location and the back-end servers are located in a different country. In essence, the NetScaler and the back-end servers are on different networks and are connected via a router.

When you configure Direct Server Return (DSR) on this NetScaler, the packet sent from the source subnet is encapsulated by the NetScaler and sent via a router and tunnel to the appropriate back-end server. The back-end server decapsulates the packet and responds directly to the client, without allowing the packet to pass via the NetScaler.

NetScaler as a Decapsulator

Consider an organization having multiple data centers each having NetScalers and back-end servers. When a packet is sent from data center A to data center B it is usually sent via an intermediary, say a router or another NetScaler. The NetScaler processes the packet and then forwards the packet to the back-end server. However, if an encapsulated packet is sent, the NetScaler must be able to decapsulate the packet before sending it to the back-end servers. To enable the NetScaler to function as a decapsulator, a tunnel is added between the router and the NetScaler. When the encapsulated packet, with additional header information, reaches the NetScaler, the data packet is decapsulated i.e. the additional header information is removed, and the packet is then forwarded to the appropriate back-end servers.

The NetScaler can also be used as a decapsulator for the Load Balancing feature, specifically in scenarios when the number of connections on a vserver exceeds a threshold value and all the new connections are then diverted to a back-up vserver.

This section includes the following details:

- Creating IP Tunnels
- Customizing IP Tunnels Globally
- GRE Payload Options in a GRE Tunnel
- IPv6 Traffic through GRE IPV4 Tunnels

Creating IP Tunnels

To create an IP tunnel by using the command line interface

At the command prompt type:

- add iptunnel <name> <remote> <remoteSubnetMask> <local> -type -protocol (ipoverip | GRE) -ipsecprofile <name>
- show iptunnel

Note: While configuring an IP tunnel in a cluster setup, the local IP address must be a striped SNIP or MIP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To remove an IP tunnel by using the command line interface

To remove an IP tunnel, type the rm iptunnel command and the name of the tunnel.

To create an IP Tunnel by using the configuration utility

Navigate to System > Network > IP Tunnels, add a new IP tunnel.

To create an IPv6 tunnel by using the command line interface

At the command prompt type:

- add ip6tunnel <name> <remoteIp> <local>
- show ip6tunnel

To remove an IPv6 tunnel by using the command line interface

To remove an IPv6 tunnel, type the rm ip6tunnel command and the name of the tunnel.

To create an IPv6 Tunnel by using the configuration utility

Navigate to System > Network > IP Tunnels > IPv6 Tunnels, and add a new IPv6 tunnel.

Customizing IP Tunnels Globally

Updated: 2013-10-31

By globally specifying the source IP address, you can assign a common source IP address across all tunnels. Also, because fragmentation is CPU-intensive, you can globally specify that the NetScaler appliance drop any packet that requires fragmentation. Alternatively, if you would like to fragment all packets as long as a CPU threshold value is not reached, you can globally specify the CPU threshold value.

To globally customize IP tunnels by using the command line interface

At the command prompt, type the following commands to globally customize IP tunnels and verify the configuration:

- set ipTunnelParam -srcIP <sourceIPAddress> -srcIPRoundRobin (YES | NO)-dropFrag [YES | NO] -dropFragCpuThreshold <Positive integer>
- show ipTunnelParam

Example

```
> set iptunnelparam –srcIP 12.12.12.22 -dropFrag Yes –dropFragCpuThreshold 50
 Done
> set iptunnelparam -srcIPRoundRobin YES -dropFrag Yes –dropFragCpuThreshold 50
 Done
```

Note: To create a new MIP or SNIP address to use as the global source IP address, use the add ns ip command before you type the set iptunnelparam command.

To globally customize IP tunnels by using the configuration utility

Navigate to System > Network, in the Settings group, click IPv4 Tunnel Global Settings.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to System > Network.
2. In the details pane, in the Settings group, click IPv4 Tunnel Global Settings.
3. In the Configure IP Tunnel Global Parameters dialog box, set the parameters. For a description of a parameter, hover the mouse cursor over the corresponding field.
4. Click OK and then click Close.

To globally customize IPv6 tunnels by using the command line interface

At the command prompt, type the following commands to globally customize IPv6 tunnels and verify the configuration:

- set ip6tunnelparam -srcIP <IPv6Address> -srcIPRoundRobin (YES | NO)-dropFrag [YES | NO] -dropFragCpuThreshold <Positive integer>
- show ip6tunnelparam

Note: To create a new VIP6 or SNIP6 address to use as the global source IP address, use the add ns ip6 command before you type the set ip6tunnelparam command.

To globally customize IPv6 tunnels by using the configuration utility

Navigate to System > Network, in the Settings group, click IPv6 Tunnel Global Settings.

GRE Payload Options in a GRE IP Tunnel

For a configured GRE IP tunnel, the NetScaler appliance encapsulates the entire Layer 2 packet, including the Ethernet header and the VLAN header (dot1q VLAN tag). IP GRE tunnels between NetScaler appliances and some 3rd party devices might not be stable, because these 3rd party devices are not programmed to process some or the Layer 2 packet headers. To configure a stable IP GRE tunnel between a NetScaler appliance and a 3rd party device, you can use the GRE payload parameter of the GRE IP tunnel command set. GRE payload setting can also be applied to a GRE with IPSec tunnel.

You can set the GRE payload parameter to do one of the following before the packet is sent through the GRE tunnel:

Ethernet with DOT1Q. Carry the Ethernet header as well the VLAN header. This is the default setting. For a tunnel bound to a netbridge, inner Ethernet header and VLAN header contains information from the ARP and bridge table of the NetScaler appliance. For a tunnel set as a next hop to a PBR rule, Inner Ethernet destination MAC address is set to zero and the VLAN header specifies the default VLAN. The encapsulated (GRE) packet sent from the NetScaler tunnel end point has the following format:

Outer Ethernet Header	Outer IP Header	GRE Header	Inner Ethernet	Inner VLAN header	Inner IP/IPv6/ARP header	Inner TCP/UDP Header	Payload

Ethernet. Carry the Ethernet header but drop the VLAN header. Because the packets do not carry any VLAN information in the tunnel, for a tunnel with this setting and bound to a netbridge, you must bind an appropriate VLAN to the netbridge so that on receiving any packets on the tunnel, the NetScaler can forward these packet to the specified VLAN. If the tunnel is set as a next hop in a PBR rule, the NetScaler routes the packets that are received on the tunnel. The encapsulated (GRE) packet sent from the NetScaler tunnel end point has the following format:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Outer Ethernet header	Outer IP header	GRE Header	Inner Ethernet header	Inner IP/IPv6/ARP header	Inner TCP/UDP header	Payload

IP. Drop the Ethernet header as well as the VLAN header. Because tunnels with this setting do not carry Layer 2 headers, these tunnels cannot be bound to a netbridge but can be set as a next hop in a PBR rule. The peer tunnel endpoint device on receiving the packet either consumes or routes it. The encapsulated (GRE) packet sent from the NetScaler tunnel end point has the following format:

Outer Ethernet header	**Outer IP header**	**GRE header**	**Inner IP/IPv6 header**	**Inner TCP/UDP header**	**Payload**

To drop Layer 2 headers of packets in a GRE IP tunnel by using the NetScaler command line

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> [-**protocol** <GRE> [-**vlan** <positive_integer>]] [-**grepayload** <grepayload>] [-**ipsecProfileName** <string>]
- **show iptunnel** <tunnelname>

Example COPY

```
> add iptunnel IPTUNNEL-1 203.0.113.133 255.255.255.0 198.51.100.15 –protocol GRE –grepayload Ethernet -ipsecProfileName IPTUN

Done
```

IPv6 Traffic through GRE IPV4 Tunnels

The NetScaler appliance supports transferring IPv6 traffic through an IPV4 GRE tunnel. This feature can be used for enabling communication between Isolated IPv6 networks without upgrading the IPv4 infrastructure between them.

For configuring this feature, you associate a PBR6 rule with the configured IPv4 GRE tunnel through which you want the NetScaler to send and receive IPv6 traffic. The source IPv6 address and destination IPv6 address parameters of the PBR6 rule specify the IPv6 networks whose traffic is to traverse the IPv4 GRE tunnel.

Note: IPSec protocol is not supported on GRE IPv4 tunnels that are configured to transfer IPv6 packets.

To create a GRE IPv4 tunnel by using the NetScaler command line

At the command prompt, type:

- add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol GRE
- show ipTunnel <name>

To associate a PBR6 rule with a GRE IPv4 tunnel by using the NetScaler command line

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- add ns pbr6 <pbrName> ALLOW –srcIPv6 <network-range> -dstIPv6 <network-range> -ipTunnel <tunnelName>
- show pbr

Example

In the following sample configuration, GRE IP tunnel TUNNEL-V6onV4 is created with remote tunnel endpoint IP address 10.10.6.30 and local tunnel endpoint IP address 10.10.5.30. The tunnel is then bound to pbr6 PBR6-V6onV4. The srcIPv6 specifies the IPv6 network connected to the local endpoint and destIPv6 specifies the IPv6 network connected to the remote endpoint. The traffic from these Ipv6 networks are allowed to traverse through the GRE IPv4 tunnel.

```
> add ipTunnel TUNNEL-V6onV4 10.10.6.30 255.255.255.255 10.10.5.30 -protocol GRE
-ipsecProfileName None
Done
> add ns pbr6 PBR6-V6onV4 ALLOW -srcIPv6 = 2001:0db8:1::1-2001:0db8:1::255 -destIPv6 =
1-2001:0db8:4::255 -ipTunnel TUNNEL-V6onV4
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Interfaces

Nov 13, 2015

Before you begin configuring interfaces, decide whether your configuration can use MAC-based forwarding mode, and either enable or disable this system setting accordingly. The number of interfaces in your configuration is different for the different models of the Citrix NetScaler appliance. In addition to configuring individual interfaces, you can logically group interfaces, using VLANs to restrict data flow within a set of interfaces, and you can aggregate links into channels. In a high availability setup, you can configure a virtual MAC (VMAC) address if necessary. If you use L2 mode, you might want to modify the aging of the bridge table.

When your configuration is complete, decide whether you should enable the system setting for path MTU discovery. NetScaler appliances can be deployed in active-active mode using VRRP. An active-active deployment, in addition to preventing downtime, makes efficient use of all the NetScaler appliances in the deployment. You can use the Network Visualizer tool to view the network configuration of a NetScaler deployment and configure interfaces, channels, VLANs, and bridge groups.

This document includes the following information:

- Configuring MAC-Based Forwarding
- Configuring Network Interfaces
- Configuring Forwarding Session Rules
- Understanding VLANs
- Configuring a VLAN
- Configuring NSVLAN
- Configuring Bridge Groups
- Configuring VMACs
- Configuring Link Aggregation
- Redundant Interface Set
- Binding an SNIP address to an Interface
- Monitoring the Bridge Table and Changing the Aging time
- Understanding NetScaler Appliances in Active-Active Mode Using VRRP
- Configuring Active-Active Mode
- Using the Network Visualizer
- Configuring Link Layer Discovery Protocol
- Jumbo Frames

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring MAC-Based Forwarding

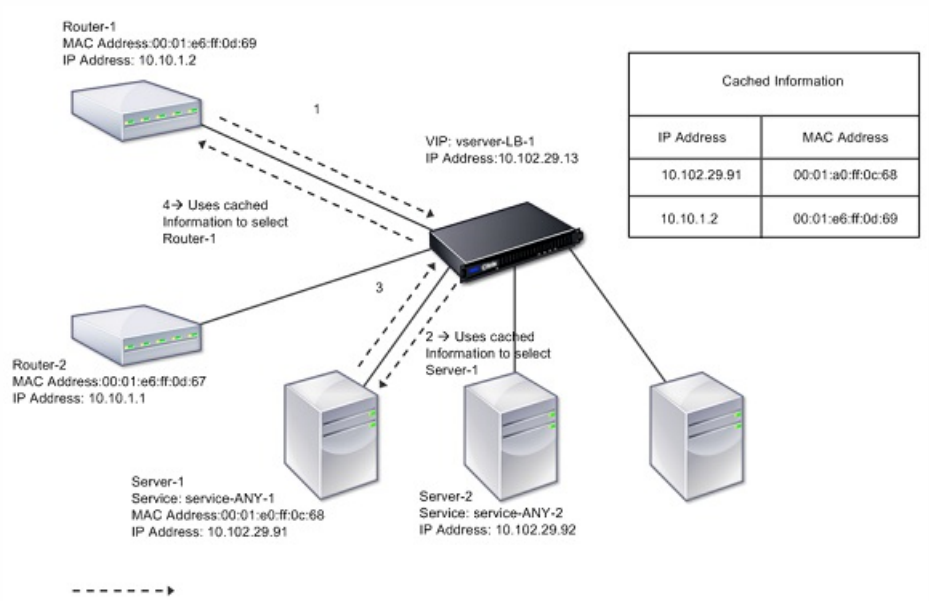
Oct 31, 2013

With MAC-based forwarding (MBF) enabled, when a request reaches the NetScaler appliance, the appliance remembers the source MAC address of the frame and uses it as the destination MAC address for the resulting replies. MAC-based forwarding can be used to avoid multiple-route/ARP lookups and to avoid asymmetrical packet flows. MAC-based forwarding may be required when the NetScaler is connected to multiple stateful devices, such as VPNs or firewalls, because it ensures that the return traffic is sent to the same device that the initial traffic came from.

MAC-based forwarding is useful when you use VPN devices, because it guarantees that all traffic flowing through a VPN passes back through the same VPN device.

The following topology diagram illustrates the process of MAC-based forwarding.

Figure 1. MAC-Based Forwarding Mode



When MAC-based forwarding (MBF) is enabled, the NetScaler caches the MAC address of:

- The source (a transmitting device such as router, firewall, or VPN device) of the inbound connection.
- The server that responds to the requests.

When a server replies through the NetScaler appliance, the appliance sets the destination MAC address of the response packet to the cached address, ensuring that the traffic flows in a symmetric manner, and then forwards the response to the client. The process bypasses the route table lookup and ARP lookup functions. However, when the NetScaler initiates a connection, it uses the route and ARP tables for the lookup function. In a direct server return configuration, you must enable MAC-based forwarding.

For more information about direct server return configurations, see "Load Balancing."

Some deployment topologies may require the incoming and outgoing paths to flow through different routers. MAC-based forwarding would break this topology design.

MBF should be disabled in the following situations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **When you configure link load balancing.** In this case, asymmetric traffic flows are desirable because of link costs.
- **When a server uses network interface card (NIC) teaming without using LACP (802.1ad Link Aggregation).** To enable MAC-based forwarding in this situation, you must use a layer 3 device between the NetScaler and server.
  Note: MBF can be enabled when the server uses NIC teaming with LACP, because the virtual interface uses one MAC address.
- When firewall clustering is used. Firewall clustering assumes that ARP is used to resolve the MAC address for inbound traffic. Sometimes the inbound MAC address can be a non-clustered MAC address and should not be used for inbound packet processing.

When MBF is disabled, the NetScaler uses L2 or L3 connectivity to forward the responses from servers to the clients. Depending on the route table, the routers used for outgoing connection and incoming connection can be different. In the case of reverse traffic (response from the server):

- If the source and destination are on different IP subnets, the NetScaler uses the route lookup to locate the destination.
- If the source is on the same subnet as the destination, the NetScaler looks up the ARP table to locate the network interface and forwards the traffic to it. If the ARP table does not exist, the NetScaler requests the ARP entries.

To enable or disable MAC-based forwarding by using the command line interface

At the command prompt, type:

- enable ns mode MBF
- disable ns mode MBF

To enable or disable MAC-based forwarding by using the configuration utility

1. Navigate to System > Settings, in the Modes and Features group, click Configure modes.
2. Select or clear the MAC-based forwarding option.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Network Interfaces

Nov 13, 2015

Network interfaces in the NetScaler appliance are numbered in <slot>/<port> notation. After configuring your interfaces, you should display the interfaces and their settings to verify the configuration. You can also display this information to troubleshoot a problem in the configuration.

To manage the network interfaces, you might have to enable some interfaces and disable others. You can reset an interface to renegotiate its settings. You can clear the accumulated statistics for an interface. To verify the configuration, you can display the interface settings. You can display the statistics for an interface to evaluate its health.

Setting the Network Interface Parameters

Updated: 2013-09-06

The network interface configuration is neither synchronized nor propagated. For an HA pair, you must perform the configuration on each unit independently.

To set the network interface parameters by using the command line interface

At the command prompt, type:

- set interface <id> [-speed <speed>] [-duplex <duplex>] [-flowControl <flowControl>] [-autoneg (DISABLED | ENABLED)] [-haMonitor (ON | OFF)] [(ON | OFF)] [-tagall (ON | OFF)] [-lacpMode <lacpMode>] [-lacpKey<positive_integer>] [-lacpPriority <positive_integer>] [-lacpTimeout (LONG | SHORT)] [-ifAlias <string>] [-throughput <positive_integer>][-bandwidthHigh <positive_integer> [-bandwidthNormal <positive_integer>]]
- show interface [<id>]

Example

```
> set interface 1/8 -duplex full
 Done
```

To set the network interface parameters by using the configuration utility

Navigate to System > Network > Interfaces, select the network interface that you want to modify (for example, 1/8), click Edit, and then set the parameters.

Enabling and Disabling Network Interfaces

Updated: 2013-08-29

By default, the network interfaces are enabled. You must disable any network interface that is not connected to the network, so that it cannot send or receive packets. Disabling a network interface that is connected to the network in a high availability setup can cause failover.

For more information about high availability, see "."

To enable or disable a network interface by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type one of the following pairs of commands to enable or disable an interface and verify the setting:

- enable interface <interface_num>
- show interface <interface_num>
- disable interface <interface_num>
- show interface <interface_num>

Example

```
> enable interface 1/8
 Done
> show interface 1/8
      Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
      flags=0x4004000 <ENABLED, DOWN, BOUND to LA/1, down, autoneg, 802.1q>
      MTU=1514, MAC=00:d0:68:15:fd:3d, downtime 906h58m40s
      Requested: media UTP, speed AUTO, duplex FULL, fctl OFF, throughput 0
      RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
      TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
      NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
      Bandwidth thresholds are not set.
Done
```

To enable or disable a network interface by using the configuration utility

1. Navigate to System > Network > Interfaces.
2. Select the network interface and, in the Action list, select Enable or Disable.

Resetting Network Interfaces

Updated: 2013-09-30

Network interface settings control properties such as duplex and speed. To renegotiate the settings of a network interface, you must reset it.

To reset a network interface by using the command line interface

At the command prompt, type the following commands to reset an interface and verify the setting:

- reset interface <interface_num>
- show interface <interface_num>

Example

```
> reset interface 1/8
 Done
```

To reset a network interface by using the configuration utility

1. Navigate to System > Network > Interfaces.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Select the network interface and, in the Action list, select Reset Interface.

Monitoring a Network Interface

Updated: 2013-08-29

You can display network interface statistics to monitor parameters such as packets sent and packets received, throughput, Link Aggregate Control Protocol (LACP) data units, and errors, and use the information to check the health of the network interface. You can clear the statistics of a network interface to monitor its statistics from the time the statistics are cleared.

To display the statistics of the network interfaces by using the command line interface

At the command prompt, type:

stat interface <interface_num>

To display the statistics of an Interface by using the configuration utility

Navigate to System > Network > Interfaces, select the network interface, and click Interface Statistics.

To clear a network interface's statistics by using the command line interface

At the command prompt, type:

clear interface <interface_num>

Example

```
> clear interface 1/8
 Done
```

To clear a network interface's statistics by using the configuration utility

1. Navigate to System > Network > Interfaces.
2. Select the network interface and, in the Action list, select Clear Statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Forwarding Session Rules

Jun 11, 2014

By default, the NetScaler appliance does not create session entries for traffic that it only forwards (L3 mode). For a case in which a client request that the appliance forwards to a server results in a response that has to return by the same path, you can create a forwarding-session rule. A forwarding-session rule creates forwarding-session entries for traffic that originates from or is destined for a particular network and is forwarded by the NetScaler. You can create forwarding session rules for IPv4 traffic as well as IPv6 traffic.

When configuring an IPv4 forwarding-session rule, you can specify either an IPv4 network address or an extended ACL as the condition for identifying IPv4 traffic for which to create a forwarding-session entry:

- **Network address**. When you specify an IPv4 network address, the appliance creates forwarding sessions for IPv4 traffic whose source or destination matches the network address.
- **Extended ACL rule**. When you specify an extended ACL rule, the appliance creates forwarding sessions for IPv4 traffic that matches the conditions specified in the extended ACL rule.

When configuring an IPv6 forwarding-session rule, you can specify either an IPv6 prefix or an ACL6 as the condition for identifying IPv6 traffic for which to create a forwarding-session entry:

- **IPv6 prefix**. When you specify an IPv6 prefix, the appliance creates forwarding sessions for IPv6 traffic whose source or destination matches the IPv6 prefix.
- **ACL6 rule**. When you specify an ACL6 rule, the appliance creates forwarding sessions for IPv6 traffic that matches the conditions specified in the ACL6 rule.

To create an IPv4 forwarding session rule by using the command line interface

At the command prompt, type the following commands to create a forwarding-session rule and verify the configuration:

- add forwardingSession <name> [<network> <netmask>] | [-aclname <string>] -connfailover (ENABLED | DISABLED)
- show forwardingSession

Example

A network address as the condition:

```
> add forwardingSession fs-nw-1 10.102.105.51 255.255.255.255
 Done
```

An ACL as the condition:

```
> add forwardingSession fs-acl-1 acl1
 Done
```

To configure an IPv4 forwarding session rule by using the configuration utility

Navigate to System > Network > Forwarding Sessions, add a new IPv4 forwarding session, or edit an existing forwarding session.

To create an IPv6 forwarding session rule by using the command line interface

At the command prompt, type the following commands to create a forwarding-session rule and verify the configuration:

- add forwardingSession <name> [<IPv6 prefix>] | [-acl6name <string>]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- show forwardingSession

Example

An IPv6 prefix as the condition:

```
> add forwardingSession fsv6-pfx-1 3ffe::/64
 Done
```

An ACL6 rule as the condition:

```
> add forwardingSession fsv6-acl6-1 –acl6name ACL6-FS
 Done
```

To configure an IPv6 forwarding session rule by using the configuration utility

Navigate to System > Network > Forwarding Sessions, add a new IPv6 forwarding session, or edit an existing forwarding session.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding VLANs

Mar 19, 2012

A NetScaler appliance supports Layer 2 port and IEEE 802.1q tagged VLANs. VLAN configurations are useful when you need to restrict traffic to certain groups of stations. You can configure a network interface as a part of multiple VLANs by using IEEE 802.1q tagging.

You can configure VLANs and bind them to IP subnets. The NetScaler then performs IP forwarding between these VLANs (if it is configured as the default router for the hosts on these subnets).

The NetScaler supports the following types of VLANs:

Port-Based VLANs. The membership of a port-based VLAN is defined by a set of network interfaces that share a common, exclusive Layer 2 broadcast domain. You can configure multiple port-based VLANs. By default, all network interfaces on the NetScaler are members of VLAN 1.

If you apply 802.1q tagging to the port, the network interface belongs to a port-based VLAN. Layer 2 traffic is bridged within a port-based VLAN, and Layer 2 broadcasts are sent to all members of the VLAN if Layer 2 mode is enabled. When you add an untagged network interface as a member of a new VLAN, it is removed from its current VLAN.

Default VLAN. By default, the network interfaces on the NetScaler are included in a single, port-based VLAN as untagged network interfaces. This VLAN is the default VLAN. It has a VLAN ID (VID) of 1. This VLAN exists permanently. It cannot be deleted, and its VID cannot be changed.

When you add a network interface to a to a different VLAN as an untagged member, the network interface is automatically removed from the default VLAN. If you unbind a network interface from its current port-based VLAN, it is added to the default VLAN again.

Tagged VLANs. 802.1q tagging (defined in the IEEE 802.1q standard) allows a networking device (such as the NetScaler) to add information to a frame at Layer 2 to identify the VLAN membership of the frame. Tagging allows network environments to have VLANs that span multiple devices. A device that receives the packet reads the tag and recognizes the VLAN to which the frame belongs. Some network devices do not support receiving both tagged and untagged packets on the same network interface—in particular, Force10 switches. In such cases, you need to contact customer support for assistance.

The network interface can be a tagged or untagged member of a VLAN. Each network interface is an untagged member of one VLAN only (its native VLAN). This network interface transmits the frames for the native VLAN as untagged frames. A network interface can be a part of more than one VLAN if the other VLANs are tagged.

When you configure tagging, be sure to match the configuration of the VLAN on both ends of the link. The port to which the NetScaler connects must be on the same VLAN as the NetScaler network interface.

Note: This VLAN configuration is neither synchronized nor propagated, therefore you must perform the configuration on each unit in an HA pair independently.

Applying Rules to Classify Frames

VLANs have two types of rules for classifying frames:

Ingress rules. Ingress rules classify each frame as belonging only to a single VLAN. When a frame is received on a network interface, the following rules are applied to classify the frame:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- If the frame is untagged, or has a tag value equal to 0, the VID of the frame is set to the port VID (PVID) of the receiving interface, which is classified as belonging to the native VLAN. (PVIDs are defined in the IEEE 802.1q standard.)
- If frame has a tag value equal to FFF, the frame is dropped.
- If the VID of the frame specifies a VLAN of which the receiving network interface is not a member, the frame is dropped. For example, if a packet is sent from a subnet associated with VLAN ID 12 to a subnet associated with VLAN ID 10, the packet is dropped. If an untagged packet with VID 9 is sent from the subnet associated with VLAN ID 10 to a network interface PVID 9, the packet is dropped.

Egress Rules. The following egress rules are applied:
- If the VID of the frame specifies a VLAN of which the transmission network interface is not a member, the frame is discarded.
- During the learning process (defined by the IEEE 802.1q standard), the Src MAC and VID are used to update the bridge lookup table of the NetScaler.
- A frame is discarded if its VID specifies a VLAN that does not have any members. (You define members by binding network interfaces to a VLAN.)

VLANs and Packet Forwarding on the NetScaler

The forwarding process on the NetScaler appliance is similar to that on any standard switch. However, the NetScaler performs forwarding only when Layer 2 mode is on. The key features of the forwarding process are:
- Topology restrictions are enforced. Enforcement involves selecting each network interface in the VLAN as a transmission port (depending on the state of the network interface), bridging restrictions (do not forward on the receiving network interface), and MTU restrictions.
- Frames are filtered on the basis of information in the bridge table lookup in the forwarding database (FDB) table of the NetScaler. The bridge table lookup is based on the destination MAC and the VID. Packets addressed to the MAC address of the NetScaler are processed at the upper layers.
- All broadcast and multicast frames are forwarded to each network interface that is a member of the VLAN, but forwarding occurs only if L2 mode is enabled. If L2 mode is disabled, the broadcast and multicast packets are dropped. This is also true for MAC addresses that are not currently in the bridging table.
- A VLAN entry has a list of member network interfaces that are part of its untagged member set. When forwarding frames to these network interfaces, a tag is not inserted in the frame.
- If the network interface is a tagged member of this VLAN, the tag is inserted in the frame when the frame is forwarded.

When a user sends any broadcast or multicast packets without the VLAN being identified, that is, during duplicate address detection (DAD) for NSIP or ND6 for the next hop of the route, the packet is sent out on all the network interfaces, with appropriate tagging based on either the Ingress and Egress rules. ND6 usually identifies a VLAN, and a data packet is sent on this VLAN only. Port-based VLANs are common to IPv4 and IPv6. For IPv6, the NetScaler supports prefix-based VLANs.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a VLAN

May 26, 2015

You can implement VLANs in the following environments:

- Single subnet
- Multiple subnets
- Single LAN
- VLANs (no tagging)
- VLANs (802.1q tagging)

If you configure VLANs that have only untagged network interfaces as their members, the total number of possible VLANs is limited to the number of network interfaces available in the NetScaler. If more IP subnets are required with a VLAN configuration, 802.1q tagging must be used.

When you bind a network interface to a VLAN, the network interface is removed from the default VLAN. If the network interfaces need to be a part of more than one VLAN, you can bind the network interfaces to the VLANs as tagged members.

You can configure the NetScaler to forward traffic between VLANs at Layer 3. In this case, a VLAN is associated with a single IP subnet. The hosts in a VLAN that belong to a single subnet use the same subnet mask and one or more default gateways connected to that subnet. Configuring Layer 3 for a VLAN is optional. Layer 3 is used for IP forwarding (inter-VLAN routing). Each VLAN has a unique IP address and subnet mask that define an IP subnet for the VLAN. In an HA configuration, this IP address is shared with the other NetScaler appliances. The NetScaler forwards packets between configured IP subnets (VLANs).

When you configure the NetScaler, you must not create overlapping IP subnets. Doing so impedes Layer 3 functionality.

Each VLAN is a unique Layer 2 broadcast domain. Two VLANs, each bound to separate IP subnets, cannot be combined into a single broadcast domain. Forwarding traffic between two VLANs requires a Layer 3 forwarding (routing) device, such as the NetScaler appliance.

Configuring VLANs in an HA Setup

VLAN configuration for a high-availability setup requires that the NetScaler appliances have the same hardware configuration, and the VLANs configured on them must be mirror images.

The correct VLAN configuration is implemented automatically when the configuration is synchronized between the NetScaler appliances. The result is identical actions on all the appliances. For example, adding network interface 0/1 to VLAN2 adds this network interface to VLAN 2 on all the appliances participating in the high-availability setup.

Note: If you use network-interface-specific commands in an HA setup, the configurations you create are not propagated to the other NetScaler appliance. You must perform these commands on each appliance in an HA pair to ensure that the configuration of the two appliances in the HA pair remains synchronized.

Creating or Modifying a VLAN

Updated: 2013-08-29

To configure a VLAN, you create a VLAN entity, and then bind network interfaces and IP addresses to the VLAN. If you remove a VLAN, its member interfaces are added to the default VLAN.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To create a VLAN by using the command line interface

At the command prompt, type:

add vlan <id> [-aliasName <string>] [-ipv6DynamicRouting (ENABLED | DISABLED)]

Example

```
> add vlan 2 –aliasName “Network A”   Done
```

To bind an interface to a VLAN by using the command line interface

At the command prompt, type:

bind vlan <id> -ifnum <slot/port>

Example

```
> bind vlan 2 -ifnum 1/8   Done
```

To bind an IP address to a VLAN by using the command line interface

At the command prompt, type:

bind vlan <id> -IPAddress <IPAddress> <netMask>

Example

```
> bind vlan 2 -IPAddress 10.102.29.54 255.255.255.0   Done
```

To remove a VLAN by using the command line interface

At the command prompt, type:

rm vlan <id>

To configure a VLAN by using the configuration utility

1. Navigate to System > Network > VLANs, add a new VLAN, or edit an existing VLAN.
2. To bind an IP address to a VLAN, under IP Bindings, select the Active option corresponding to the IP address that you want to bind to the VLAN (for example, 10.102.29.54). The Type column displays the IP address type (such as mapped IP, virtual IP, or subnet IP) for each IP address in the IP Address column.
3. To bind a network interface to a VLAN, under Interface Bindings, select the Active option corresponding to the interface that you want to bind to the VLAN.

Monitoring VLANS

Updated: 2013-08-29

You can display VLAN statistics such as packets received, bytes received, packets sent, and bytes sent, and use the information to identify anomalies and or debug a VLAN.

To view the statistics of a VLAN by using the command line interface

At the command prompt, type:

stat vlan <vlanID>

Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
stat vlan 2
```

To view the statistics of a VLAN by using the configuration utility

1. Navigate to System > Network > VLANs.
2. Select the VLAN, and click Statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

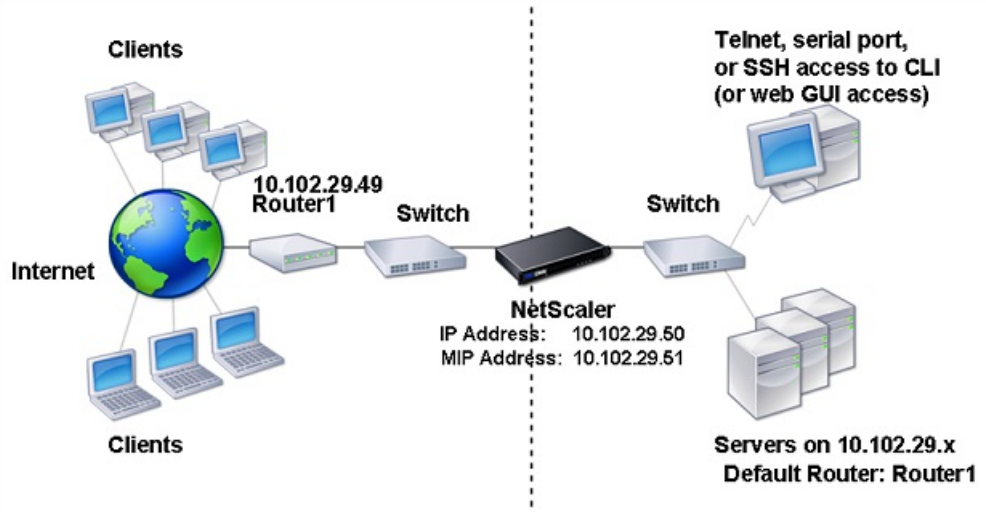
Configuring VLANs on a Single Subnet

Aug 29, 2013

Before configuring a VLAN on a single subnet, make sure that Layer 2 Mode is enabled.

The following figure shows a single subnet environment

Figure 1. VLAN on a Single Subnet



In the above figure:

1. The default router for the NetScaler and the servers is Router 1.
2. Layer 2 mode must be enabled on the NetScaler for the NetScaler to have direct access to the servers.

3. For this subnet, a virtual server can be configured for load balancing on the NetScaler appliance.

To configure a VLAN on a single subnet, follow the procedures described in Creating or Modifying a VLAN. VLAN configuration parameters are not required, because the network interfaces are members of this VLAN.

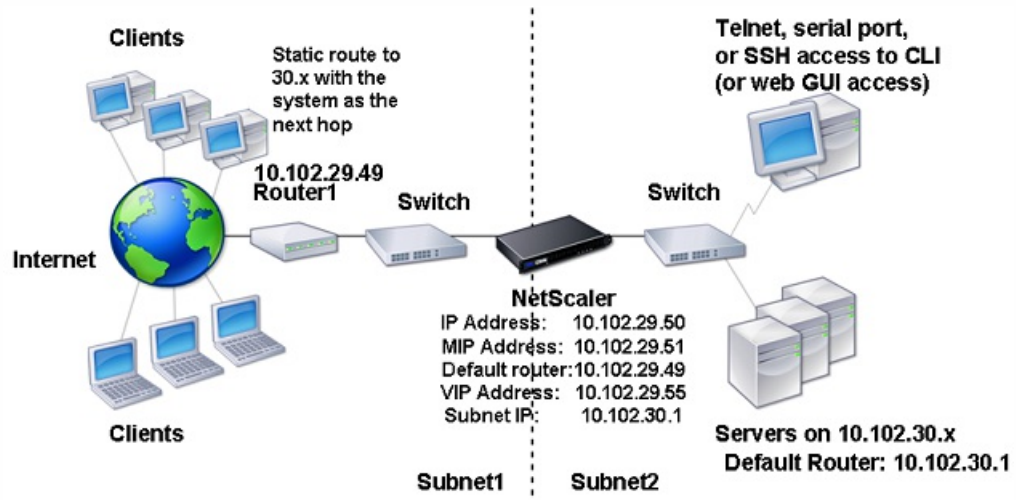
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring VLANs on Multiple Subnets

Aug 29, 2013

To configure a single VLAN across multiple subnets, you must add a VIP for the VLAN and configure the routing appropriately. The following figure shows a single VLAN configured across multiple subnets.

Figure 1. Multiple Subnets in a Single VLAN



To configure a single VLAN across multiple subnets, perform the following tasks:

1. Disable Layer 2 mode. For the procedure to disable Layer 2 mode, see "Enabling and Disabling Layer 2 Mode."

2. Add a VIP.

   For the procedure to add a VIP, see "Configuring and Managing Virtual IP Addresses (VIPs)."

3. Configure RNAT ID.

   For the procedure to configure the RNAT ID, see "Configuring RNAT."

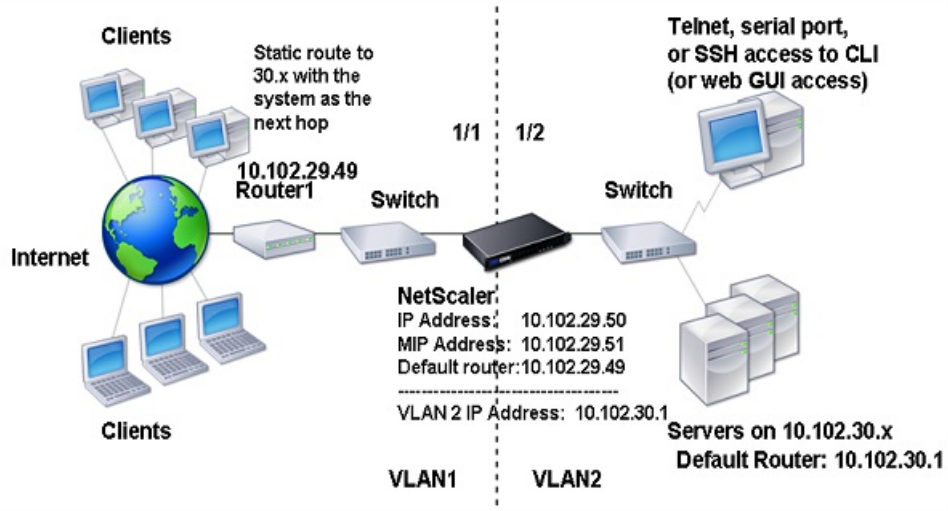
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Multiple Untagged VLANs across Multiple Subnets

Aug 29, 2013

In environments with multiple untagged VLANs across multiple subnets, a VLAN is configured for each IP subnet. A network interface is bound to one VLAN only. The following figure shows this configuration.

Figure 1. Multiple Subnets with VLANs - No Tagging



To implement the configuration shown in the above figure, perform the following tasks:

1. Add VLAN 2.
2. Bind the 1/2 network interface of the NetScaler to VLAN 2 as an untagged network interface.
3. Bind the IP address and subnet mask to VLAN 2.

For procedures on these tasks, see Creating or Modifying a VLAN.

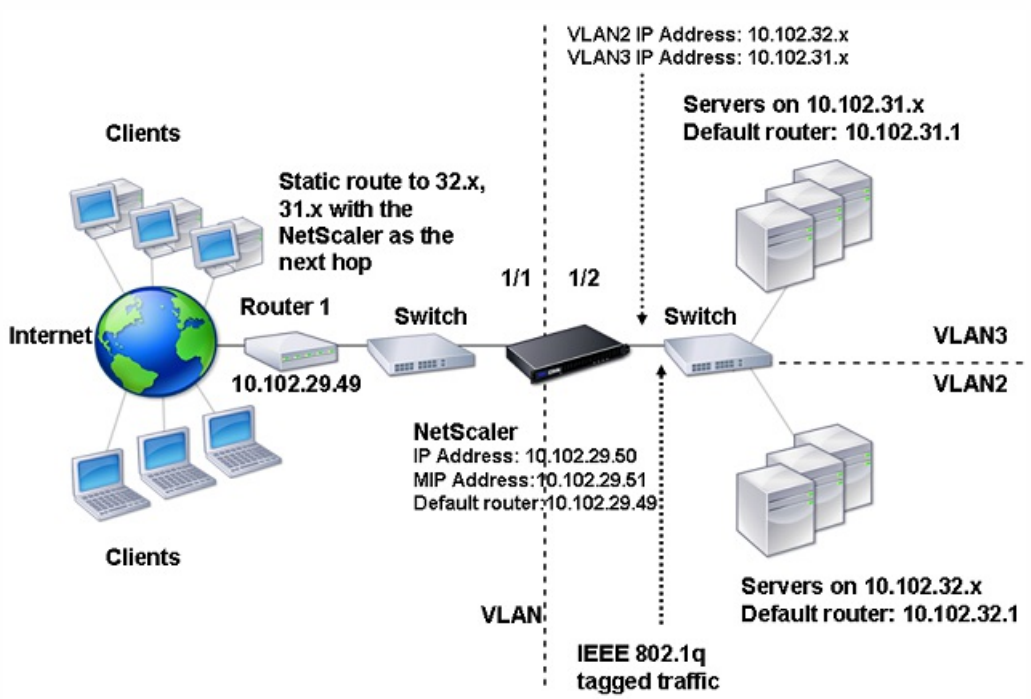
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Multiple VLANs with 802.1q Tagging

Aug 29, 2013

For multiple VLANs with 802.1q tagging, each VLAN is configured with a different IP subnet. Each network interface is in one VLAN. One of the VLANs is set up as tagged. The following figure shows this configuration.

Figure 1. Multiple VLANs with IEEE 802.1q Tagging



To implement the configuration shown in the above figure, perform the following tasks:

1. Add VLAN 2.

   For the procedure to create a VLAN, see "Creating or Modifying a VLAN" in Configuring a VLAN.

2. Bind the 1/2 network interface of the NetScaler to VLAN 2 as an untagged network interface.

   For the procedure to bind a network interface to a VLAN, see "Creating or Modifying a VLAN" in Configuring a VLAN.

3. Bind the IP address and netmask to VLAN 2.

   For the procedure to bind an IP address to a VLAN, see "Creating or Modifying a VLAN" in Configuring a VLAN.

4. Add VLAN 3.

   For the procedure to create a VLAN, see "Creating or Modifying a VLAN" in Configuring a VLAN.

5. Bind the 1/2 network interface of the NetScaler to VLAN 3 as a tagged network interface.

   For the procedure to bind a network interface to a VLAN, see "Creating or Modifying a VLAN" in Configuring a VLAN.

   For the procedure to bind a tagged network interface, see "Creating or Modifying a VLAN" in Configuring a VLAN.

6. Bind the IP address and netmask to VLAN 3.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For the procedure to bind an IP address to a VLAN, see "Creating or Modifying a VLAN" in [Configuring a VLAN].

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring NSVLAN

Jul 05, 2013

NSVLAN is a VLAN to which the NetScaler management IP (NSIP) address's subnet is bound. The NSIP subnet is available only on interfaces that are associated with NSVLAN. By default, NSVLAN is VLAN1, but you can designate a different VLAN as NSVLAN. If you do so, you must reboot the NetScaler appliance for the change to take effect. After the reboot, NSIP subnet traffic is restricted to the new NSVLAN.

The traffic from the NetScaler IP subnet can be tagged (802.1q) with the VLAN ID specified for NSVLAN. You must configure the attached switch interface to tag and allow this same VLAN ID on the connected interface.

If you remove your NSVLAN configuration, the NSIP subnet is automatically bound to VLAN1, restoring the default NSVLAN.

To configure NSVLAN by using the command line interface

At the command prompt, type:

- set ns config -nsvlan <positive_integer> -ifnum <interface_name> ... [-tagged (YES | NO)]
- show ns config

Note: The configuration takes effect after the NetScaler appliance is rebooted.

Example

```
> set ns config -nsvlan 300 -ifnum 1/1 1/2 1/3 -tagged NO
 Done

> save config
 Done
```

To restore the default NSVLAN configuration by using the command line interface

At the command prompt, type:

- unset ns config -nsvlan
- show ns config

Example

```
> unset ns config -nsvlan
 Done
```

To configure NSVLAN by using the configuration utility

Navigate to System > Settings, in the Settings group, click Change NSVLAN Settings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Bridge Groups

Nov 20, 2017

Typically, when you want to merge two or more VLANs into a single domain, you change the VLAN configuration on all the devices in the separate domains. This can be a tedious task. To more easily merge multiple VLANs into a single broadcast domain, you can use bridge groups.

The bridge groups feature works the same way as a VLAN. Multiple VLANS can be bound to a single bridge group, and all VLANs bound to same bridge group form a single broadcast domain. You can bind only Layer 2 VLANs to a bridge group. For Layer 3 functionality, you must assign an IP address to a bridge group.

In Layer 2 mode, a broadcast packet received on an interface belonging to a particular VLAN is bridged to other VLANs that belong to the same bridge group. In the case of a unicast packet, the NetScaler appliance searches its bridge table for the learned MAC addresses of all the VLANs belonging to same bridge group.

In Layer 3 forwarding mode, an IP subnet is bound to a bridge group. The NetScaler accepts incoming packets belonging to the bound subnet and forwards the packets only on VLANs that are bound to the bridge group.

IPv6 routing can be enabled on a configured bridge group.

> **Note**
>
> Bridge Group feature and Bridge BPDU mode cannot work together.

To add a bridge group and bind VLANs by using the command line interface

To add a bridge group and bind VLANs and verify the configuration, type the following commands:

- add bridgegroup <id> [-ipv6DynamicRouting (ENABLED | DISABLED)]
- show bridgegroup <id>
- bind bridgegroup <id> -vlan <positive_integer>
- show bridgegroup <id>

Example

```
> add bridgegroup 12
 Done
```

To remove a bridge group by using the command line interface

At the command prompt, type:

rm bridgegroup <id>

Example

```
 rm bridgegroup 12
```

To configure a bridge group by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Navigate to System > Network > Bridge Groups, add a new bridge group, or edit an existing bridge group.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring VMACs

Sep 07, 2015

The primary and secondary nodes in a high availability (HA) setup share the Virtual MAC address (VMAC) floating entity. The primary node owns the floating IP addresses (such as MIP, SNIP, and VIP) and responds to ARP requests for these IP addresses with its own MAC address. Therefore, the ARP table of an external device, such as an upstream router, is updated with the floating IP address and the MAC address of the primary node.

When a failover occurs, the secondary node takes over as the new primary node. The former secondary node uses Gratuitous ARP (GARP) to advertise the floating IP addresses that it had learned from the old primary node. The MAC address that the new primary node advertises is the MAC address of its own network interface. Some devices (a few routers) do not accept these GARP messages. Therefore, these external devices retain the IP address-to-MAC address mapping that the old primary node had advertised. This can result in a GSLB site going down.

Therefore, you must configure a VMAC on both nodes of an HA pair. This means that both nodes have identical MAC addresses. When a failover occurs, the MAC address of the secondary node remains unchanged, and the ARP tables on the external devices do not need to be updated.

For the procedures to configure a VMAC, see "Configuring Virtual MAC Addresses."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Link Aggregation

Sep 06, 2013

Link aggregation combines data coming from multiple ports into a single high-speed link. Configuring link aggregation increases the capacity and availability of the communication channel between the NetScaler appliance and other connected devices. An aggregated link is also referred to as a "channel." You can configure the channels manually, or you can use Link Aggregation Control Protocol (LACP). You cannot apply LACP to a manually configured channel, nor can you manually configure a channel created by LACP.

When a network interface is bound to a channel, the channel parameters have precedence over the network interface parameters. (That is, the network interface parameters are ignored.) A network interface can be bound only to one channel.

When a network interface is bound to a channel, it drops its VLAN configuration. When network interfaces are bound to a channel, either manually or by LACP, they are removed from the VLANs that they originally belonged to and added to the default VLAN. However, you can bind the channel back to the old VLAN, or to a new one. For example, if you bind the network interfaces 1/2 and 1/3 to a VLAN with ID 2, and then bind them to a channel LA/1, the network interfaces are moved to the default VLAN, but you can bind them back to VLAN 2.

This section includes the following details:

- Configuring Link Aggregation Manually
- Configuring Link Aggregation by Using the Link Aggregation Control Protocol
- Configuring Link Redundancy using LACP channels

Configuring Link Aggregation Manually

Updated: 2013-08-29

When you create a link aggregation channel, its state is DOWN until you bind an active interface to it. You can modify a channel at any time. You can remove channels, or you can enable/disable them.

To create a link aggregation channel by using the command line interface

At the command prompt, type:

- add channel <id> [-ifnum <interfaceName> ...] [-state (ENABLED | DISABLED)] [-speed <speed>] [-flowControl <flowControl>] [-haMonitor (ON | OFF)][-tagall (ON | OFF)] [-ifAlias <string>] [-throughput <positive_integer>] [-bandwidthHigh <positive_integer> [-bandwidthNormal <positive_integer>]]
- show channel

Example

```
add channel LA/1 -ifnum 1/8
show channels
```

To bind an interface to or unbind an interface from an existing link aggregation channel by using the command line interface

At the command prompt, type one of the following commands:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- bind channel <id> <interfaceName>
- unbind channel <id> <interfaceName>

Example

```
bind channel LA/1 1/8
```

To modify a link aggregation channel by using the command line interface

At the command prompt, type the set channel command, the channel ID, and the parameters to be changed, with their new values.

To configure a link aggregation channel by using the configuration utility

Navigate to System > Network > Channels, add a new channel, or edit an existing channel.

To remove a link aggregation channel by using the command line interface

Important: When a channel is removed, the network interfaces bound to it induce network loops that decrease network performance. You must disable the network interfaces before you remove the channel.
At the command prompt, type:

rm channel <id>
Example

```
> rm channel LA/1
 Done
```

To remove a link aggregation channel by using the configuration utility

Important: When a channel is removed, the network interfaces bound to it induce network loops that decrease network performance. You must disable the network interfaces before you remove the channel.
Navigate to System > Network > Channels, select the channel that you want to remove and click Delete.

Configuring Link Aggregation by Using the Link Aggregation Control Protocol

The Link Aggregation Control Protocol (LACP) enables network devices to exchange link aggregation information by exchanging LACP Data Units (LACPDUs). Therefore, you cannot enable LACP on network interfaces that are members of a channel that you created manually.

When using LACP to configure link aggregation, you use different commands and parameters for modifying link aggregation channels than you do for creating link aggregation channels. To remove a channel, you must disable LACP on all interfaces that are part of the channel.

Note: In an High Availability configuration, LACP configurations are neither propagated nor synchronized.

Configuring the LACP System Priority

Updated: 2013-10-01

The LACP system priority determines which peer device of an LACP LA channel can have control over the LA channel. This number is globally applied to all LACP channels on the appliance. The lower the value, the higher the priority.

To configure the LACP system priority by using the command line interface

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to set the priority for a standalone appliance and verify the configuration:

- set lacp -sysPriority <positive_integer>
- show lacp

Example:

set lacp -sysPriority 50

To set the priority for a specific cluster node, log on to the cluster IP address and at the command prompt, type the following commands:

- set lacp -sysPriority <positive_integer> -ownerNode <positive_integer>
- show lacp

Example:

set lacp -sysPriority 50 -ownerNode 2

To configure the LACP system priority by using the configuration utility

1. Navigate to System > Network > Interfaces and, in the Action list, select Set LACP.
2. Specify the system priority and the owner node (applicable only for a cluster setup).

Creating Link Aggregation Channels

Updated: 2013-08-29

For creating a link aggregation channel by using LACP, you need to enable LACP and specify the same LACP key on each interface that you want to be the part of the channel. For example, if you enable LACP and set the LACP Key to 3 on interfaces 1/1 and 1/2, a link aggregation channel LA/3 is created and interfaces 1/1 and 1/2 are automatically bound to it.

Note: When enabling LACP on a network interface, you must specify the LACP Key.
By default, LACP is disabled on all network interfaces.

To create an LACP channel by using the command line interface

At the command prompt, type:

- set interface <id> [-lacpMode <lacpMode>] [-lacpKey<positive_integer>] [-lacpPriority <positive_integer>] [-lacpTimeout (LONG | SHORT)]
- show interface [<id>]

To create an LACP channel by using the configuration utility

Navigate to System > Network > Interfaces, open the network interface, and set the parameters.

Modifying Link aggregation Channels

Updated: 2013-08-29

After you have created an LACP channel by specifying interfaces, you can modify properties of the channel.

To modify an LACP channel using the command line interface

At the command prompt, type:

- set channel <id> [-ifnum <interfaceName> ...] [-state (ENABLED | DISABLED)] [-speed <speed>] [-flowControl

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

<flowControl>] [-haMonitor (ON | OFF)] [-ifAlias <string>] [-throughput <positive_integer>] [-tagall (ON | OFF)] [-bandwidthHigh <positive_integer> [-bandwidthNormal <positive_integer>]]

- show channel

Example

```
> set channel LA/3 -state ENABLED -speed 10000
 Done
```

To modify an LACP channel by using the configuration utility

Navigate to System > Network > Channels, and modify an existing LACP channel.

Removing a Link Aggregation Channel

Updated: 2013-08-29

To remove a link aggregation channel that was created by using LACP, you need to disable LACP on all the interfaces that are part of the channel.

To remove an LACP channel by using the command line interface

At the command prompt, type:

- set interface <id> -lacpMode Disable
- show interface [<id>]

To remove an LACP channel by using the configuration utility

Navigate to System > Network > Interfaces, open the network interface, and clear the Enable LACP option.

Configuring Link Redundancy using LACP channels

Updated: 2014-04-08

Link Redundancy using LACP channels enables the NetScaler ADC to divide an LACP channel into logical subchannels, with one subchannel active and the others in standby mode. If the active subchannel fails to meet a minimum threshold of throughput, one of the standby subchannels becomes active and takes over.

A subchannel is created from links that are part of the LACP channel and are connected to a particular device. For example, for an LACP channel with four interfaces on a NetScaler ADC, with two of the interfaces connected to device A and the other two connected to device B, the ADC creates two logical subchannels, one subchannel with two links to device A, and another subchannel with two links to device B.

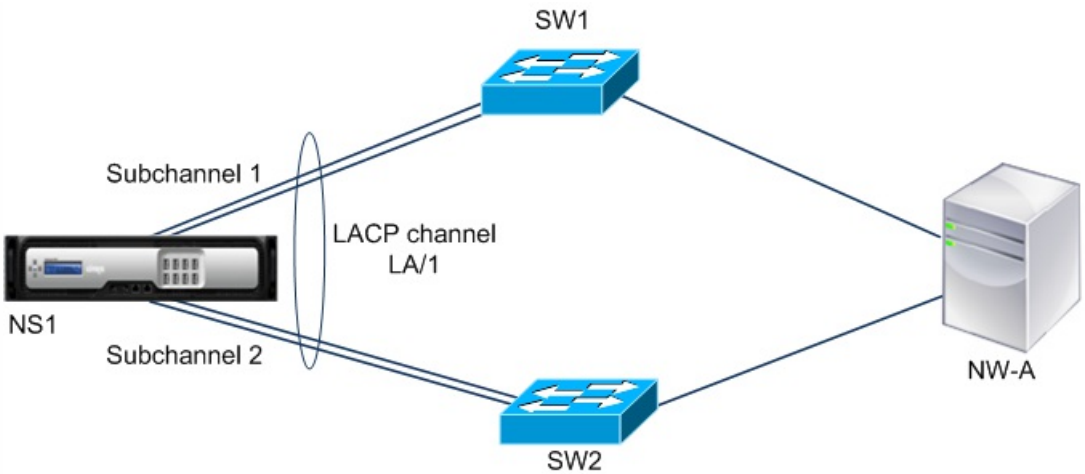
To configure link redundancy for an LACP channel, set the lrMinThroughput parameter, which specifies the minimum throughput threshold (in Mbps) to be met by the active subchannel. Setting this parameter automatically creates the subchannels. When the maximum supported throughput of the active channel falls below the lrMinThroughput value, link failover occurs and a standby subchannel becomes active.

If you unset the lrMinThroughput parameter of an LACP channel, or set the value to zero, link redundancy for that channel is disabled, which is the default setting.

Example

Consider an example of link redundancy configured between NetScaler ADC NS1 and switches SW1 and SW2.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.



NS1 is connected to network device NW-A through SW1 and SW2.

On NS1, LACP channel LA/1 is created from interfaces 1/1, 1/2, 1/3, and 1/4. Interfaces 1/1 and 1/2 of NS1 are connected to SW1, and interfaces 1/3 and 1/4 are connected to SW2. Each of the four links supports a maximum throughput of 1000Mbps.

When the lrMinThroughput parameter is set to some value (say 2000), NS1 creates two logical subchannels from LA/1, one subchannel (say subchannel 1) using interfaces 1/1 and 1/2 (connected to SW1), and the other subchannel (subchannel 2) using interfaces 1/3 and 1/4 (connected to SW2).

NS1 applies an algorithm to make one subchannel (say subchannel 1) active and put the other on standby. NS1 and network device NW-A are accessible to each other through only the active subchannel.

Say subchannel 1 is active, and its maximum supported throughput falls below the lrMinThroughput value (for example, one of its links fails, and the maximum supported throughput falls to 1000 Mbps). Subchannel 2 becomes active and takes over.

Points to Consider when Configuring Link Redundancy in a High Availability setup

In a high availability (HA) configuration, if you want to configure throughput (throughput parameter) based HA failover and link redundancy (lrMinThroughput parameter) on an LACP channel, you must set the throughput parameter to a value less than or equal to that of the lrMinThroughput parameter.

The maximum supported throughput of an LACP channel is calculated as the maximum supported throughput of the active subchannel.

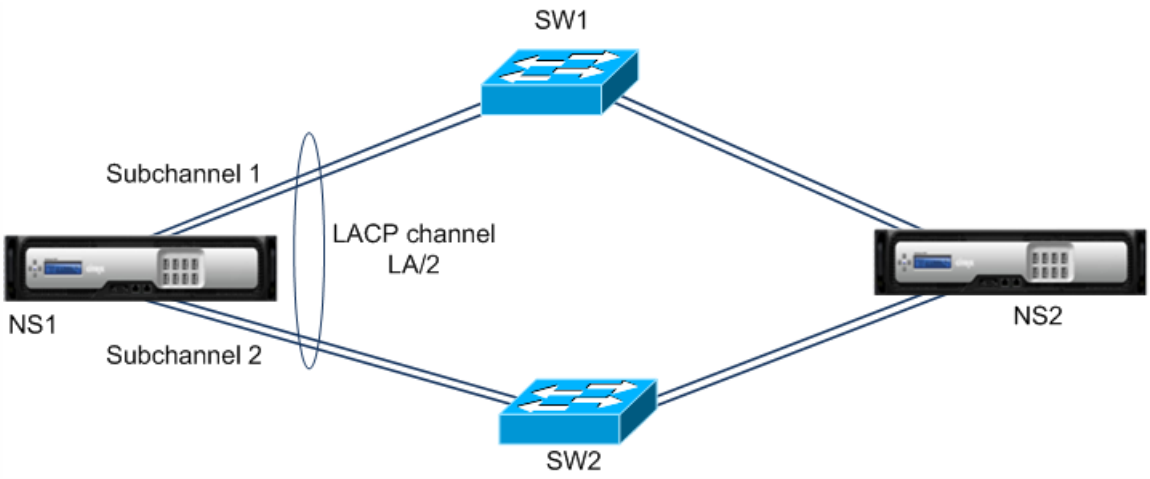
If the throughput parameter value is equal to or less than the lrminthroughput parameter value, HA failover occurs when both of the following conditions exist at the same time:

- None of the subchannels' maximum supported throughput meet the lrMinThroughput parameter value
- The maximum supported throughput of the LACP channel does not meet the throughput parameter value

Consider an example of an HA setup that has NetScaler ADCs NS1 and NS2, with switches SW1 and SW2. NS1 is connected to NS2 through SW1 and SW2.

On NS1, LACP channel LA/1 is created from interfaces 1/1, 1/2, 1/3, and 1/4. Interfaces 1/1 and 1/2 of NS1 are connected to SW1, and interfaces 1/3 and 1/4 are connected to SW2. Each of the four links supports a maximum throughput of 1000 Mbps.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.



Following are the LACP-parameter settings in this example:

Parameter	Value
Throughput	2000
lrminthroughput	2000

NS1 forms two subchannels from LA/1, one subchannel (say subchannel 1) using interfaces 1/1 and 1/2 (connected to SW1), and the other subchannel (subchannel 2) using interfaces 1/3 and 1/4 (connected to SW2). Each of the two subchannels supports a maximum throughput of 2000 Mbps. Applying an algorithm, NS1 makes one subchannel (say subchannel 1) active and the other standby.

Say subchannel 1 is active, and its maximum supported throughput falls below the lrMinThroughput value (for example, one of its links fails, and maximum supported throughput falls to 1000 Mbps). Subchannel 2 becomes active and takes over. HA failover does not occur, because the maximum supported throughput of the LACP channel is not less than the throughput parameter value:

Maximum supported throughput of the LACP channel = Maximum supported throughput of the active channel = Maximum supported throughput of subchannel 2 = 2000 Mbps

If subchannel 2's maximum supported throughput also falls below the lrminthroughput value (for example, one of its links fails, and the maximum supported throughput falls to 1000 Mbps), HA failover occurs, because the maximum supported throughput of the LACP channel is then less than the throughput parameter value:

To configure link redundancy for a LACP channel by using the command line interface

At the command prompt, type the following commands to configure the channel and verify the configuration:

- set channel <id> -lrMinThroughput <positive_integer>
- show channel

Example

```
> set channel la/1 –lrMinThroughput  2000
Done
> set channel la/2 –throughput 2000 –lrMinThroughput  2000
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Done

To configure link redundancy for a LACP channel by using configuration utility

1. Navigate to System > Network > Channels.
2. In the details pane, select an LACP channel for which you want to configure link redundancy, and then click Edit.
3. In the Configure LACP channel dialog box, set the lrMinThroughput parameter.
4. Click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Redundant Interface Set

Oct 20, 2016

A redundant interface set is a set of interfaces where one of the interfaces is active and the remaining ones are standby. If the active interface fails, one of the standby interfaces takes over and becomes active.

The following are the main benefits of using redundant interface sets:

- A redundant interface set ensures connection reliability between the NetScaler appliance and a peer device by providing back up links between them.
- Unlike link redundancy using LACP, no configuration is required on the peer device for a redundant interface set. To the peer device, redundant interface set appear as individual interfaces and not as a set or collection.
- In an high availability configuration (HA), redundant interface sets can minimize the number the HA failovers.

A link redundant set is specified in LR/X notation, where X can range from 1 to 4. For example, LR/1.

> **Note**
>
> Redundant Interface Set was formerly known as 'NIC bundling' when first introduced in 10.5 release.

How Redundant Interface Set Works

For a redundant interface set, the NetScaler appliance derives a MAC address on the basis of an internal algorithm and assigns it to the redundant interface set. This MAC address is shared by all the member interfaces and is used only by the active interface at a time. The active interface broadcasts GARP messages, which contains the MAC address assigned to the redundant interface set and not the interface's own physical MAC address. When the current active interface fails and is taken over by another interface, the new active interface sends GARP messages. The peer device updates its forwarding table with the new active interface information. The standby interfaces do not send any GARP messages. The standby interfaces do not send any packets and they drop any packets they receive.

In a redundant interface set, selection of the member interface as active is based on either of the following factors:

- **Redundant interface priority**. This is a parameter of an interface and it defines the priority of the interface in a redundant interface set for the active member selection. This parameter specifies a positive integer. Lower the value higher the priority of active member selection. The member interface with the highest priority (lowest value) is selected as the active interface of the redundant interface set.
- **Binding order of the member interfaces**. If all the member interfaces have the same redundant interface priority, the member interface that was bound first to the redundant interface set is selected as the active interface of the redundant interface set.

In a redundant interface set, active interface selection is triggered in one of the following events:

- When the current active interface fails or you disable it.
- When you set the priority of a standby interface to a value lower than that of the current active interface. The standby interface takes over as the active interface.
- When you bind an interface whose priority is lower than that of the current active interface. The newly bound interface takes over as the active interface.

Points to Consider for Configuring Redundant Interface Sets

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2015-03-2

Consider the following points before you configure a redundant interface set:

- In a high availability configuration, redundant interface set configurations do not propagate or synchronize to the secondary node.
- In a NetScaler cluster configuration, redundant interface sets are not supported.
- You can configure a maximum of four redundant interface sets.
- You can bind a maximum of 16 interfaces to a redundant interface set.
- Member interfaces of a redundant interface set cannot be bound to another redundant interface set.
- Member interfaces of a redundant interface set cannot be bound to a LA channel.
- LA channels cannot be bound to a redundant interface set.

Configuration Steps

Configuring redundant interface set on a NetScaler appliance consists of the following tasks:

- **Create a redundant interface set**. Use the channel command operation for creating a redundant interface set.

  A link redundant set is specified in LR/X notation, where X can range from 1 to 4. For example, LR/1.

- **Bind interfaces to the redundant interface set**. Associate the desired interfaces with the redundant interface set. An interface cannot be a part of multiple redundant interface sets.
- **(Optional) Set a redundant interface priority on the member interface**. Use the interface command operation for setting the redundant interface priority on a desired member interface of a redundant interface set.

To create a redundant interface set by using the command line interface

At the command prompt:

- add channel <ID>
- show channel <ID>

To bind interfaces to a redundant interface set by using the command line interface

At the command prompt:

- bind channel <ID> <ifnum>
- show channel <ID>

To set a redundant interface priority of an interface by using the command line interface

At the command prompt:

- set interface <ID> -lrsetpriority <positive_integer>
- show interface <ID>

Example

In the following example, redundant interface set LR/1 is created, and interfaces 1/1, 1/2, 1/3, and 1/4 are bound to LR/1. The redundant interface priority is set to a default value of 1024 for all these member interfaces. Output of the show

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

channel command displays that the interface 1/1 is the current active interface for the redundant interface set lr/1.

```
> add channel lr/1
 Done
> bind channel lr/1 1/1 1/2 1/3 1/4
 Done
> show channel
1)    Interface LR/1 (Link Redundant) #23
      flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
      MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
      Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
            throughput 0
      Actual: throughput 1000
      LLDP Mode: NONE,
      RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
      TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
      NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
      Bandwidth thresholds are not set.
            1/1: UTP-1000-FULL-OFF        UP  0h14m06s   LR Active Member
            1/2: UTP-1000-FULL-OFF        UP  0h14m06s   LR Inactive Member
            1/3: UTP-1000-FULL-OFF        UP  0h14m06s   LR Inactive Member
            1/4: UTP-1000-FULL-OFF        UP  0h14m06s   LR Inactive Member
Done
```

In the following example, redundant interface priority of the member interface 1/4 is set to 100, which is lower than the set redundant interface priority of all the other member interfaces of LR/1.

Output of the show channel command displays that the interface 1/4 is the current active interface for the redundant interface set LR/1.

```
> set interface 1/4 -lrsetPriority 100
Done
> show channel
1)     Interface LR/1 (Link Redundant) #23
      flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
      MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
      Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
            throughput 0
      Actual: throughput 1000
      LLDP Mode: NONE,
      RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
      TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
      NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
      Bandwidth thresholds are not set.
            1/1: UTP-1000-FULL-OFF        UP  0h14m06s   LR Inactive Member
            1/2: UTP-1000-FULL-OFF        UP  0h14m06s   LR Inactive Member
            1/3: UTP-1000-FULL-OFF        UP  0h14m06s   LR Inactive Member
            1/4: UTP-1000-FULL-OFF        UP  0h14m06s   LR Active Member
Done
```

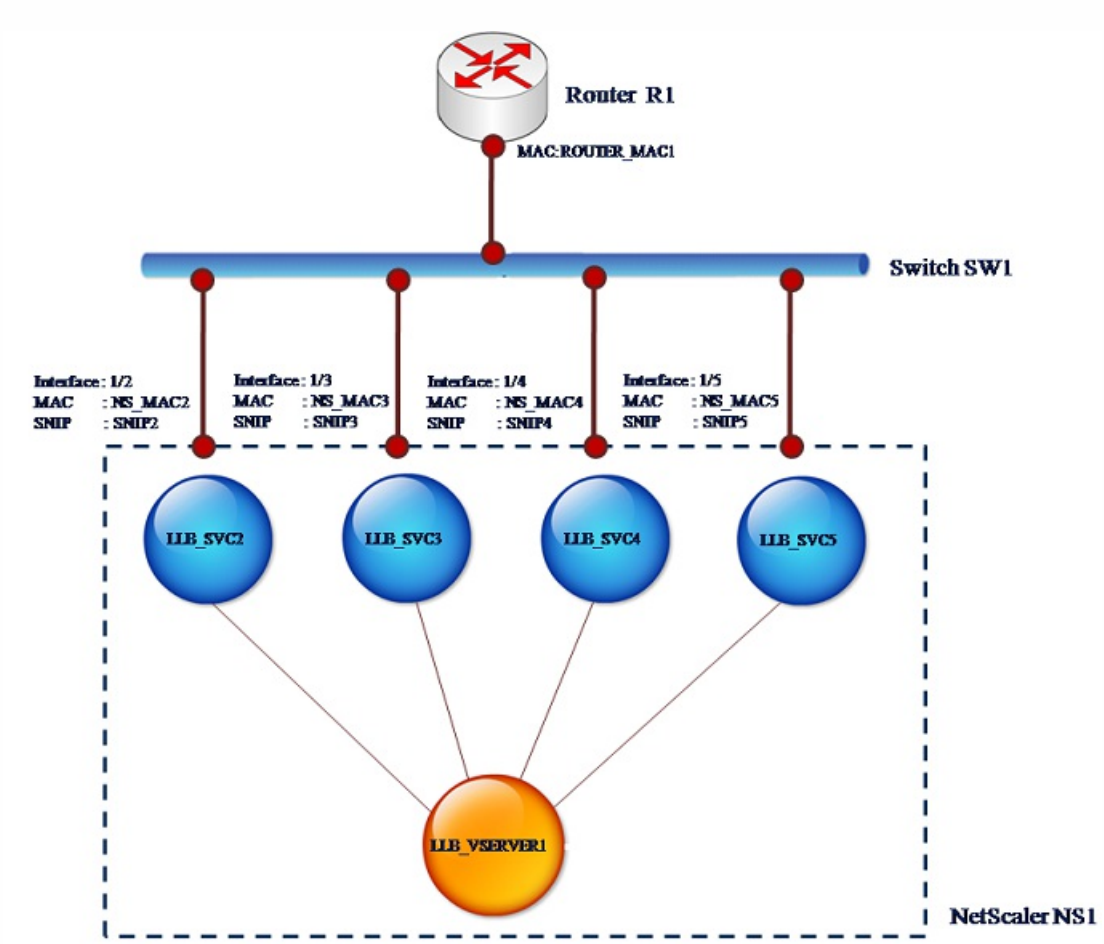
 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding an SNIP address to an Interface

Aug 29, 2013

You can now bind a NetScaler owned SNIP address to an interface without using Layer 3 VLANs. Any packets related to the SNIP address will go only through the bound interface.

This feature can be useful in a scenario where the upstream switch does not support Link Aggregation channels and you want the NetScaler appliance to load balance traffic, originated from a server, across the four links to the upstream switch as shown in the following illustration.



The following tables describe the example settings for the scenario:

Entity	Name	Value
SNIP addresses on NS1	SNIP2 (for reference purpose only)	10.10.10.2
	SNIP3 (for reference purpose only)	10.10.10.3
	SNIP4 (for reference purpose only)	10.10.10.4

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity	Name	Value
	SNIP5 (for reference purpose only)	10.10.10.5
LLB virtual server on NS1	LLB_VSERVER1	-
Transparent monitor on NS1	TRANS_MON	-
LLB services on NS1	LLB_SVC2	10.10.10.240
	LLB_SVC3	10.10.10.120
	LLB_SVC4	10.10.10.60
	LLB_SVC5	10.10.10.30
MAC address of interface 1/2 on NS1	NS_MAC_2 (for reference purpose only)	00:e0:ed:0f:bc:e0
MAC address of interface 1/3 on NS1	NS_MAC_3 (for reference purpose only)	00:e0:ed:0f:bc:df
MAC address of interface 1/4 on NS1	NS_MAC_4 (for reference purpose only)	00:e0:ed:0f:bc:de
MAC address of interface 1/5 on NS1	NS_MAC_5 (for reference purpose only)	00:e0:ed:1c:89:53
IP address of Router R1	Router_IP (for reference purpose only)	10.10.10.1
MAC address of interface of R1	ROUTER_MAC1 (for reference purpose only)	00:21:a1:2d:db:cc

To configure the example settings

1. Add four different SNIPs in different subnet ranges. This is for ARP to be resolved on four different links. For more information on creating a SNIP address, see "Configuring Subnet IP Addresses (SNIPs)."
   Command Line Interface example

```
> add ns ip 10.10.10.2 255.255.255.0 -type SNIP
 Done
> add ns ip 10.10.10.3 255.255.255.128 –type SNIP
Done
> add ns ip 10.10.10.4 255.255.255.192 –type SNIP
Done
> add ns ip 10.10.10.5 255.255.255.224 –type SNIP
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Add four different dummy services in the added SNIP subnets. This is to ensure that the traffic is sent out with source IP as one of the four configured SNIPs. For more information on creating a service, see "Configuring Services."
   Command Line Interface example

   ```
   > add service LLB_SVC2 10.10.10.240 any *
    Done
   > add service LLB_SVC3 10.10.10.120 any *
    Done
   > add service LLB_SVC4 10.10.10.60 any *
    Done
   > add service LLB_SVC5 10.10.10.30 any *
    Done
   ```
3. Add a transparent ping monitor for monitoring the gateway. Bind the monitor to each of the configured dummy services. This is to make the state of the services as UP. For more information on creating a transparent monitor, see "Creating and Binding a Transparent Monitor."
   Command Line Interface example

   ```
   > add monitor TRANS_MON ping -destIP 10.10.10.1 -transparent YES
    Done
   > bind monitor TRANS_MON LLB_SVC2
    Done
   > bind monitor TRANS_MON LLB_SVC3
    Done
   > bind monitor TRANS_MON LLB_SVC4
    Done
   > bind monitor TRANS_MON LLB_SVC5
    Done
   ```
4. Add a link load balancing (LLB) virtual server and bind the dummy services to it. For more information on creating an LLB virtual server, see "Configuring an LLB Virtual Server and Binding a Service."
   Command Line Interface example

   ```
   > add lb vserver LLB_VSERVER1 any
    Done
   > set lb vserver LLB_VSERVER1 -lbmethod ROUNDROBIN
    Done
   > bind lb vserver LLB_VSERVER1 LLB_SVC2
    Done
   > bind lb vserver LLB_VSERVER1 LLB_SVC2
    Done
   > bind lb vserver LLB_VSERVER1 LLB_SVC2
    Done
   > bind lb vserver LLB_VSERVER1 LLB_SVC2
    Done
   ```
5. Add the LLB virtual server as the default LLB route. For more information on creating an LLB route see "Configuring an

© 1999-2017 Citrix Systems, Inc. All rights reserved.

LLB Route."

Command Line Interface example

```
> add lb route 0.0.0.0 0.0.0.0  LLB_VSERVER1
 Done
```

6. Add an ARP entry for each of the dummy services with the MAC address of the gateway. This way the gateway is reachable through these dummy services. For more information on adding an ARP entry, see "Configuring Static ARP."

   Command Line Interface example

```
> add arp -ipaddress 10.10.10.240 -mac 00:21:a1:2d:db:cc -ifnum 1/2
 Done
> add arp -ipaddress 10.10.10.120 -mac 00:21:a1:2d:db:cc -ifnum 1/3
 Done
> add arp -ipaddress 10.10.10.60 -mac 00:21:a1:2d:db:cc -ifnum 1/4
 Done
> add arp -ipaddress 10.10.10.30 -mac 00:21:a1:2d:db:cc -ifnum 1/5
 Done
```

7. Bind a specific interface to an SNIP by adding an ARP entry for each of these SNIPs. This is to ensure that the response traffic will reach the same interface through which the request went out. For more information on adding an ARP entry, see "Configuring Static ARP."

   Command Line Interface example

```
> add arp -ipAddress 10.10.10.2 -mac 00:e0:ed:0f:bc:e0 -ifnum 1/2
 Done
> add arp -ipAddress 10.10.10.3 -mac 00:e0:ed:0f:bc:df -ifnum 1/3
 Done
> add arp -ipAddress 10.10.10.4 -mac 00:e0:ed:0f:bc:de -ifnum 1/4
 Done
> add arp -ipAddress 10.10.10.5 -mac 00:e0:ed:1c:89:53 -ifnum 1/5
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring the Bridge Table and Changing the Aging time

Aug 29, 2013

NetScaler appliance bridges frames on the basis of bridge table lookup of the destination MAC address and the VLAN ID. However, the appliance performs forwarding only when Layer 2 mode is enabled.

The bridge table is dynamically generated, but you can display it, modify the aging time for the bridge table, and view bridging statistics.

All the MAC entries in the bridge table are updated with the aging time.

To change the aging time by using the command line interface

At the command prompt, type:

- set bridgetable -bridgeAge <positive_integer>
- show bridgetable

Example

```
> set bridgetable -bridgeage 70
 Done
```

To change the aging time by using the configuration utility

1. Navigate to System > Network > Bridge Table.
2. Click Change Ageing Time, and set the Ageing Time (seconds) parameter.
3. In the Action list, select Change Ageing Time, and set the Ageing Time (seconds) parameter.

To view the statistics of a bridge table by using the command line interface

At the command prompt, type:

```
stat bridge
```

To view the statistics of a bridge table by using the configuration utility

Navigate to System > Network > Bridge Table, select the MAC address, and click Statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding NetScaler Appliances in Active-Active Mode Using VRRP

Aug 29, 2013

An active-active deployment, in addition to preventing downtime, makes efficient use of all the NetScaler appliances in the deployment. In active-active deployment mode, the same VIPs are configured on all NetScaler appliances in the configuration, but with different priorities, so that a given VIP can be active on only one appliance at a time.

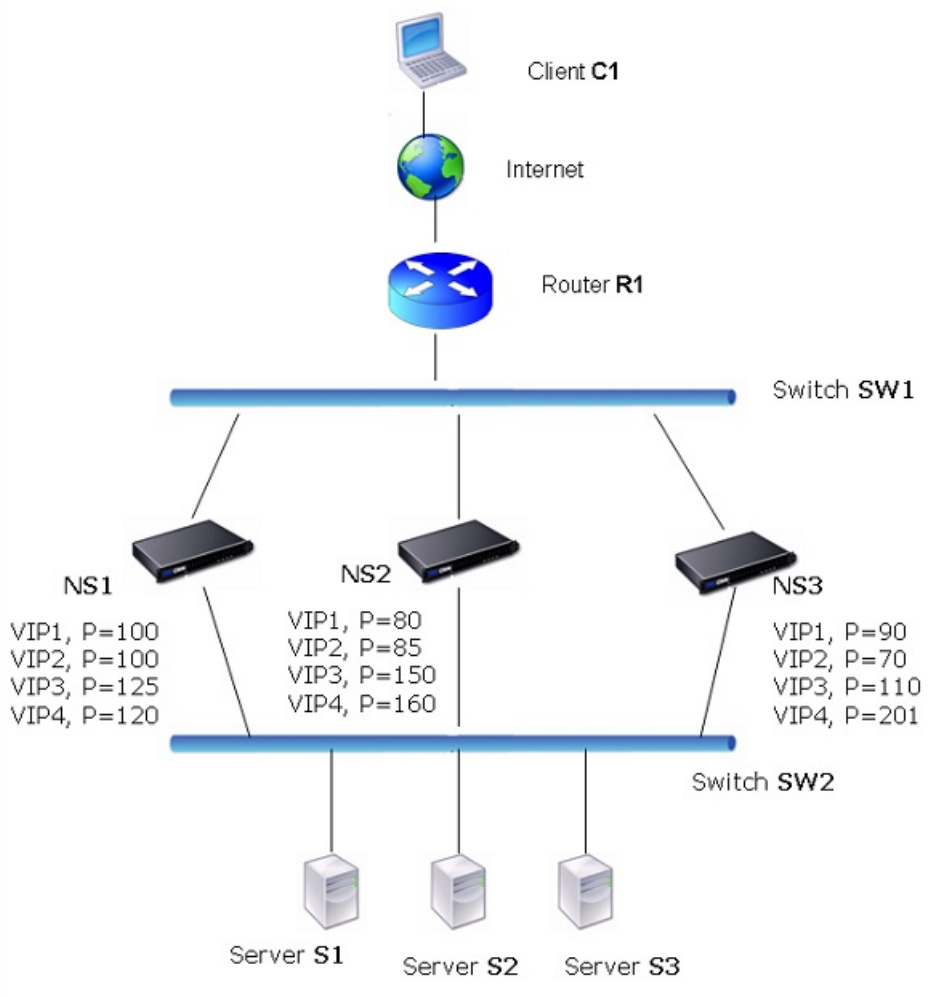
Note: This feature is supported only on NetScaler nCore builds.

The active VIP is called the master VIP, and the corresponding VIPs on the other NetScaler appliances are called the backup VIPs. If a master VIP fails, the backup VIP with the highest priority takes over and becomes the master VIP. All the NetScaler appliances in an active-active deployment use the Virtual Router Redundancy Protocol (VRRP) protocol to advertise their VIPs and the corresponding priorities at regular intervals.

NetScaler appliances in active-active mode can be configured so that no NetScaler is idle. In this configuration, different sets of VIPs are active on each NetScaler. For example, in the following diagram, VIP1, VIP2, VIP3, and VIP4 are configured on appliances NS1, NS2, and NS3. Because of their priorities, VIP1 and VIP 2 are active on NS1, VIP3 is active on NS2 and VIP 4 is active on NS3. If, for example, NS1 fails, VIP1 on NS3 and VIP2 on NS2 become active.

Figure 1. An Active-Active Configuration

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The NetScaler appliances in the above diagram process traffic as follows:

1. Client C1 sends a request to VIP1. The request reaches R1.
2. R1 does not have an ARP entry for VIP1, so it broadcasts an ARP request for VIP1.
3. VIP1 is active in NS1, so NS1 replies with a source MAC address as the VMAC (for example VMAC1) associated with VIP1, and VIP1 as the source IP address.
4. SW1 learns the port for VIP1 from the ARP reply and updates its bridge table.
5. R1 updates the ARP entry with VMAC1 and VIP1.
6. R1 forwards the packet to the VIP1 on NS1.
7. NS1's load balancing algorithm selects server S2, and NS1 opens a connection between one of its SNIP or MIP addresses and S2.
8. S2 replies to the SNIP or MIP on the NetScaler.
9. NS1 sends S2's reply to the client. In the reply, NS1 inserts MAC address of the physical interface as the source MAC address and VIP1 as the source IP address.
10. Should NS1 fail, the NetScaler appliances use the VRRP protocol to select the VIP1 with the highest priority. In this case, VIP1 on NS3 becomes active, and the following two steps update the active-active configuration.
11. NS3 broadcasts a GARP message for VIP1. In the message, VMAC1 is the source MAC address and VIP1 is the source IP address.
12. SW1 learns the new port for VMAC1 from the GARP broadcast and updates its bridge table to send subsequent client requests for VIP1 to NS3. R1 updates its ARP table.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The priority of a VIP can be modified by health tracking. If you enable health tracking, you should make sure that preemption is also enabled, so that a VIP whose priority is lowered can be preempted by another VIP.

In some situations, traffic might reach a backup VIP. To avoid dropping such traffic, you can enable sharing, on a per-node basis, as you create an active-active configuration. Or you can enable the global send to master option. On a node on which sharing is enabled, it takes precedence over send to master.

Health Tracking

Base priority (BP-range 1-255) ordinarily determines which VIP is the master VIP, but effective priority (EP) can also affect the determination.

For example, if a VIP on NS1 has a priority of 101 and same VIP on NS2 has a priority of 99, the VIP on NS1 is active. However, if two vservers are using the VIP on NS1 and one of them goes DOWN, health tracking can reduce the EP of VIP on NS1. VRRP then makes the VIP on NS2 the active VIP.

Following are the health tracking options for modifying EP:

- **NONE.** No tracking. EP = BP
- **ALL.** If all virtual servers are UP, then EP = BP. Otherwise, EP = 0.
- **ONE.** If at least one virtual server is UP, then EP = BP. Otherwise, EP = 0.
- **PROGRESSIVE.** If ALL virtual servers are UP, then EP = BP. If ALL virtual servers are DOWN then EP = 0. Otherwise EP = BP (1 - K/N), where N is the total number of virtual servers associated with the VIP and k is the number of virtual servers that are down.

Note: If you specify a value other than NONE, preemption should be enabled, so that the backup VIP with the highest priority becomes active if the priority of the master VIP is downgraded.

Preemption

Preemption of an active VIP by another VIP that attains a higher priority is enabled by default, and normally should be enabled. In some cases, however, you may want to disable it. Preemption is a per-node setting for each VIP.

Preemption can occur in the following situations:

- An active VIP goes down and a VIP with a lower priority takes its place. If the VIP with the higher priority comes back online, it preempts the currently active VIP.
- Health tracking causes the priority of a backup VIP to become higher than that of the active VIP. The backup VIP then preempts the active VIP.

Sharing

In the event that traffic reaches a backup VIP, the traffic is dropped unless the sharing option is enabled on the backup VIP. This behavior is a per node setting for each VIP and is disabled by default.

In the figure "An Active-Active Configuration," VIP1 on NS1 is active and VIP1 VIPs on NS2 and NS3 are backups. Under certain circumstances, traffic may reach VIP1 on NS2. If Sharing is enabled on NS2, this traffic is processed instead of dropped.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Active-Active Mode

Jan 18, 2016

On each NetScaler appliance that you want to deploy in active-active mode, you must add a Virtual MAC (VMAC) and bind the VMAC to a VIP. The VMAC for a given VIP must be same on each appliance. For example, if VIP 10.102.29.5, is created on the appliances, a virtual router ID (VRID) must be created on each NetScaler and bound to VIP 10.102.29.5 on each NetScaler. When you bind a VMAC to a VIP, the NetScaler sends VRRP advertisements to each VLAN that is bound to that VIP. The VMAC can be shared by different VIPs configured on the same NetScaler.

Configuring IPv4 Active-Active Mode

Perform the following tasks on each of the NetScaler appliances to be included in the active-active configuration:

- **Add a VMAC address**. Add a VMAC address by adding a VRID. You can also specify a priority and enable or disable preemption and sharing on this VRID address.
- **Add a VIP address and associate the VMAC's VRID**. Add a VIP address and set the VRID parameter to the newly created VRID. The attributes of the VRID (for example, priority and preemption) are bound to this VIP address.
  Note: The same VIP address must be added to all the other NetScaler appliances.

To add a VMAC address by using theNetScaler command line

At the command prompt, type:

- **add vrID** <id> [-**priority** <positive_integer>] [-**preemption** (**ENABLED** | **DISABLED**)][-**sharing** (**ENABLED** | **DISABLED**)] [-**tracking** <tracking>]
- **show vrid**

To add a VIP address by using the NetScaler command line

At the command prompt, type:

- **add ns ip** <IPv4Address> -type VIP -vrid <value>
- **show ns ip**

To configure a VMAC by using the configuration utility

1. Navigate to **System** > **Network** > **VMAC**, on the **VMAC** tab, add a new VMAC, or edit an existing VMAC.
2. Set the following parameters:
   - Virtual Router ID
   - Priority
   - Tracking
   - Preemption
   - Sharing

To configure a VIP address and associate the VRID to it by using the NetScaler configuration utility

1. Navigate to **System** > **Network** > **IPs**, on the **IPV4s** tab, add an IP address of type VIP.
2. While adding the IP address, select the virtual router ID from the **Virtual Router Id** drop down box.

Sample Configuration

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following sample configuration is for deploying NetScaler appliances NS1 and NS2 in IPv4 active-active mode. VIP address 203.0.113.10 is configured on both NS1 and NS2, with a different priority value on each appliance. On each appliance, this VIP address is bound to a VMAC address. 203.0.113.10 is master on NS2, because its priority (200) on NS2 is higher than on NS1 (100).

Sample Configuration COPY

```
Settings on NS1

> add vrid 10 –Priority 100 –Preemption Enabled –sharing Enabled

Done

> add ns ip 203.0.113.10 –type VIP –vrid 10

Done

Settings on NS2

> add vrid 30 –Priority 200 –Preemption Enabled –sharing Enabled

Done

> add ns ip 2001:db8::5001 –type VIP –vrid 30

Done
```

Configuring Send to Master

Updated: 2013-08-29

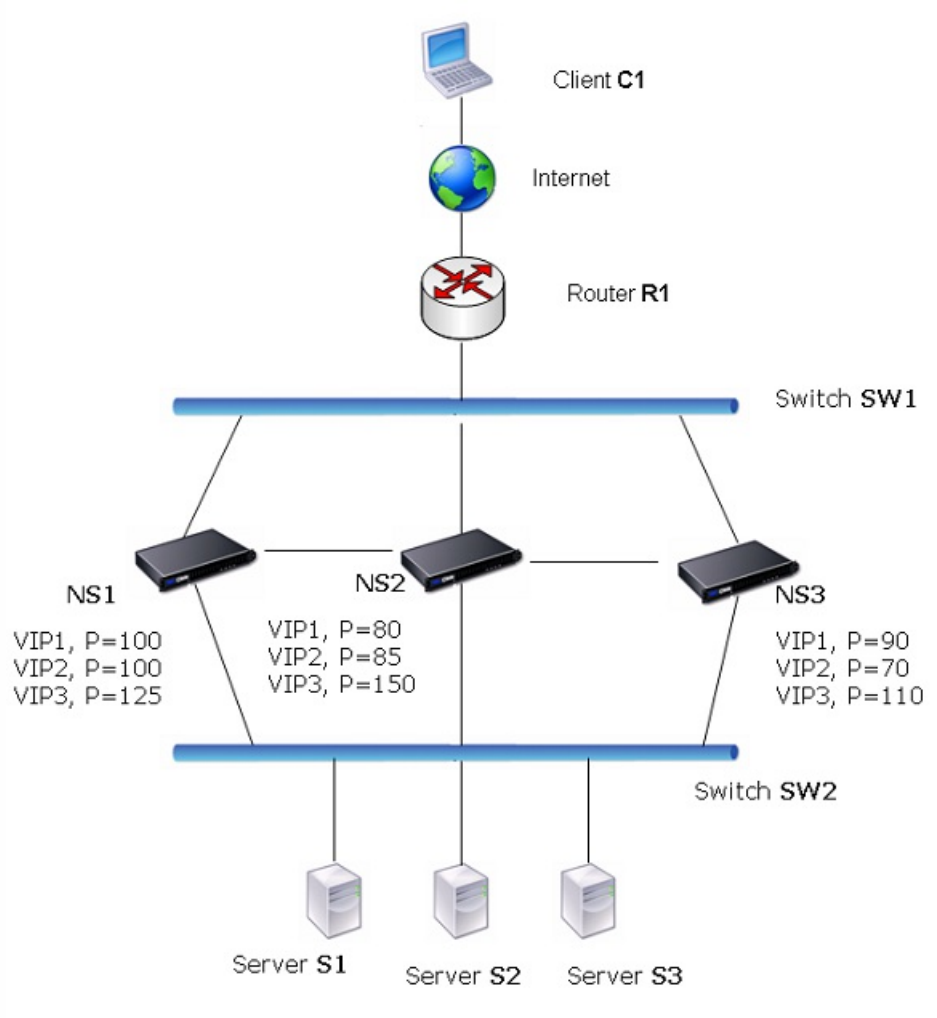
Usually, the traffic destined to a VIP reaches the NetScaler appliance on which the VIP is active, because an ARP request with the VIP and a VMAC on that appliance has reached the upstream router. But in some cases, such as static routes configured on the upstream router for the VIP subnet, or a topology that blocks this route, the traffic can reach a NetScaler appliance on which the VIP is in backup state. If you want this appliance to forward the data packets to the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

appliance on which the VIP is active, you need to enable the send to master option. This behavior is a per node setting and is disabled by default.

For example, in the following diagram, VIP1 is configured on NS1, NS2, and NS3 and is active on NS1. Under certain circumstances, traffic for VIP1 (active on NS1) may reach VIP1 on NS3. When the send to master option is enabled on NS3, NS3 forwards the traffic to NS1 through NS2 by using route entries for NS1.

Figure 1. An Active-Active Configuration with Send to Master Option Enabled



To enable send to master by using the command line interface

At the command prompt, type:

set vrIDParam -sendToMaster (ENABLED | DISABLED)

Example

```
> set vrIDParam -sendToMaster ENABLED
 Done
```

To enable send to master by using the configuration utility

1. Navigate to System > Network, in the Settings group, click Virtual Router Parameters.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Select the Send to Master option.

Configuring VRRP Communication Intervals

In an active-active deployment, all NetScaler nodes use the Virtual Router Redundancy Protocol (VRRP) to advertise their master VIP addresses and the corresponding priorities in VRRP advertisement packets (hello messages) at regular intervals.

VRRP uses the following communication intervals:

- **Hello Interval**. Interval between the VRRP hello messages that a node of a master VIP address sends to its peer nodes.
- **Dead Interval**. Time after which a node of a backup VIP address considers the state of the master VIP address as DOWN if VRRP hello messages are not received from the node of the master VIP address. After the dead interval, the backup VIP address takes over and becomes the master VIP address.

You can change these intervals to a desired value. Both of these communication intervals are per node setting for all VIP addresses in that node.

To configure VRRP communication intervals by using the command line interface

At the command prompt, type:

```
set vrIDParam [-helloInterval <msecs>] [-deadInterval <secs>]
```

Example

```
> set vrIDParam -helloInterval 500 -deadInterval 2
 Done
```

To configure VRRP communication intervals by using the configuration utility

1. Navigate to System > Network, in the Settings group, click Virtual Router Parameters.
2. Under Configure Virtual Router Parameter, set the Hello Interval and Dead Interval parameters.
3. Click OK.

Example 1: Nodes with the Same VRRP Dead Intervals

Consider an active-active deployment consisting of NetScaler ADCs NS1, NS2, and NS3. Virtual IP addresses VIP1, VIP2, VIP3 are configured on each of these ADCs. Because of their priorities, VIP1 is active on NS1, VIP2 is active on NS2, and VIP3 is active on NS3.

As shown in the table below, the dead interval is set to the same value (2 seconds) on all the three nodes.

The VRRP communication intervals (hello interval and dead interval) of a node apply to all the VRIDs configured on the node, and in turn apply to all VIP addresses associated with the VRIDs on the node.

On each node, the VIP addresses that are active (master) on that node use the hello interval, and the dead interval is used by the VIP addresses that are inactive (backup) on that node.

Preemption is disabled for the VIP addresses in all the three nodes.

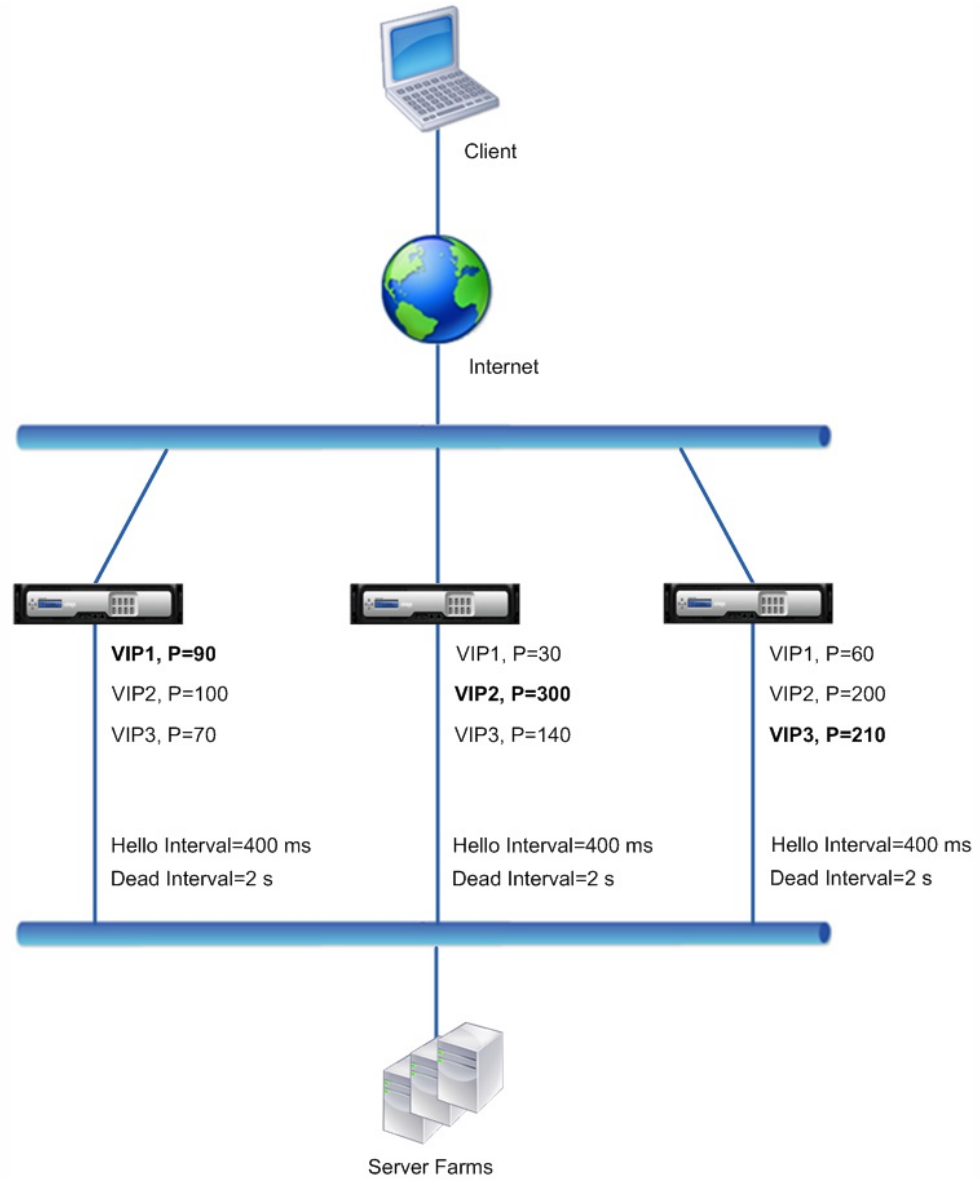
The following table lists the settings used in this example.

	Settings on NS1	**Settings on NS2**	**Settings on NS3**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VIP addresses			
VIP1 (for reference purposes only)	• IP address: 192.0.1.10 • VRID: 10 • Priority: 90	• IP address: 192.0.1.10 • VRID: 10 • Priority: 30	• IP address: 192.0.1.10 • VRID: 10 • Priority: 60
VIP2 (for reference purposes only)	• IP address: 192.0.1.20 • VRID: 20 • Priority: 100	• IP address: 192.0.1.20 • VRID: 20 • Priority: 300	• IP address: 192.0.1.20 • VRID: 20 • Priority: 200
VIP3 (for reference purposes only)	• IP address: 192.0.1.30 • VRID: 30 • Priority: 70	• IP address: 192.0.1.30 • VRID: 30 • Priority: 140	• IP address: 192.0.1.30 • VRID: 30 • Priority: 210
Communication Intervals			
Hello Interval	400 milliseconds	400 milliseconds	400 milliseconds
Dead Interval	2 seconds	2 seconds	2 seconds

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The execution flow is as follows:

1. NS1 sends hello messages at a set hello interval of 400 ms to NS2 and NS3 for the VIP1 address, because VIP1 is active (the master) on NS1. Similarly, NS2 sends hello messages for VIP2, and NS3 sends hello messages for VIP3.
2. On NS1, the set dead interval applies to VIP2 and VIP3, because they are inactive (backups) on NS1. Similarly, on NS2, the set dead interval applies to VIP1 and VIP3, and on NS3, the set dead interval applies to VIP1 and VIP2.
3. If NS1 goes down, NS2 and NS3 consider NS1 to be down if they receive no hello messages from NS1 for 2 seconds (the dead interval). VIP1 on NS3 takes over and becomes active (master), because its VRID priority (60) is higher than that of VIP1 of NS2 (30).

Example 2: Nodes with Different VRRP Dead Intervals

Consider a VRRP deployment similar to the deployment described in Example1 but with a different dead interval on each node (NS1, NS2, and NS3). Preemption is disabled for the VIP addresses in all the three nodes.

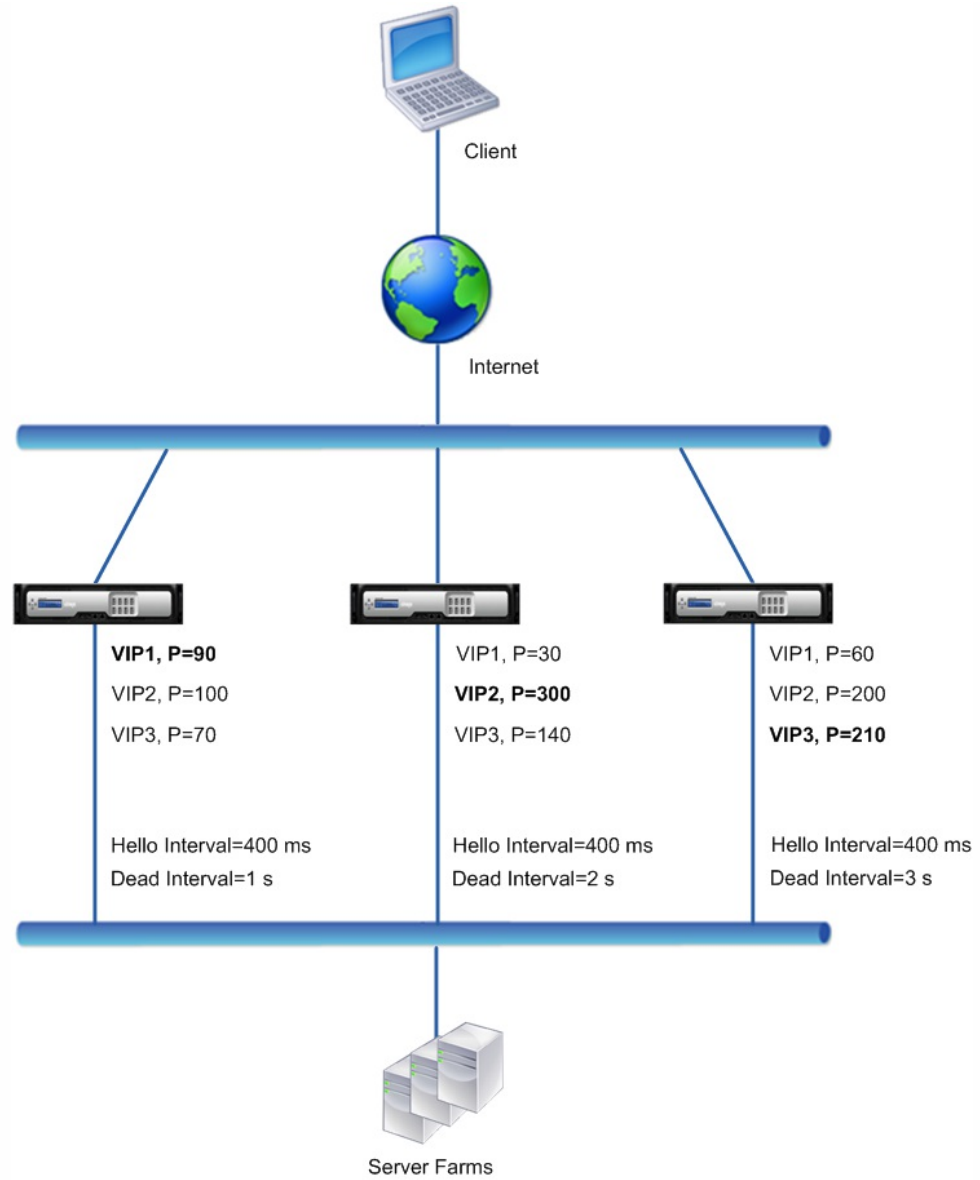
The following table lists the settings used in this example.

	Settings on NS1	**Settings on NS2**	**Settings on NS3**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

VIP addresses			
VIP1 (for reference purposes only)	• IP address: 192.0.1.10 • VRID: 10 • Priority: 90	• IP address: 192.0.1.10 • VRID: 10 • Priority: 30	• IP address: 192.0.1.10 • VRID: 10 • Priority: 60
VIP2 (for reference purposes only)	• IP address: 192.0.1.20 • VRID: 20 • Priority: 100	• IP address: 192.0.1.20 • VRID: 20 • Priority: 300	• IP address: 192.0.1.20 • VRID: 20 • Priority: 200
VIP3 (for reference purposes only)	• IP address: 192.0.1.30 • VRID: 30 • Priority: 70	• IP address: 192.0.1.30 • VRID: 30 • Priority: 140	• IP address: 192.0.1.30 • VRID: 30 • Priority: 210
Communication Intervals			
Hello Interval	400 milliseconds	400 milliseconds	400 milliseconds
Dead Interval	1 second	2 seconds	3 seconds

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The execution flow is as follows when NS1 goes down:

1. NS2 considers NS1 to be down after not receiving any hello messages from NS1 for 2 seconds (NS2's dead interval).
2. VIP1 on NS2 takes over and becomes active (master). NS2 now starts sending hello messages for VIP1.

Even though VIP1 on NS3 has a higher VRIP priority (60) than does VIP1 on NS2 (30), NS3's larger dead interval (3 seconds, vs. 2 seconds for NS2), prevents VIP1 on NS3 from taking over before VIP 1 on NS2 has already done so.

Example 3: Nodes with Different Dead Intervals and Preemption Enabled

Consider a VRRP deployment similar to the deployment described in Example1 but with different dead intervals on the three nodes, NS1, NS2, and NS3, and with preemption enabled for the VIP1 address on NS3.

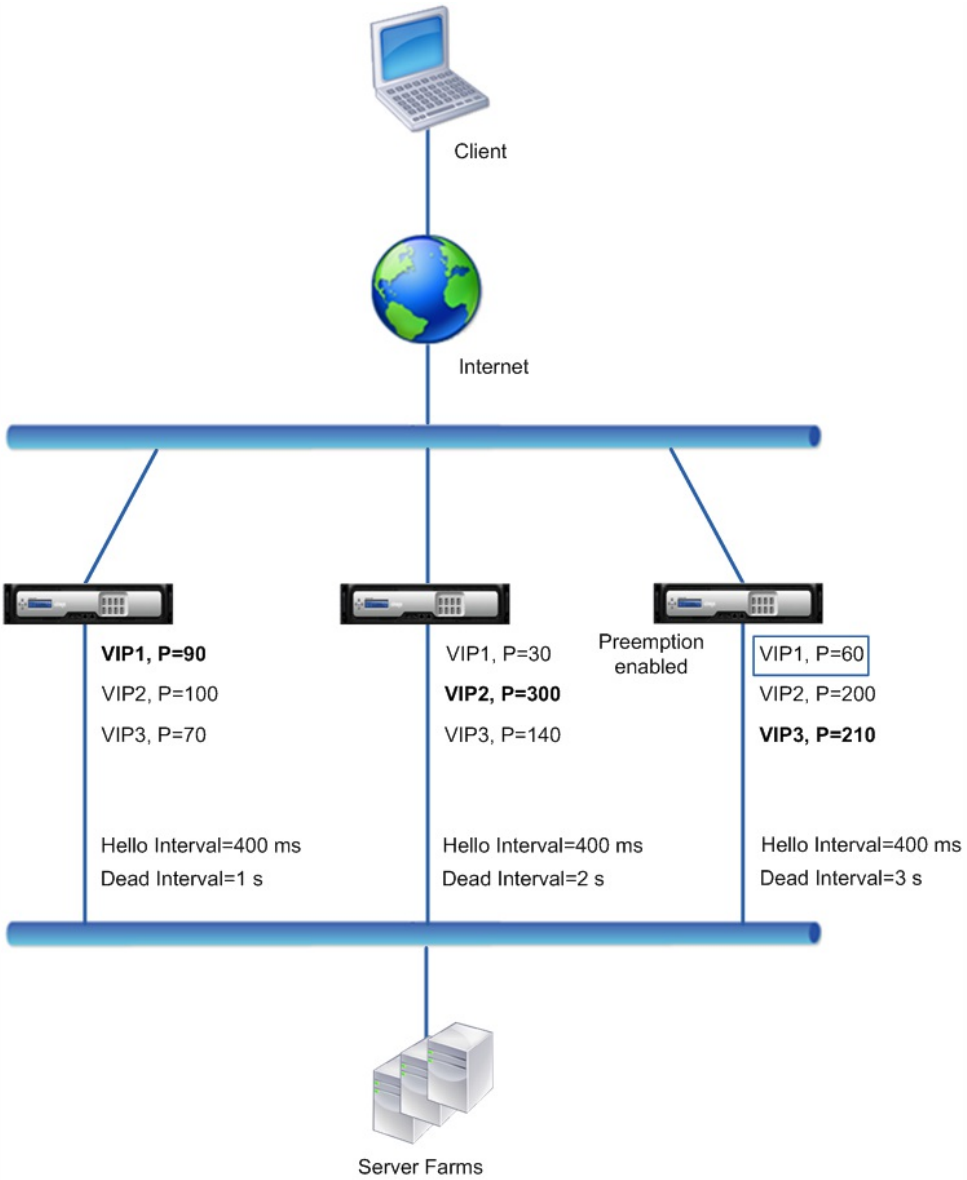
The following table lists the settings used in this example.

	Settings on NS1	**Settings on NS2**	**Settings on NS3**
VIP addresses			

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VIP1 (for reference purposes only)	• IP address: 192.0.1.10 • VRID: 10 • Priority: 90	• IP address: 192.0.1.10 • VRID: 10 • Priority: 30	• IP address: 192.0.1.10 • VRID: 10 • Priority: 60
VIP2 (for reference purposes only)	• IP address: 192.0.1.20 • VRID: 20 • Priority: 100	• IP address: 192.0.1.20 • VRID: 20 • Priority: 300	• IP address: 192.0.1.20 • VRID: 20 • Priority: 200
VIP3 (for reference purposes only)	• IP address: 192.0.1.30 • VRID: 30 • Priority: 70	• IP address: 192.0.1.30 • VRID: 30 • Priority: 140	• IP address: 192.0.1.30 • VRID: 30 • Priority: 210
Communication Intervals			
Hello Interval	400 milliseconds	400 milliseconds	400 milliseconds
Dead Interval	1 second	2 seconds	3 seconds

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The execution flow is as follows when NS1 goes down:

1. NS2 considers NS1 to down after not receiving any hello messages from NS1 for 2 seconds (NS2's set dead interval). At this time, NS3, with a dead interval of 3 seconds, does not consider NS1 to be down.
2. VIP1 on NS2 takes over and becomes active (master). NS2 now starts sending hello messages for VIP1.
3. Upon receiving hello messages from NS2 for VIP1, NS3 preempts NS2 for VIP1 because preemption is enabled for VIP1 of NS3 and the VRID priority (60) of VIP1 of NS3 is higher than that (30) of VIP1 of NS2.
4. VIP1 on NS3 takes over and becomes active (master). NS3 now starts sending hello messages for VIP1.

Changing the Priority of a VIP Address Automatically in an Active-Active Configuration (VRRP)

To ensure that a backup VIP address takes over as the master VIP before the node of the current master VIP address goes down completely, you can configure a node to change the priority of a VIP address when the state of an interface on the node changes. For example, the node reduces the priority of a VIP address when the state of an interface changes to DOWN, and increases the priority when the state of the interface changes to UP.
This feature is a per node configuration for each VIP address.

To configure this feature on a node for a VIP address, you set the Reduced Priority (trackifNumPriority) parameter, and then associate the interfaces whose state is to be tracked for changing the priority of the VIP address. When any of the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

associated interface's state changes to DOWN or UP, the node reduces or increases the priority of the VIP address by the configured Reduced Priority (trackifNumPriority) value.

To set reduced priority and bind interfaces to the virtual router ID by using the command line interface

At the command prompt, type:

- set vrID <id> [-trackifNumPriority <positive_integer>]
- bind vrID <id> -trackifNum <interface_name>
- show vrID <id>

Example

```
> set vrID 125 -trackifNumPriority 10
> bind vrID 125 -trackifNum 1/4 1/5
```

To set reduced priority and bind interfaces to the virtual router ID by using the configuration utility

1. Navigate to System > Network > VAMC and select a virtual router ID.
2. Under Configure VMAC, set the Reduced Priority parameter.
3. Select the Interfaces tracked for the vrid option and, under Associate Interfaces, add interfaces to the virtual router ID.

Example

Consider an active-active deployment consisting of NetScaler ADCs NS1, NS2, and NS3. Virtual IP addresses VIP1, VIP2, VIP3, and VIP4 are configured on each of these ADCs. Because of their priorities, VIP1 and VIP4 are active on NS1, VIP2 is active on NS2, and VIP3 is active on NS3.

To ensure that the active VIP addresses on NS1 are taken over by either NS2 or NS3 before NS1 goes down completely, interface-based health tracking is configured for the VIP1 and VIP4 addresses on NS1. Configuring interface-based health tracking for a VIP address includes associating the desired interfaces and setting the reduced priority (trackifNumPriority) parameter for the associated VRID of the VIP address. For example, on NS1, interfaces 1/2, 1/3, and 1/5 are associated to the VRID of VIP1, and reduced priority is set to 20.

Preemption is enabled for these VIP addresses in all three nodes.

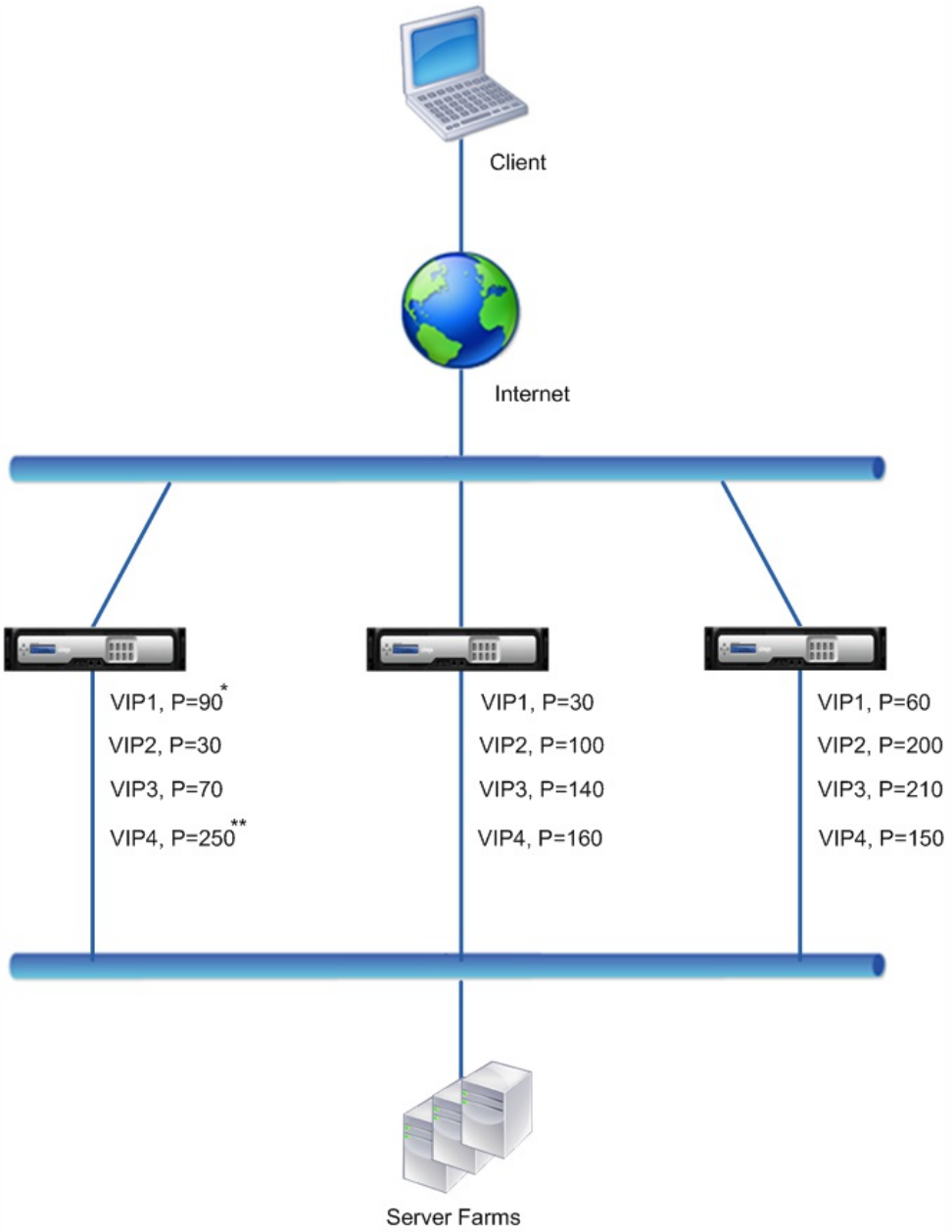
The following table lists the settings used in this example.

	Settings on NS1	**Settings on NS2**	**Settings on NS3**
VIP addresses			
VIP1 (for reference purposes only)	• IP address: 192.0.1.10 • VRID: 10 • Priority: 90	• IP address: 192.0.1.10 • VRID: 11 • Priority: 30	• IP address: 192.0.1.10 • VRID: 15 • Priority: 60

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VIP2 (for reference purposes only)	• IP address: 192.0.1.20 • VRID: 20 • Priority: 100	• IP address: 192.0.1.20 • VRID: 21 • Priority: 300	• IP address: 192.0.1.20 • VRID: 25 • Priority: 200
VIP3 (for reference purposes only)	• IP address: 192.0.1.30 • VRID: 30 • Priority: 70	• IP address: 192.0.1.30 • VRID: 31 • Priority: 140	• IP address: 192.0.1.30 • VRID: 35 • Priority: 210
VIP4 (for reference purposes only)	• IP address: 192.0.1.40 • VRID: 40 • Priority: 250 • Track interfaces: 1/5, 1/7 • Reduced priority: 55	• IP address: 192.0.1.40 • VRID: 41 • Priority: 160	• IP address: 192.0.1.40 • VRID: 45 • Priority: 150

© 1999-2017 Citrix Systems, Inc. All rights reserved.



* Packet Interfaces = 1/2, 1/3, 1/5
Reduced Priority = 20

** Packet Interfaces = 1/5, 1/7
Reduced Priority = 55

The execution flow is as follows on NS1 when multiple interface on NS1 goes down:

1. If interface 1/3 goes down, the priority of address VIP1 is reduced by 20 (VIP1's reduced priority value), because interface 1/3 is associated with VIP1:
   - Effective priority of VIP1 = (Current priority - reduced priority) = (90-20) = 70
2. Similarly, if interface 1/5 goes down, the priority of address VIP1 is further reduced:
   - Effective priority of VIP1 = (Current priority - reduced priority) = (70-20) = 50
3. At this point, the effective priority of VIP1 on NS1 is less than the priority of VIP1 on NS3. NS3 preempts NS1 for VIP1. VIP1 on NS3 takes over and becomes active (master).
4. Also, because interface 1/5 is also associated with VIP4, the priority of VIP4 is reduced by the VIP4's reduced priority value (55).

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Effective priority of VIP4 = (250 - 55) = 195

5. If interface 1/7 goes down, the priority of VIP4 is further reduced:
   - Effective priority of VIP4 = (Current priority - reduced priority) = (195-55) = 145
6. At this point, the effective priority of VIP4 on NS1 is less than the priority of VIP4 on NS2. NS2 preempts NS1 for VIP4. VIP4 on NS3 takes over and becomes active (master). This configuration ensures that none of the four VIP addresses are active on NS1 before it completely goes down.

Delaying Preemption

By default, a backup VIP address preempts the master VIP address immediately after its priority becomes higher than that of the master VIP. When configuring a backup VIP address, you can specify an amount of time by which to delay the preemption. Preemption delay time is a per-node setting for each backup VIP address.

The preemption delay setting for a backup VIP does not apply in the following conditions:

- The node of the master VIP goes down. In this case, the backup VIP takes over as the master VIP after the dead interval set on the backup VIP's node.
- The priority of the master VIP is set to zero. The backup VIP takes over as the master VIP after the dead interval set on the backup VIP's node.

Example: Delaying Preemption

Consider an active-active deployment consisting of NetScaler appliances NS1 and NS2. Virtual IP address VIP1 is configured on each of these appliances. Because of their priorities, VIP1 is master on NS2. Preemption is enabled and preemption delay time is set for VIP1 on these two nodes.

The following table lists the settings used in this example.

	Settings on NS1	Settings on NS2
VIP1 (for reference purposes only)	· IP address: 192.0.1.10 · VRID: 10 · Priority: 100 · Preemption: Enabled · Preemption delay time: 1000 seconds	· IP address: 192.0.1.10 · VRID: 10 · Priority: 200 · Preemption: Enabled · Preemption delay time: 2000 seconds
Dead Interval	1 Seconds	2 Seconds

Following are some examples of possible preemption behavior in this setup:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- If the priority of VIP1 on NS1 is set to a value (for example, 210) higher than that of VIP1 on NS2, VIP1 on NS1 takes over as master after its set preemption delay time (1000 secs).
- If a third node NS3 with the following VRRP settings is added to this deployment, VIP1 on NS3 becomes the master after its set preemption delay time (3000 secs).
  - VIP1
    - VRID: 30
    - IP address:
    - Priority = 300
    - Preemption delay time = 3000 seconds
- If NS2 goes down, VIP1 on NS1 takes over as master after 1 second (set dead interval on NS1). Preemption delay time for VIP1 on NS1 does not apply in this case.
- If NS2 goes down and NS1 restarts, VIP1 on NS1 becomes the master 1 second (set dead interval on NS1) after NS1 comes up. Preemption delay time for VIP1 on NS1 does not apply in this case.
- If the priority of VIP1 on NS2 is set to zero, VIP1 goes to standby mode. VIP1 on NS1 takes over as master after 1 second (set dead interval on NS1). Preemption delay time for VIP1 on NS1 does not apply in this case.

Configuration Steps

To configure preemption delay time for a VIP address, you set the preemption delay timer parameter of the associated VMAC address. You can the set this parameter when you add the address, or you can modify an existing VMAC address.

To configure preemption delay time by using the NetScaler command line

- To set the preemption delay time while adding a VMAC, at the command prompt, type:
  - **add vrID** <id> -**preemptiondelaytimer** <secs>
  - **show vrID**

- To set the preemption delay time while modifying a VMAC, at the command prompt, type:
  - **set vrID** <id> -**preemptiondelaytimer** <secs>
  - **show vrID**

To configure preemption delay time by using the configuration utility

1. Navigate to **System** > **Network** > **VMAC**.
2. On the **VMAC** tab. While adding a new VMAC, or editing an existing VMAC, set the **Preemption Delay Timer** parameter.

Sample Configuration COPY

```
The following configuration uses the settings listed in table in section Example: Delaying Preemption.

Settings on NS1

> set vrid param –deadInterval 1

Done
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> add ns ip 192.0.1.10 255.255.255.255 –type VIP

Done

> add vrid 10 –Priority 100 –Preemption Enable –preemptiondelaytimer 1000

Done

> bind ns ip 192.0.1.10 255.255.255.255 –vrid 10

Done

Settings on NS2

> set vrid param –deadInterval 2

Done

> add ns ip 192.0.1.10 255.255.255.255 –type VIP

Done

> add vrid 20 –Priority 200 –Preemption Enable –preemptiondelaytimer 2000

Done

> set ns ip 192.0.1.10 255.255.255.255 –vrid 10

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Keeping a VIP Address in the Backup State

You can force a VIP address to always stay in backup state. This operation is helpful in maintenance or testing of a VRRP deployment.

When a VIP address is forced to stay in backup state, it does not participate in VRRP state transitions. Also, it cannot become master even if all other nodes go down.

To force a VIP address to stay in backup state, you set the priority of the associated VMAC address to zero. To ensure that none of the VIP addresses of a node handle traffic during a maintenance process on the node, set all the priorities to zero.

You can the set the priority of a VMAC address while adding or modifying the address.

To force a VIP address to stay in the backup state by using the NetScaler command line

- To set the priority while adding a VMAC, at the command prompt, type:
  - **add vrID** <id> -**priority** 0
  - **show vrID**
- To set the priority while modifying a VMAC, at the command prompt, type:
  - **set vrID** <id> -**priority** 0
  - **show vrID**

To force a VIP address to stay in backup state by using the configuration utility

1. Navigate to **System** > **Network** > **VMAC**.
2. On the **VMAC** tab, while adding a new VMAC or editing an existing VMAC, set the **Priority** parameter to zero.

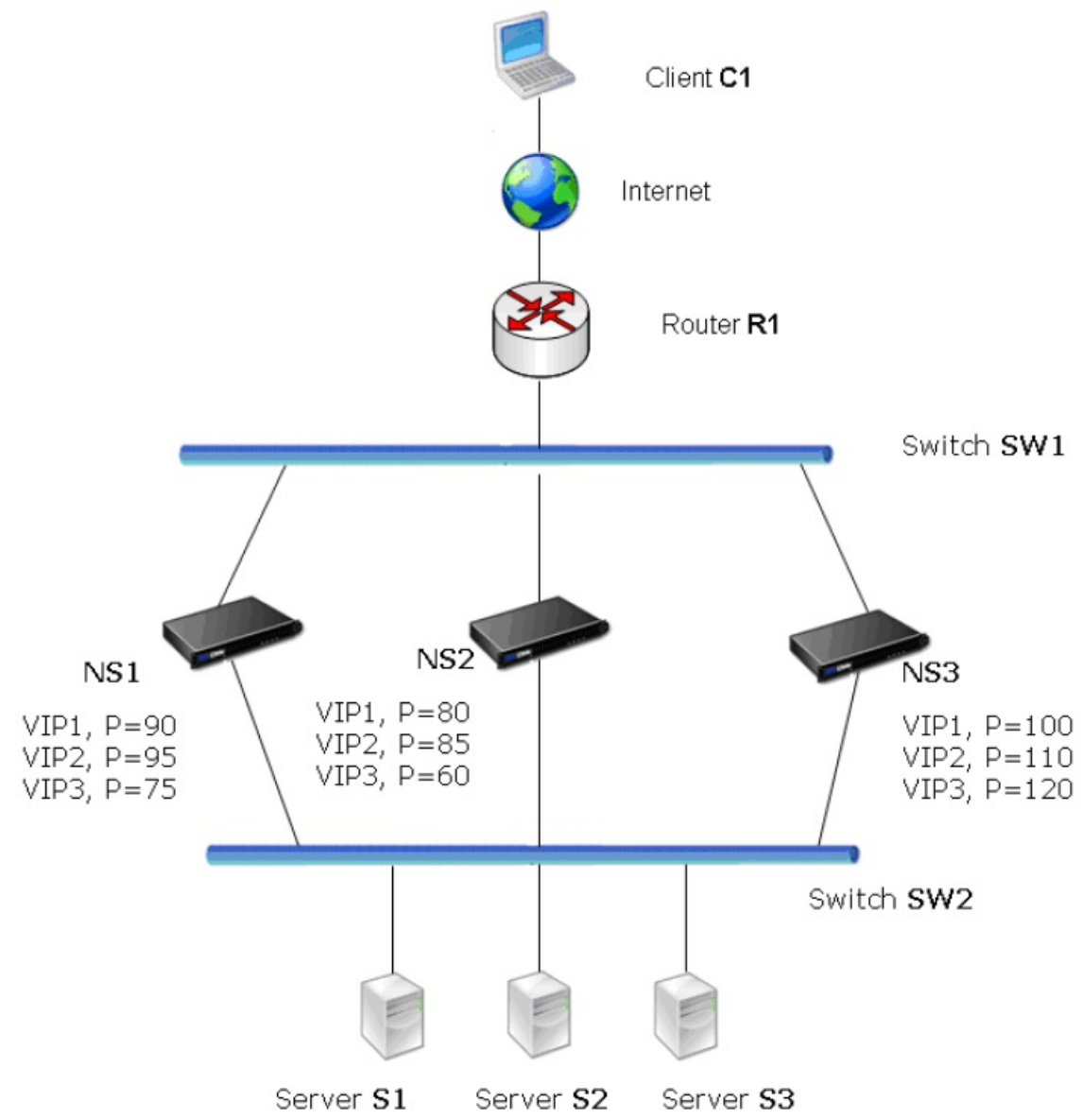
An Active-Active Deployment Scenario

Following is an example of a possible active-active deployment scenario.

In the following diagram, VIP1, VIP 2 and VIP3 are configured on all three appliances, NS1, NS2, and NS3. Base Priorities for each VIPs are as shown in the diagram. Health tracking is disabled for each VIP. The priorities of VIPs are set so that VIP1, VIP2, and VIP3 are active on NS3. If NS3 fails, VIP1, VIP2, and VIP3 become active on NS1.

Figure 2. An Active-Active Deployment Scenario

© 1999-2017 Citrix Systems, Inc. All rights reserved.



You can force a VIP address to always stay in backup state. This operation is helpful in maintenance or testing of a VRRP deployment.

When a VIP address is forced to stay in backup state, it does not participate in VRRP state transitions. Also, it cannot become master even if all other nodes go down.

To force a VIP address to stay in backup state, you set the priority of the associated VMAC address to zero. To ensure that none of the VIP addresses of a node handle traffic during a maintenance process on the node, set all the priorities to zero.

You can the set the priority of a VMAC address while adding or modifying the address.

To force a VIP address to stay in the backup state by using the NetScaler command line

- To set the priority while adding a VMAC, at the command prompt, type:
  - **add vrID** <id> -**priority** 0
  - **show vrID**
- To set the priority while modifying a VMAC, at the command prompt, type:
  - **set vrID** <id> -**priority** 0

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- **show vrID**

To force a VIP address to stay in backup state by using the configuration utility

1. Navigate to **System** > **Network** > **VMAC**.
2. On the **VMAC** tab, while adding a new VMAC or editing an existing VMAC, set the **Priority** parameter to zero.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using the Network Visualizer

Aug 29, 2013

The network visualizer shows a graphical view of all the interfaces, channels, VLANs, IP addresses, and bindings to VLANs on a NetScaler appliance. An enabled interface or channel has a black label. A disabled interface or channel has a red label.

This complete picture of the appliance's network connections can be useful for detecting flaws in the network design and for optimizing the network. It can also help a new administrator easily understand the appliance's network configuration.

To open the Network Visualizer

Navigate to **System** > **Network**. In **Monitor Connections**, click **Network Visualizer**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Link Layer Discovery Protocol

Jun 11, 2014

The NetScaler ADC supports the industry standard (EEE 802.1AB) Link Layer Discovery Protocol (LLDP). LLDP is a layer 2 protocol that enables the NetScaler ADC to advertise its identity and capabilities to the directly connected devices, and also learn the identity and capabilities of these neighbour devices. Using LLDP, the NetScaler ADC transmits and receives information in the form of LLDP messages known as LLDP packet data units (LLDPUs).

An LLDPU is a sequence of type, length, value (TLV) information elements. Each TLV holds a specific type of information about the device that transmits the LLDPDU. The NetScaler ADC sends the following TLVs in each LLDPU:

- Chassis ID
- Port ID
- Time-to-live value
- System name
- System description
- Port description
- System capabilities
- Management address
- Port VLAN ID
- Link aggregation

Note: You cannot specify the TLVs to be sent in LLDP messages.
NetScaler interfaces support the following LLDP modes:

- **NONE**. The interface neither receives from nor transmits LLDP messages to the directly connected device.
- **TRANSMITTER**. The interface transmits LLDP messages to the directly connected device but does not receive LLPD messages from the directly connected device.
- **RECEIVER**. The interface receives LLDP messages from the directly connected device but does transmit LLPD messages to the directly connected device.
- **TRANSCEIVER**. The interface transmits LLDP messages to and receives LLDP messages from the directly connected device.

The LLDP mode of an interface depends on the LLDP mode configured at the global and the interface levels. The following table shows the modes resulting from the available combinations of global- and interface-level settings:

Interface Level LLDP mode	**Global Level LLDP mode**			
	NONE	**TRANSMITTER**	**RECEIVER**	**TRANSCEIVER**
NONE	NONE	NONE	NONE	NONE
TRANSMITTER	NONE	TRANSMITTER	NONE	TRANSMITTER
RECEIVER	NONE	NONE	RECEIVER	RECEIVER

© 1999-2017 Citrix Systems, Inc. All rights reserved.

TRANSCEIVER	NONE	TRANSMITTER	RECEIVER	TRANSCEIVER

Transmitting LLDP messages

The NetScaler ADC transmits LLDPUs from interfaces that are operating in either TRANSMITTER or TRANSCEIVER LLDP mode.

Following are the global LLDP transmitting parameters on the NetScaler ADC:

- **Timer**. Interval, in seconds, between LLDPUs that the NetScaler ADC sends to a directly connected device.
- **Holdtime Multiplier**. A multiplier for calculating the duration for which the receiving device stores the LLDP information in its database before discarding or removing it. The duration is calculated as the **Holdtime Multiplier** parameter value multiplied by the Timer parameter value.

Receiving LLDP Messages

The NetScaler ADC stores the LLDPDU information in its Management Information base (MIB). The stored LLDP information is classified or grouped under the ID of the interface that received the LLDPDU. The NetScaler ADC retains this LLDP information for the duration specified in the received LLDPU.

If the ADC receives another LLDPDU on an interface before the stored LLDP information for that interface is discarded, the ADC replaces the stored LLDP information for that interface with information in the new LLDPDU.

Configuration Steps

Configuring LLDP on a NetScaler appliance consists of the following tasks:

1. Set global level LLDP parameters. In this task, you set the global LLDP parameters such as LLDP Timer, Hold Time Multiplier, and LLDP mode.
2. Set the interface level LLDP parameters. In this task, you set the LLDP mode for an interface.
3. (Optional) Display neighbor-device information. You can display the neighbor-device LLDP information collected on all of the NetScaler ADC's interfaces, or just the LLDP information collected on specified interfaces. If you do not specify an interface, the information is shown for all interfaces.

Following are the prerequisites for configuring LLDP on a NetScaler ADC:

1. Make sure that you understand the standard LLDP protocol (IEEE 802.1AB).
2. Verify that you have configured LLDP on the desired directly connected devices.

To set global level LLDP parameters by using the command line interface

At the command prompt, type:

- set lldp param [-[-holdtimeTxMult <positive_integer>][-timer <positive_integer>] [-Mode <Mode>]
- show lldp param

To set the global level LLDP parameters by using configuration utility

1. Navigate to System > Network, and click Configure LLDP Parameters.
2. Set the following parameters:
   - Hold Timer Multiplier
   - Timer
   - Mode

To configure an interface for LLDP by using the command line interface

At the command prompt, type:

- set interface <id> -lldpmode <lldpmode>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- show interface <id>

To configure an interface for LLDP by using configuration utility

Navigate to System > Network > Interfaces, open the interface, and set the LLDP mode parameter.

To display neighbor device information by using the command line interface

At the command prompt, type one of the following commands:

- show lldp neighbors
- show lldp neighbors <ifnum>

To display neighbor device information by using configuration utility

Navigate to System > Network > Interfaces and, in the Action list, select View LLDP Neighbors.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Jumbo Frames

Jun 20, 2014

NetScaler appliances support receiving and transmitting jumbo frames containing up to 9216 bytes of IP data. Jumbo frames can transfer large files more efficiently than it is possible with the standard IP MTU size of 1500 bytes.

A NetScaler appliance can use jumbo frames in the following deployment scenarios:

- Jumbo to Jumbo. The appliance receives data as jumbo frames and sends it as jumbo frames.
- Non-Jumbo to Jumbo. The appliance receives data as regular frames and sends it as jumbo frames.
- Jumbo to Non-Jumbo. The appliance receives data as jumbo frames and sends it as regular frames.

The NetScaler appliance supports jumbo frames in a load balancing configuration for the following protocols:

- TCP
- Any protocol over TCP (for example, HTTP)
- SIP
- RADIUS

This document includes the following information:

- Configuring Jumbo Frames Support on a NetScaler Appliance
- Use Case 1 - Jumbo to Jumbo Setup
- Use Case 2 - Non-Jumbo to Jumbo Setup
- Use Case 3 - Coexistence of Jumbo and Non-Jumbo flows on the Same Set of Interfaces

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Jumbo Frames Support on a NetScaler Appliance

Apr 10, 2014

To enable the NetScaler appliance to support jumbo frames, you set the MTU to more than 1500 on interfaces or LA channels, and on VLANs on which you want the NetScaler appliance to support jumbo frames.

Points to consider before setting the MTU of interfaces, LA channels, or VLANs on a NetScaler appliance

1. Jumbo frames are not supported on NetScaler MPX 15000 and MPX 17000 Platforms.
2. When you create an LA channel, the channel takes the MTU of the first bound interface if no MTU is specified for the channel.
3. The MTU for a channel is propagated to all the bound interfaces.
4. When an interface is bound to the channel whose MTU is different from the interface's MTU, the interface goes onto the inactive list.
5. When you change the MTU of a member interface, the interface goes onto the inactive list.
6. When an interface is unbound from the channel, the interface retains the MTU value of the channel.
7. You can set the MTU for an interface, channel, or VLAN to a value in the range of 1500-9216.
8. You cannot set the MTU on the default VLAN. The NetScaler appliance uses the MTU of the interface through which it receives or sends data from or to the default VLAN.
9. For TCP based traffic on a load balancing configuration on a NetScaler appliance, MSSs are set accordingly at each end point for supporting jumbo frames:
   - For a connection between a client and a load balancing virtual server on the NetScaler appliance, the MSS on the NetScaler appliance is set in a TCP profile, which is then bound to the load balancing virtual server.
   - For a connection between the NetScaler appliance and a server, the MSS on NS1 is set in a TCP profile, which is then bound to the service representing the server on the NetScaler appliance.
   - By default, a TCP profile nstcp_default_profile is bound to all TCP based load balancing servers and services on the NetScaler appliance.
   - For supporting jumbo frames, you can either change the MSS value of the TCP profile nstcp_default_profile, or create a custom TCP profile and set its MSS accordingly, and then bind the custom TCP profile to the desired load balancing virtual servers and services.
   - The default MSS value of any TCP profile is 1460.

To set the MTU of an interface by using the command line interface

At the command prompt, type:

- set interface <id> -mtu <positive_integer>
- show interface <id>

Example

```
> set interface 10/1 –mtu 9000
Done
```

To set the MTU of a channel by using the command line interface

At the command prompt, type:

- set channel <id> -mtu <positive_integer>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- show channel <id>

Example

```
> set channel LA/1 –mtu 9000
Done
```

To set the MTU of a VLAN by using the command line interface

At the command prompt, type:

- add vlan <id> -mtu <positive_integer>
- show vlan <id>

Example

```
> set vlan 20 –mtu 9000
Done
```

To set the MTU of an interface by using the configuration utility

Navigate to System > Network > Interfaces, open the interface, and set the Maximum Transmission Unit parameter.

To set the MTU of a channel by using the configuration utility

Navigate to System > Network > Channels, open the channel, and set the Maximum Transmission Unit parameter.

To set the MTU of a VLAN by using the configuration utility

Navigate to System > Network > VLANs, open the VLAN, and set the Maximum Transmission Unit parameter.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 1 – Jumbo to Jumbo Setup

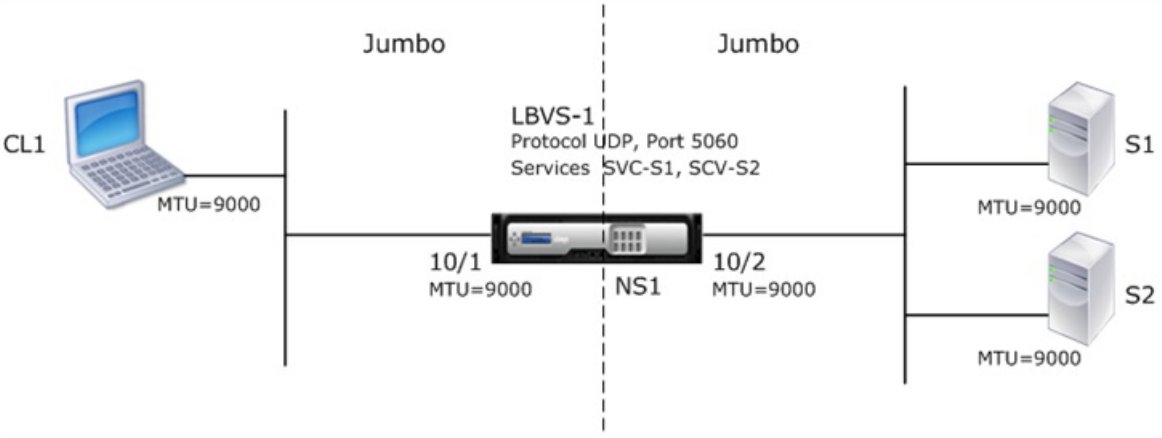
May 06, 2014

Consider an example of a jumbo to jumbo setup in which SIP load balancing virtual server LBVS-1, configured on NetScaler appliance NS1, is used to load balance SIP traffic across servers S1 and S2. The connection between client CL1 and NS1, and the connection between NS1 and the servers support jumbo frames.

Interface 10/1 of NS1 receives or sends traffic from or to client CL1. Interface 10/2 of NS1 receives or sends traffic from or to server S1 or S2. Interfaces 10/1 and 10/2 of NS1 are part of VLAN 10 and VLAN 20, respectively.

For supporting jumbo frames, the MTU is set to 9216, on NS1, for interfaces 10/1, 10/2, and VLANs VLAN 10, VLAN 20.

All other network devices, including CL1, S1, S2, in this setup example are also configured for supporting jumbo frames.



The following table lists the settings used in the example.

Entity	Name	Details
IP address of client CL1		192.0.2.10
IP address of servers	S1	198.51.100.19
	S2	198.51.100.20
SNIP address on NS1		198.51.100.18
MTU specified for interfaces and VLANs on NS1	10/1	9000
	10/2	9000
	VLAN 10	9000
	VLAN 20	9000

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity	Name	Details
Services on NS1 representing servers	SVC-S1	• IP address: 198.51.100.19 • Protocol: SIP • Port: 5060
	SVC-S2	• IP address: 198.51.100.20 • Protocol: SIP • Port: 5060
Load balancing virtual server on VLAN 10	LBVS-1	• IP address: 203.0.113.15 • Protocol: SIP • Port:5060 • Bound services: SVC-S1, SVC-S2

Following is the traffic flow of CL1's request to NS1:

1. CL1 creates a 20000-byte SIP request to send to LBVS-1 of NS1.
2. CL1 sends the request data in IP fragments to LBVS-1. The size of each IP fragment is either equal to or less than the MTU (9000) set on the interface from which CL1 sends these fragments to NS1.
   - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000
   - Size of the second IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
   - Size of the last IP fragment=[IP header + SIP data segment] = [20 + 2048] = 2068
3. NS1 receives the request IP fragments at interface 10/1. NS1 accepts these fragments, because the size of each of these fragments is equal to or less than the MTU (9000) of interface 10/1.
4. NS1 reassembles these IP fragments to form the 20000-byte SIP request. NS1 processes this request.
5. LBVS-1's load balancing algorithm selects server S1.
6. NS1 sends the request data in IP fragments to S1. The size of each IP fragment is either equal or less than the MTU (9000) of the interface 10/2, from which NS1 sends these fragments to S1. The IP packets are sourced with a SNIP address of NS1.
   - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000
   - Size of the second IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
   - Size of the last IP fragment=[IP header + SIP data segment] = [20 + 2048] = 2068

Following is the traffic flow of S1's response to CL1 in this example:

1. Server S1 creates a 30000-byte SIP response to send to the SNIP address of NS1.
2. S1 sends the response data in IP fragments to the SNIP address of NS1. The size of each IP fragment is either equal to or less than the MTU (9000) set on the interface from which S1 sends these fragments to NS1.
   - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000
   - Size of the second and third IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
   - Size of the last IP fragment=[IP header + SIP data segment] = [20 + 3068] = 3088
3. NS1 receives the response IP fragments at interface 10/2. NS1 accepts these fragments, because the size of each fragment is equal to or less than the MTU (9000) of interface 10/2.
4. NS1 reassembles these IP fragments to form the 30000-byte SIP response. NS1 processes this response.
5. NS1 sends the response data in IP fragments to CL1. The size of each IP fragment is either equal or less than the MTU (9000) of the interface 10/1, from which NS1 sends these fragments to CL1. The IP fragments are sourced with LBVS-1's IP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000
- Size of the second and third IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
- Size of the last IP fragment=[IP header + SIP data segment] = [20 + 3068] = 3088

Configuration Tasks

Updated: 2014-01-16

The following table list the tasks, NetScaler commands, and examples for creating the required configuration on the NetScaler appliance.

Tasks	NetScaler Command Syntax	Examples
Set the MTU of the desired interfaces for supporting jumbo frames	set interface <id> -mtu <positive_integer> show interface <id>	set int 10/1 -mtu 9000 set int 10/2 -mtu 9000
Create VLANs and set the MTU of the desired VLANs for supporting jumbo frames	add vlan <id> -mtu <positive_integer> show vlan <id>	add vlan 10 -mtu 9000 add vlan 20 -mtu 9000
Bind interfaces to VLANs	bind vlan <id> -ifnum <interface_name> show vlan <id>	bind vlan 10 -ifnum 10/1 bind vlan 20 -ifnum 10/2
Add a SNIP address	add ns ip <IPAddress> <netmask> -type SNIP show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP
Create services representing SIP servers	add service <serviceName> <ip> SIP_UDP <port> show service <name>	add service SVC-S1 198.51.100.19 SIP_UDP 5060 add service SVC-S2 198.51.100.20 SIP_UDP 5060
Create SIP load balancing virtual servers and bind the services to it	add lb vserver <name> SIP_UDP <ip> <port> bind lb vserver <vserverName> <serviceName> show lb vserver <name>	add lb vserver LBVS-1 SIP_UDP 203.0.113.15 5060 bind lb vserver LBVS-1 SVC-S1 bind lb vserver LBVS-1 SVC-S2
Save the configuration	save ns config show ns config	

1.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 2 – Non-Jumbo to Jumbo Setup

May 06, 2014

Consider an example of a regular to jumbo setup in which load balancing virtual server LBVS-1, configured on a NetScaler appliance NS1, is used to load balance traffic across servers S1 and S2. The connection between client CL1 and NS1 supports regular frames, and the connection between NS1 and the servers supports jumbo frames.

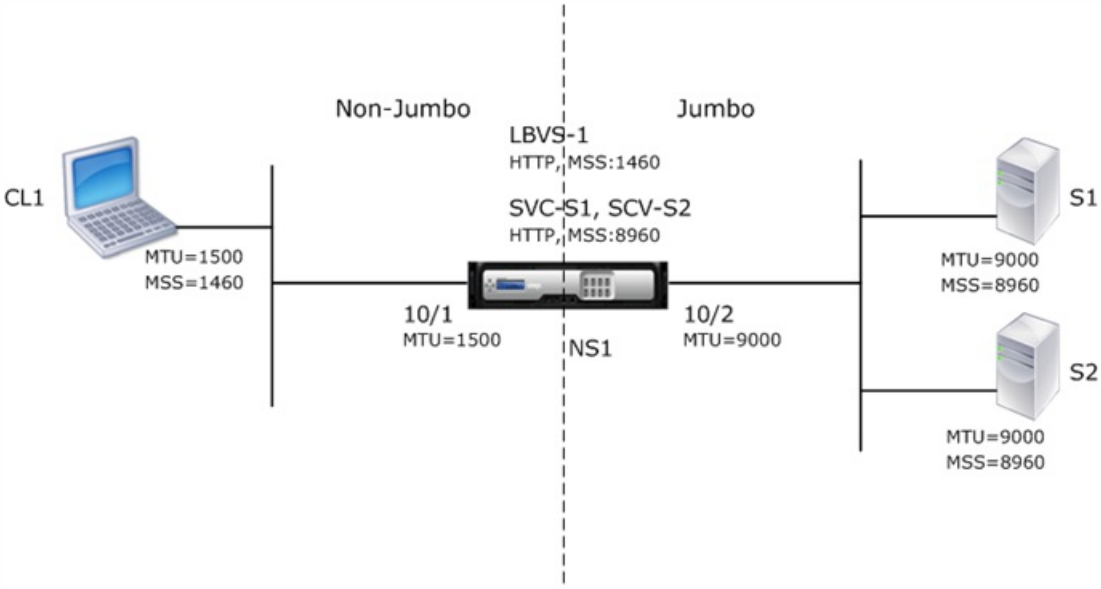
Interface 10/1 of NS1 receives or sends traffic from or to client CL1. Interface 10/2 of NS1 receives or sends traffic from or to server S1 or S2.

Interfaces 10/1 and 10/2 of NS1 are part of VLAN 10 and VLAN 20, respectively. For supporting only regular frames between CL1 and NS1, the MTU is set to the default value of 1500 for both interface 10/1 and VLAN 10

For supporting jumbo frames between NS1 and the servers, the MTU is set to 9000 for interface 10/2 and VLAN 20. Servers and all other network devices between NS1 and the servers are also configured for supporting jumbo frames.

Since HTTP traffic is based on TCP, MSSs are set accordingly at each end point for supporting jumbo frames.

- For supporting jumbo frames for the connection between a SNIP address of NS1 and S1 or S2, the MSS on NS1 is set accordingly in a custom TCP profile, which is bound to the services (SVC-S1 and SVC-S1) representing S1 and S2 on NS1.
- For supporting only regular frames for the connection between CL1 and virtual server LBVS-1 of NS1, default TCP profile nstcp_default_profile is used that is by default bound to LBVS-1 and has the MSS set to the default value of 1460.



The following table lists the settings used in this example.

Entity	Name	Details
IP address of client CL1	CL1	192.0.2.10
IP address of servers	S1	198.51.100.19
	S2	198.51.100.20

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity	Name	Details
SNIP address on NS1		198.51.100.18
MTU specified for interfaces and VLANs on NS1	10/1	1500
	10/2	9000
	VLAN 10	1500
	VLAN 20	9000
Default TCP profile	nstcp_default_profile	MSS:1460
Custom TCP profile	NS1-SERVERS-JUMBO	MSS: 8960
Services on NS1 representing servers	SVC-S1	• IP address: 198.51.100.19 • Protocol: HTTP • Port: 80 • TCP profile: NS1-SERVERS-JUMBO (MSS: 8960)
	SVC-S2	• IP address: 198.51.100.20 • Protocol: HTTP • Port: 80 • TCP profile: NS1-SERVERS-JUMBO (MSS: 8960)
Load balancing virtual server on VLAN 10	LBVS-1	• IP address = 203.0.113.15 • Protocol: HTTP • Port:80 • Bound services: SVC-S1, SVC-S2 • TCP Profile: nstcp_default_profile (MSS:1460)

Following is the traffic flow of CL1's request to S1 in this example:

1. Client CL1 creates a 200-byte HTTP request to send to virtual server LBVS-1 of NS1.
2. CL1 opens a connection to LBVS-1 of NS1. CL1 and NS1 exchange their respective TCP MSS values while establishing the connection.
3. Because NS1's MSS is larger than the HTTP request, CL1 sends the request data in a single IP packet to NS1.
   Size of the request packet = [IP Header + TCP Header + TCP Request] = [20 + 20 + 200] = 240
4. NS1 receives the request packet at interface 10/1 and then processes the HTTP request data in the packet.
5. LBVS-1's load balancing algorithm selects server S1, and NS1 opens a connection between one of its SNIP addresses and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

S1. NS1 and CL1 exchange their respective TCP MSS values while establishing the connection.

6. Because S1's MSS is larger than the HTTP request, NS1 sends the request data in a single IP packet to S1.
   Size of the request packet = [IP Header + TCP Header + [TCP Request] = [20 + 20 + 200] = 240

Following is the traffic flow of S1's response to CL1 in this example:

1. Server S1 creates an 18000-byte HTTP response to send to the SNIP address of NS1.
2. S1 segments the response data into multiples of NS1's MSS and sends these segments in IP packets to NS1. These IP packets are sourced from S1's IP address and destined to the SNIP address of NS1.
   - Size of the first two packet = [IP Header + TCP Header + (TCP segment=NS1's MSS size)] = [20 + 20 + 8960] = 9000
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 2080] = 2120
3. NS1 receives the response packets at interface 10/2.
4. From these IP packets, NS1 assembles all the TCP segments to form the HTTP response data of 18000 bytes. NS1 processes this response.
5. NS1 segments the response data into multiples of CL1's MSS and sends these segments in IP packets, from interface 10/1, to CL1. These IP packets are sourced from LBVS-1's IP address and destined to CL1's IP address.
   - Size of all packets except the last one = [IP Header + TCP Header + (TCP payload=CL1's MSS size)] = [20 + 20 + 1460] = 1500
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 480] = 520

Configuration Tasks

Updated: 2014-01-16

The following table list the tasks, NetScaler commands, and examples for creating the required configuration on the NetScaler appliance.

Tasks	NetScaler Command line Syntax	Examples
Set the MTU of the desired interfaces for supporting jumbo frames	set interface <id> -mtu <positive_integer> show interface <id>	set int 10/1 -mtu 1500 set int 10/2 -mtu 9000
Create VLANs and set the MTU of the desired VLANs for supporting jumbo frames	add vlan <id> -mtu <positive_integer> show vlan <id>	add vlan 10 -mtu 1500 add vlan 20 -mtu 9000
Bind interfaces to VLANs	bind vlan <id> -ifnum <interface_name> show vlan <id>	bind vlan 10 -ifnum 10/1 bind vlan 20 -ifnum 10/2
Add a SNIP address	add ns ip <IPAddress> <netmask> -type SNIP show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP
Create services representing HTTP servers	add service <serviceName> <ip> HTTP <port> show service <name>	add service SVC-S1 198.51.100.19 http 80 add service SVC-S2 198.51.100.20 http 80
Create HTTP load balancing virtual servers and bind the services to it	add lb vserver <name> HTTP <ip> <port> bind lb vserver <vserverName> <serviceName> show lb vserver <name>	add lb vserver LBVS-1 http 203.0.113.15 80 bind lb vserver LBVS-1 SVC-S1 bind lb vserver LBVS-1 SVC-S2
Create a custom TCP profile and set its MSS for	add tcpProfile <name> -mss <positive_integer> show tcpProfile	add tcpprofile NS1-SERVERS-JUMBO -mss 8960

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Tasks	NetScaler Command line Syntax	Examples
supporting jumbo frames	<name>	
Bind the custom TCP profile to the desired services	set service <Name> -tcpProfileName <string> show service <name>	set service SVC-S1 -tcpProfileName NS1-SERVERS-JUMBO set service SVC-S2 -tcpProfileName NS1-SERVERS-JUMBO
Save the configuration	save ns config show ns config	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 3 – Coexistence of Jumbo and Non-Jumbo flows on the Same Set of Interfaces

May 06, 2014

Consider an example in which load balancing virtual servers LBVS-1 and LBVS-2 are configured on NetScaler appliance NS1. LBVS-1 is used to load balance HTTP traffic across servers S1 and S2, and LBVS-2 is used to load balance traffic across servers S3 and S4.

CL1 is on VLAN 10, S1 and S2 are on VLAN20, CL2 is on VLAN 30, and S3 and S4 are on VLAN 40. VLAN 10 and VLAN 20 support jumbo frames, and VLAN 30 and VLAN 40 support only regular frames.

In other words, the connection between CL1 and NS1, and the connection between NS1 and server S1 or S2 support jumbo frames. The connection between CL2 and NS1, and the connection between NS1 and server S3 or S4 support only regular frames.

Interface 10/1 of NS1 receives or sends traffic from or to clients. Interface 10/2 of NS1 receives or sends traffic from or to the servers.

Interface 10/1 is bound to both VLAN 10 and VLAN 30 as a tagged interface, and interface 10/2 is bound to both VLAN 20 and VLAN 40 as a tagged interface.

For supporting jumbo frames, the MTU is set to 9216 for interfaces 10/1 and 10/2.

On NS1, the MTU is set to 9000 for VLAN 10 and VLAN 20 for supporting jumbo frames, and the MTU is set to the default value of 1500 for VLAN 30 and VLAN 40 for supporting only regular frames.

The effective MTU on a NetScaler interface for VLAN tagged packets is of the MTU of the interface or the MTU of the VLAN, whichever is lower. For example:

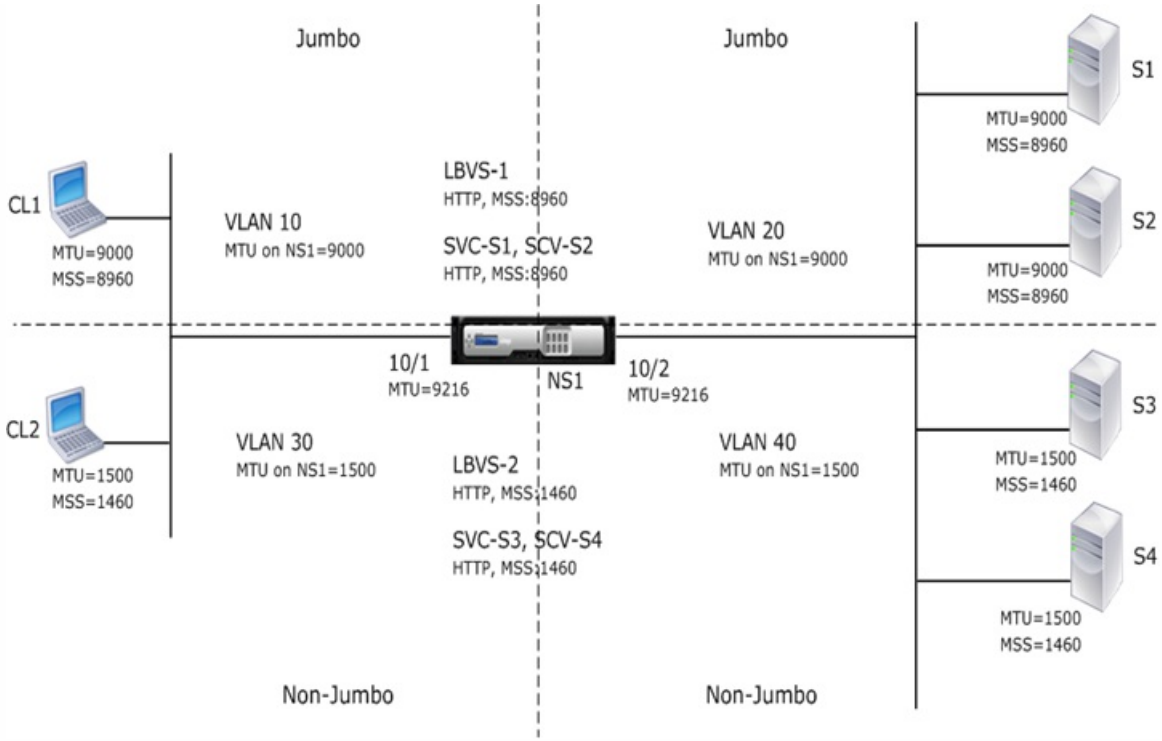
- The MTU of interface 10/1 is 9216. The MTU of VLAN 10 is 9000. On interface 10/1, the MTU of VLAN 10 tagged packets is 9000.
- The MTU of interface 10/2 is 9216. The MTU of VLAN 20 is 9000. On interface 10/2, the MTU of VLAN 20 tagged packets is 9000.
- The MTU of interface 10/1 is 9216. The MTU of VLAN 30 is 1500. On interface 10/1, the MTU of VLAN 30 tagged packets is 1500.
- The MTU of interface 10/2 is 9216. The MTU of VLAN 40 is 1500. On interface 10/2, the MTU of VLAN 40 tagged packets is 9000.

CL1, S1, S2, and all network devices between CL1 and S1 or S2 are configured for jumbo frames.

Since HTTP traffic is based on TCP, MSSs are set accordingly at each end point for supporting jumbo frames.

- For the connection between CL1 and virtual server LBVS-1 of NS1, the MSS on NS1 is set in a TCP profile, which is then bound to LBVS-1.
- For the connection between a SNIP address of NS1 and S1, the MSS on NS1 is set in a TCP profile, which is then bound to the service (SVC-S1) representing S1 on NS1.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The following table lists the settings used in this example.

Entity	Name	Details
IP address of clients	CL1	192.0.2.10
	CL2	192.0.3.20
IP address of servers	S1	198.51.100.19
	S2	198.51.100.20
	S3	198.51.101.19
	S4	198.51.101.20
SNIP addresses on NS1		• 198.51.100.18 • 198.51.101.18
MTU specified for interfaces and VLANs on NS1	10/1	9216
	10/2	9216
	VLAN 10	9000

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	VLAN 20	9000
	VLAN 30	1500
	VLAN 40	1500
Default TCP profile	nstcp_default_profile	MSS:1460
Custom TCP profile	ALL-JUMBO	MSS: 8960
Services on NS1 representing servers	SVC-S1	• IP address: 198.51.100.19 • Protocol: HTTP • Port: 80 • TCP profile: ALL-JUMBO (MSS: 8960)
	SVC-S2	• IP address: 198.51.100.20 • Protocol: HTTP • Port: 80 • TCP profile: ALL-JUMBO (MSS: 8960)
	SVC-S3	• IP address: 198.51.101.19 • Protocol: HTTP • Port: 80 • TCP Profile: nstcp_default_profile (MSS:1460)
	SVC-S4	• IP address: 198.51.101.20 • Protocol: HTTP • Port: 80 • TCP Profile: nstcp_default_profile (MSS:1460)
Load balancing virtual servers on NS1	LBVS-1	• IP address = 203.0.113.15 • Protocol: HTTP • Port:80 • Bound services: SVC-S1, SVC-S2 • TCP profile: ALL-JUMBO (MSS: 8960)
	LBVS-2	• IP address = 203.0.114.15 • Protocol: HTTP • Port:80 • Bound services: SVC-S3, SVC-S4

© 1999-2017 Citrix Systems, Inc. All rights reserved.

		• TCP Profile: nstcp_default_profile (MSS:1460)

Following is the traffic flow of CL1's request to S1:

1. Client CL1 creates a 20000-byte HTTP request to send to virtual server LBVS-1 of NS1.
2. CL1 opens a connection to LBVS-1 of NS1. CL1 and NS1 exchange their TCP MSS values while establishing the connection.
3. Because NS1's MSS value is smaller than the HTTP request, CL1 segments the request data into multiples of NS1's MSS and sends these segments in IP packets tagged as VLAN 10 to NS1.
   - Size of the first two packets = [IP Header + TCP Header + (TCP segment=NS1 MSS)] = [20 + 20 + 8960] = 9000
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 2080] = 2120
4. NS1 receives these packets at interface 10/1. NS1 accepts these packets because the size of these packets is equal to or less than the effective MTU (9000) of interface 10/1 for VLAN 10 tagged packets.
5. From the IP packets, NS1 assembles all the TCP segments to form the 20000-byte HTTP request. NS1 processes this request.
6. LBVS-1's load balancing algorithm selects server S1, and NS1 opens a connection between one of its SNIP addresses and S1. NS1 and CL1 exchange their respective TCP MSS values while establishing the connection.
7. NS1 segments the request data into multiples of S1's MSS and sends these segments in IP packets tagged as VLAN 20 to S1.
   - Size of the first two packets = [IP Header + TCP Header + (TCP payload=S1 MSS)] = [20 + 20 + 8960] = 9000
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 2080] = 2120

Following is the traffic flow of S1's response to CL1:

1. Server S1 creates a 30000-byte HTTP response to send to the SNIP address of NS1.
2. S1 segments the response data into multiples of NS1's MSS and sends these segments in IP packets tagged as VLAN 20 to NS1. These IP packets are sourced from S1's IP address and destined to the SNIP address of NS1.
   - Size of first three packet = [IP Header + TCP Header + (TCP segment=NS1's MSS size)] = [20 + 20 + 8960] = 9000
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 3120] = 3160
3. NS1 receives the response packets at interface 10/2. NS1 accepts these packets, because their size is equal to or less than the effective MTU value (9000) of interface 10/2 for VLAN 20 tagged packets.
4. From these IP packets, NS1 assembles all the TCP segments to form the 30000-byte HTTP response. NS1 processes this response.
5. NS1 segments the response data into multiples of CL1's MSS and sends these segments in IP packets tagged as VLAN 10, from interface 10/1, to CL1. These IP packets are sourced from LBVS's IP address and destined to CL1's IP address.
   - Size of first three packet = [IP Header + TCP Header + [(TCP payload=CL1's MSS size)] = [20 + 20 + 8960] = 9000
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 3120] = 3160

Configuration Tasks

Updated: 2014-01-16

Following table lists tasks, commands, and examples for creating the required configuration on the NetScaler appliance.

Task	Syntax	Examples
Set the MTU of the desired interfaces for supporting jumbo frames	set interface <id> -mtu <positive_integer>	set int 10/1 –mtu 9216 set int 10/2 –mtu 9216

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Task	Syntax	Examples
Create VLANs and set the MTU of the desired VLANs for supporting jumbo frames	add vlan <id> -mtu <positive_integer> show vlan <id>	add vlan 10 –mtu 9000 add vlan 20 –mtu 9000
		add vlan 30 –mtu 1500 add vlan 40 –mtu 1500
Bind interfaces to VLANs	bind vlan <id> -ifnum <interface_name> show vlan <id>	bind vlan 10 –ifnum 10/1 –tagged bind vlan 20 –ifnum 10/2 -tagged
		bind vlan 30 –ifnum 10/1 –tagged bind vlan 40 –ifnum 10/2 -tagged
Add a SNIP address	add ns ip <IPAddress> <netmask> -type SNIP show ns ip	add ns ip 198.51.100.18 255.255.255.0 –type SNIP
		add ns ip 198.51.101.18 255.255.255.0 –type SNIP
Create services representing HTTP servers	add service <serviceName> <ip> HTTP <port> show service <name>	add service SVC-S1 198.51.100.19 http 80 add service SVC-S2 198.51.100.20 http 80
		add service SVC-S3 198.51.101.19 http 80 add service SVC-S4 198.51.101.20 http 80
Create HTTP load balancing virtual servers and bind the services to it	add lb vserver <name> HTTP <ip> <port> bind lb vserver <vserverName> <serviceName>	add lb vserver LBVS-1 http 203.0.113.15 80 bind lb vserver LBVS-1 SVC-S1 bind lb vserver LBVS-1 SVC-S2

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Task	Syntax	Examples
	show lb vserver <name>	add lb vserver LBVS-2 http 203.0.114.15 80 bind lb vserver LBVS-2 SVC-S3 bind lb vserver LBVS-2 SVC-S4
Create a custom TCP profile and set its MSS for supporting jumbo frames	add tcpProfile <name> -mss <positive_integer> show tcpProfile <name>	add tcpprofile ALL-JUMBO –mss 8960
Bind the custom TCP profile to the desired load balancing virtual server and services	set service <Name> -tcpProfileName <string> show service <name>	set lb vserver LBVS-1 –tcpProfileName ALL-JUMBO set service SVC-S1 –tcpProfileName ALL-JUMBO set service SVC-S2 –tcpProfileName ALL-JUMBO
Save the configuration	save ns config show ns config	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Access Control Lists

May 10, 2012

Access Control Lists (ACLs) filter IP traffic and secure your network from unauthorized access. An ACL is a set of conditions that the NetScaler ADC evaluates to determine whether to allow access. For example, the Finance department probably does not want to allow its resources to be accessed by other departments, such as HR and Documentation, and those departments want to restrict access to their data.

When the NetScaler ADC receives a data packet, it compares the information in the data packet with the conditions specified in the ACL and allows or denies access. The administrator of the organization can configure ACLs to function in the following processing modes:

- ALLOW—Process the packet.
- BRIDGE—Bridge the packet to the destination without processing it. The packet is directly sent by Layer 2 and Layer 3 forwarding.
- DENY—Drop the packet.

ACL rules are the first level of defense on the NetScaler ADC.

NetScaler supports the following types of ACLs:

- **Simple ACLs** filter packets on the basis of their source IP address and, optionally, their protocol, destination port, or traffic domain. Any packet that has the characteristics specified in the ACL is dropped.
- **Extended ACLs** filter data packets on the basis of various parameters, such as source IP address, source port, action, and protocol. An extended ACL defines the conditions that a packet must satisfy for the NetScaler ADC to process the packet, bridge the packet, or drop the packet.

Nomenclature

In the NetScaler user interfaces, the terms simple ACL and extended ACL refer to ACLs that process IPv4 packets. An ACL that processes IPv6 packets is called a simple ACL6 and or extended ACL6. When discussing both types, this documentation sometimes refers to both of them as simple ACLs or extended ACLs.

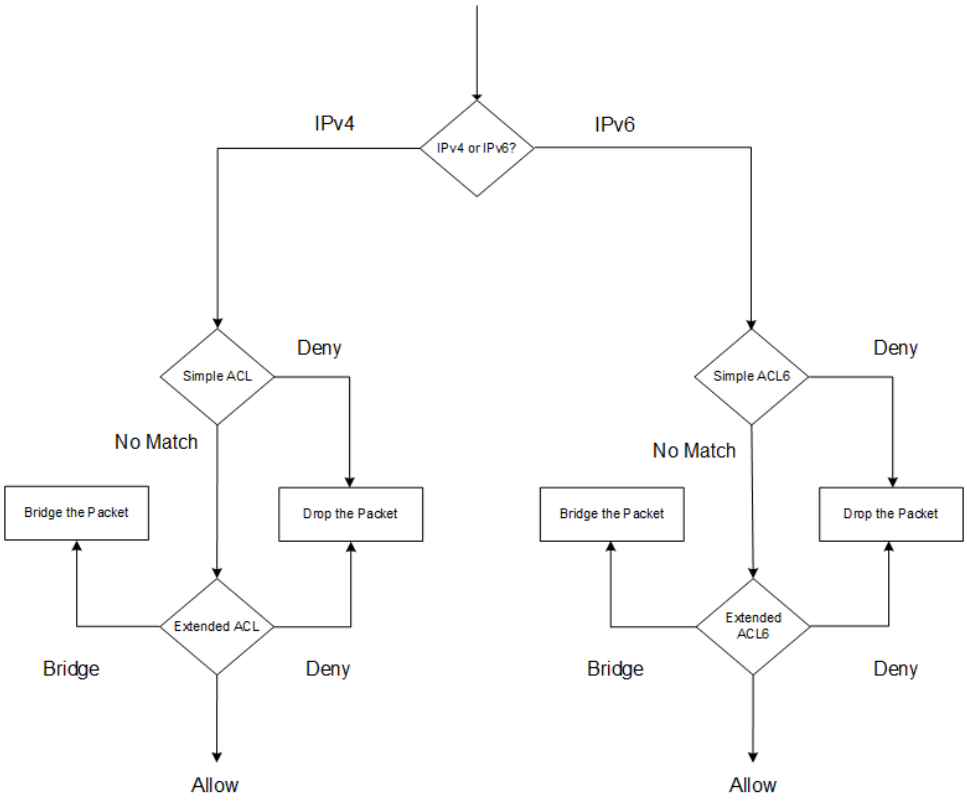
ACL Precedence

If both simple and extended ACLs are configured, incoming packets are compared to the simple ACLs first.

The NetScaler ADC first determines whether the incoming packet is an IPv4 or an IPv6 packet, and then compares the packet's characteristics to either simple ACLs or simple ACL6s. If a match is found, the packet is dropped. If no match is found, the packet is compared to extended ACLs or extended ACL6s. If that comparison results in a match, the packet is handled as specified in the ACL. The packet can be bridged, dropped, or allowed. If no match is found, the packet is allowed.

Figure 1. Simple and Extended ACLs Flow Sequence

© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Simple ACLs and Simple ACL6s

May 21, 2015

A simple ACL or simple ACL6 uses few parameters and can be configured only to drop IP packets. Packets can be dropped on the basis of their source IP address and, optionally, their protocol, destination port, or traffic domain.

When creating a simple ACL or simple ACL6, you can specify a time to live (TTL), in seconds, after which the ACL expires. ACLs with TTLs are not saved when you save the configuration. You can display simple ACLs and simple ACL6s to verify their configuration, and you can display their statistics.

This section includes the following details:

- Configuring Simple ACLs and Simple ACL6s
- Displaying Simple ACL and Simple ACL6 Statistics
- Terminating Established Connections

Configuring Simple ACLs and Simple ACL6s

Configuring a simple ACL or simple ACL6 on a NetScaler ADC can include the following tasks.

- Create simple ACLs or simple ACL6s to drop (deny) packets on the basis of their source IP address and, optionally, their protocol, destination port, or traffic domain.
- Remove simple ACLs or simple ACL6s. These ACLs cannot be modified once created. If you need to modify a simple ACL or simple ACL6, you must remove it and create a new one.

To create a simple ACL by using the command line interface

At the command prompt, type the following commands to add an ACL and verify the configuration:

- **add ns simpleacl** <aclname> DENY -**srcIP** <ip_addr> [-**destPort**<port> -**protocol** (TCP | UDP)] [-**TTL** <positive_integer>]
- **show ns simpleacl** [<aclname>]

Example

```
> add simpleacl rule1 DENY -srcIP 10.102.29.5 -TTL 600
Done
```

To create a simple ACL6 by using the command line interface

At the command prompt, type the following commands to add a simple ACL6 and verify the configuration:

- **add ns simpleacl6** <aclname> DENY -**srcIPv6** <ipv6_addr|null> [-**destPort**<port> -**protocol** (TCP | UDP)] [-**TTL** <positive_integer>]
- **show ns simpleacl6** [<aclname>]

Example

```
>  add ns simpleacl6 rule1 DENY –srcIPv6 3ffe:192:168:215::82 -destPort 80 -Protocol TCP -TTL 9000
 Done
```

To remove a single simple ACL by using the command line interface

At the command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **rm ns simpleacl** <aclname>
- **show ns simpleacl**

To remove a single simple ACL6 by using the command line interface

At the command prompt, type:

- **rm ns simpleacl6**<aclname>
- **show ns simpleacl6**

To remove all simple ACLs by using the command line interface

At the command prompt, type:

- **clear ns simpleacl**
- **show ns simpleacl**

To remove all simple ACL6s by using the command line interface

At the command prompt, type:

- **clear ns simpleacl6**
- **show ns simpleacl6**

To create a simple ACL by using the configuration utility

Navigate to System > Network > ACLs and, on the **Simple ACLs** tab, add a new simple ACL.

To create a simple ACL6 by using the configuration utility

Navigate to System > Network > ACLs and, on the **Simple ACL6s** tab, add a new simple ACL6.

To remove a single simple ACL by using the configuration utility

Navigate to System > Network > ACLs and, on the **Simple ACLs** tab, delete the simple ACL.

To remove a single simple ACL6 by using the configuration utility

Navigate to System > Network > ACLs and, on the **Simple ACL6s** tab, delete the simple ACL6.

To remove all simple ACLs by using the configuration utility

1. Navigate to System > Network > ACLs.
2. On the **Simple ACLs** tab, in the **Action** list, click **Clear**.

To remove all simple ACL6s by using the configuration utility

1. Navigate to System > Network > ACLs.
2. On the **Simple ACL6s** tab, in the **Action** list, click **Clear**.

Displaying Simple ACL and Simple ACL6 Statistics

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You can display the simple ACL (or simple ACL6) statistics, which include the number of hits, the number of misses, and the number of simple ACLs configured.

The following table describes statistics you can display for simple ACLs and simple ACL6s.

Statistic	Indicates
ACL hits	Packets matching an ACL
ACL misses	Packets not matching any ACL
ACL count	Number of ACLs configured

To display simple ACL statistics by using the command line interface

At the command prompt, type:

stat ns simpleacl

Example

```
> stat ns simpleacl

SimpleACL Statistics

                        Rate (/s)        Total
SimpleACL hits                0              0
SimpleACL misses                0          51872
SimpleACLs count               --              2
Done
```

To display simple ACL6 statistics by using the command line interface

At the command prompt, type:

stat ns simpleacl6

To display simple ACL statistics by using the configuration utility

Navigate to System > Network > ACLs and, on the **Simple ACLs** tab, select the ACL and click **Statistics**.

To display simple ACL6 statistics by using the configuration utility

Navigate to System > Network > ACLs and, on the **Simple ACL6s** tab, select the simple ACL6 and click **Statistics**.

Terminating Established Connections

For a simple ACL or simple ACL6, the NetScaler ADC blocks any new connections that match the conditions specified in the ACL. Packets related to existing connections that were established before the ACL was created are not blocked. To terminate previously established connections that match an existing ACL, you can run a flush operation from the command

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

line interface or the configuration utility.

Flush can be useful in the following cases:

- You receive a list of blacklisted IP addresses and want to completely block those IP addresses from accessing the NetScaler ADC. In this case, you create simple ACLs or simple ACL6s to block any new connections from these IP addresses, and then flush any existing connections associated with those addresses.
- You want to terminate a large number of connections from a particular network without taking the time to terminate them one by one.

When you run flush, the NetScaler ADC searches through all of its established connections and terminates those that match conditions specified in any of the simple ACLs configured on the ADC.

Note: If you plan to create more than one simple ACL and flush existing connections that match any of them, you can minimize the effect on performance by first creating all of the simple ACLs and then running flush only once.

To terminate all established IPv4 connections that match any of your configured simple ACLs by using the command line interface

At the command prompt, type:

flush simpleacl -estSessions

To terminate all established IPv6 connections that match any of your configured simple ACL6s by using the command line interface

At the command prompt, type:

flush simpleacl6 -estSessions

To terminate all established IPv4 connections that match any of your configured simple ACLs by using the configuration utility

1. Navigate to System > Network > ACLs.
2. On the **Simple ACLs** tab, in the **Action** list, click **Flush**.

To terminate all established IPv6 connections that match any of your configured simple ACL6s by using the configuration utility

1. Navigate to System > Network > ACLs.
2. On the **Simple ACL6s** tab, in the **Action** list, click **Flush**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Extended ACLs and Extended ACL6s

Jan 31, 2011

Extended ACLs and extended ACL6s provide parameters and actions not available with simple ACLs. You can filter data on the basis of parameters such as source IP address, source port, action, and protocol. You can specify tasks to allow a packet, deny a packet, or bridge a packet.

Extended ACLs and ACL6s can be modified after they are created, and you can renumber their priorities to specify the order in which they are evaluated.

Note: If you configure both simple and extended ACLs, simple ACLs take precedence over extended ACLs.
The following actions can be performed on extended ACLs and ACL6s: Modify, Apply, Disable, Enable, Remove, and Renumber (the priority). You can display extended ACLs and ACL6s to verify their configuration, and you can display their statistics.

You can configure the NetScaler ADC to log details for packets that match an extended ACL. However, you cannot log details of packets that match an ext

Applying Extended ACLs and Extended ACL6s

Unlike simple ACLs and ACL6s, extended ACLs and ACL6s created on the NetScaler ADC do not work until they are applied. Also, if you make any modifications to an extended ACL or ACL6, such as disabling the ACLs, changing a priority, or deleting the ACLs, you must reapply the extended ACLs or ACL6s. You must also reapply them after enabling logging. The procedure to apply extended ACLs or ACL6s reapplies all of them. For example, if you have applied extended ACL rules 1 through 10, and you then create and apply rule 11, the first 10 rules are applied afresh.

If a session has a DENY ACL related to it, that session is terminated when you apply the ACLs.

Extended ACLs and ACL6s are enabled by default. When they are applied, the NetScaler ADC starts comparing incoming packets against them. However, if you disable them, they are not used until you re-enable them, even if they are reapplied.

Renumbering the priorities of Extended ACLs and Extended ACL6s

Priority numbers determine the order in which extended ACLs or ACL6s are matched against a packet. An ACL with a lower priority number has a higher priority. It is evaluated before ACLs with higher priority numbers (lower priorities), and the first ACL to match the packet determines the action applied to the packet.

When you create an extended ACL or ACL6, the NetScaler ADC automatically assigns it a priority number that is a multiple of 10, unless you specify otherwise. For example, if two extended ACLs have priorities of 20 and 30, respectively, and you want a third ACL to have a value between those numbers, you might assign it a value of 25. If you later want to retain the order in which the ACLs are evaluated but restore their numbering to multiples of 10, you can use the renumber procedure.

This section includes the following details:

- Configuring Extended ACLs and Extended ACL6s
- Logging Extended ACLs (IPv4 Only)
- Displaying Extended ACL and Extended ACL6s Statistics
- Sample Configurations

Configuring Extended ACLs and Extended ACL6s

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring an extended ACL or ACL6 on a NetScaler ADC consists of the following tasks.

- Create an extended ACL or ACL6 to either allow, deny, or bridge a packet. You can specify an IP address or range of IP addresses to match against the source or destination IP addresses of the packets. You can specify a protocol to match against the protocol of incoming packets.
- (Optional) You can modify extended ACLs or ACL6s that you previously created. Or, if you want to temporarily take one out of use you can disable it, and later reenable it.
- Apply extended ACLs or ACL6s. After you create, modify, disable or reenable, or delete an extended ACL or ACL6, you must apply the extended ACLs or ACL6s to activate them.
- (Optional) Renumber the priorities of extended ACLs or ACL6s. If you have configured ACLs with priorities that are not multiples of 10 and want to restore the numbering to multiples of 10, use the renumber procedure.

To create an extended ACL by using the command line interface

At the command prompt, type:

- **add ns acl** <aclname> <aclaction> [-**srcIP** [<operator>] <srcIPVal>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIP** [<operator>] <destIPVal>] [-**destPort** [<operator>] <destPortVal>] [-**TTL** <positive_integer>] [-**srcMac** <mac_addr>] [(-**protocol** <protocol> [-established]) | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer>] [-**interface** <interface_name>] [-**icmpType** <positive_integer> [-**icmpCode** <positive_integer>]] [-**priority** <positive_integer>] [-**state** (ENABLED | DISABLED)] [-**logstate** (ENABLED | DISABLED) [-**ratelimit** <positive_integer>]]
- **show ns acl** [<aclName>]

Example

```
> add ns acl restrict DENY -srcport 45-1024 -destIP 192.168.1.1 -protocol TCP
 Done
```

To create an extended ACL6 by using the command line interface

At the command prompt, type:

- **add ns acl6** <acl6name> <acl6action> [-**srcIPv6** [<operator>] <srcIPv6Val>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIPv6** [<operator>] <destIPv6Val>] [-**destPort** [<operator>] <destPortVal>] [-**TTL** <positive_integer>] [-**srcMac** <mac_addr>] [(-**protocol** <protocol> [-established]) | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer>] [-**interface** <interface_name>] [-**icmpType** <positive_integer> [-icmpCode <positive_integer>]] [-**priority** <positive_integer>] [-**state** (ENABLED | DISABLED)]
- **show ns acl6** [<aclName>]

Example

```
> add ns acl6 rule6  DENY -srcport 45-1024 -destIPv6 2001::45 -protocol TCP
 Done
```

To modify an extended ACL by using the command line interface

To modify an extended ACL, type the **set ns acl** command, the name of the extended ACL, and the parameters to be changed, with their new values.

To modify an extended ACL6 by using the command line interface

To modify an extended ACL6, type the **set ns acl** command, the name of the extended ACL6, and the parameters to be changed, with their new values.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To disable or enable an extended ACL by using the command line interface

At the command prompt, type one of the following commands:

- **disable ns acl** <aclname>
- **enable ns acl** <aclname>

To disable or enable an extended ACL6 by using the command line interface

At the command prompt, type one of the following commands:

- **disable ns acl6** <aclname>
- **enable ns acl6**<aclname>

To apply extended ACLs by using the command line interface

At the command prompt, type:

apply ns acls

To apply extended ACL6s by using the command line interface

At the command prompt, type:

apply ns acls6

To renumber the priorities of extended ACLs by using the command line interface

At the command prompt, type:

renumber ns acls

To renumber the priorities of extended ACL6s by using the command line interface

At the command prompt, type:

renumber ns acls6

To configure an extended ACL by using the configuration utility

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab, add a new extended ACL or edit an existing extended ACL. To enable or disable an existing extended ACL, select it, and then select **Enable** or **Disable** from the **Action** list.

To configure an extended ACL6sACL6s by using the configuration utility

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab, add a new extended ACL6 or edit an existing extended ACL6. To enable or disable an existing extended ACL6, select it, and then select **Enable** or **Disable** from the **Action** list.

To apply extended ACLs by using the configuration utility

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab, in the **Action** list, click **Apply**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To apply extended ACL6sACL6s by using the configuration utility

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab, in the **Action** list, click **Apply**.

To renumber the priorities of extended ACLs by using the configuration utility

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab, in the **Action** list, click **Renumber Priority (s)**.

To renumber the priorities of extended ACL6sACL6s by using the configuration utility

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab, in the **Action** list, click **Renumber Priority (s)**.

Logging Extended ACLs (IPv4 Only)

You can configure the NetScaler ADC to log details for packets that match extended ACLs.

Note: You cannot enable logging for extended ACL6s.
In addition to the ACL name, the logged details include packet-specific information such as the source and destination IP addresses. The information is stored either in the syslog file or in the nslog file, depending on the type of global logging (syslog or nslog) enabled.

Logging must be enabled at both the global level and the ACL level. The global setting takes precedence. For more information about enabling logging globally, see "."

To optimize logging, when multiple packets from the same flow match an ACL, only the first packet's details are logged, and the counter is incremented for every packet that belongs to the same flow. A flow is defined as a set of packets that have the same values for the source IP address, destination IP address, source port, destination port, and protocol parameters. To avoid flooding of log messages, the NetScaler ADC performs internal rate limiting so that packets belonging to the same flow are not repeatedly logged. The total number of different flows that can be logged at any given time is limited to 10,000.

Note: You must apply ACLs after you enable logging.

To configure extended ACL Logging by using the command line interface

At the command prompt, type the following commands to configure logging and verify the configuration:

- **set ns acl** <aclName> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** <positive_integer>]
- **show ns acl** [<aclName>]

Example

```
> set ns acl restrict -logstate ENABLED -ratelimit 120
Warning: ACL modified, apply ACLs to activate change
```

To configure extended ACL Logging by using the configuration utility

1. Navigate to System > Network > ACLs and, on the **Extended ACLs** tab, open the extended ACL.
2. Set the following parameters:
   - **Log State**— Enable or disable logging of events related to the extended ACL rule. The log messages are stored in the configured syslog or auditlog server.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Log Rate Limit**—Maximum number of log messages to be generated per second. If you set this parameter, you must enable the Log State parameter.

Displaying Extended ACL and Extended ACL6s Statistics

You can display statistics of extended ACLs and ACL6s.

The following table lists the statistics associated with extended ACLs and ACL6s, and their descriptions.

Statistic	Specifies
Allow ACL hits	Packets matching ACLs with processing mode set to ALLOW. The NetScaler ADC processes these packets.
NAT ACL hits	Packets matching a NAT ACL, resulting in a NAT session.
Deny ACL hits	Packets dropped because they match ACLs with processing mode set to DENY.
Bridge ACL hits	Packets matching a bridge ACL, which in transparent mode bypasses service processing.
ACL hits	Packets matching an ACL.
ACL misses	Packets not matching any ACL.

To display the statistics of all extended ACLs by using the command line interface

At the command prompt, type:

stat ns acl

To display the statistics of all extended ACL6s by using the command line interface

At the command prompt, type:

stat ns acl6

To display the statistics of an extended ACL by using the configuration utility

Navigate to System > Network > ACLs, on the **Extended ACLs** tab, select the extended ACL, and click **Statistics**.

To display the statistics of an extended ACL6 by using the configuration utility

Navigate to System > Network > ACLs, on the **Extended ACL6s** tab, select the extended ACL, and click **Statistics**.

Sample Configurations

The following table shows examples of configuring extended ACL rules through the command line interface.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Action - ALLOW	
Tasks	**Steps**
Create an extended ACL rule to allow a particular host to access the servers.	>add ns acl allow-client ALLOW -srcIP = 40.40.40.1 Done
Create an extended ACL rule to allow a particular network to access the servers.	>add ns acl allow-client-net ALLOW -srcIP = 40.40.40.0-40.40.40.255 Done
Create extended ACL rules to allow HTTP, TFTP, and ICMP traffic.	>add acl allow-http ALLOW -protocol tcP -destport 80 Done Done >add acl allow-tftp ALLOW -protocol udp -destport 69 Done >add acl allow-icmp ALLOW -protocol icmp Done
Create an extended ACL rule to allow access to a particular destination/network.	>add acl allow-dest-access ALLOW -destip 20.20.20.0-20.20.20.255 Done
Create an extended ACL rule to allow traffic coming from a particular VLAN.	>add acl allow-vlan ALLOW -vlan 3000 Done
Action - DENY	
Tasks	**Steps**
Create an extended ACL rule to deny access to the servers by a particular host.	>add ns acl deny-client DENY -srcIP = 50.50.50.1 Done
Create an extended ACL rule to deny access to the servers from a particular network.	> add ns acl deny-client-net DENY -srcIP = 50.50.50.0-50.50.50.255 Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Create extended ACL rules to deny Telnet and FTP traffic.	>add ns acl deny-client-Telnet DENY -protocol TCP -destPort 23 Done > add ns acl deny-client-FTP DENY -protocol TCP -destPort 20-21 Done
Create an extended ACL rule to deny TCP traffic to port 80 from a particular host/network.	>add ns acl deny-client-TCP DENY -protocol TCP -destPort 80 -destIP 20.20.20.0-20.20.20.255 Done
Create an extended ACL rule to deny traffic from a particular VLAN.	> add acl deny-vlan DENY -vlan 2000 Done
Action - BRIDGE	
Tasks	**Steps**
Create an extended ACL rule to bridge FTP traffic.	>add ns acl bridge-ftp BRIDGE -protocol TCP -destport 21 Done >add ns acl bridge-ftp-data BRIDGE -protocol TCP -destport 21 Done
Create an extended ACL rule to bridge all traffic from a particular VLAN.	>add ns acl bridge-client-vlan BRIDGE -vlan 1000 Done
MAC Address Filtering	
Tasks	**Steps**
Create an extended ACL rule to allow traffic from a particular MAC address to a particular host.	>add ns acl allow-mac-host ALLOW -srcMAC 2a:c1:69:92:a0:7b -destIP 10.10.10.1 Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Create an extended ACL rule to allow traffic from hosts with a specific MAC UUID.	> add ns acl allow-mac-uuid ALLOW -srcMAC 2a:c1:69:92:a0:7b -srcMacMask 000000111111 Done
ACL with RNAT (Typically, RNAT is used to allow servers configured with private non-routable IP addresses to initiate connections to the Internet.)	
Tasks	**Steps**
Create an RNAT rule for a particular host.	>add ns acl rnat-acl-host ALLOW -srcIP 40.40.40.1 Done >apply ns acls Done >set rnat rnat-acl Done
Create an RNAT rule for a particular network.	>add ns acl rnat-acl-network ALLOW -srcIP 40.40.40.0-40.40.40.255 Done >set rnat rnat-acl-network -NATIP 5.5.5.1 Done
ACL with Forwarding Session	
Create a forwarding session rule for a case in which a client request forwarded to a server results in a response that has to return by the same path.	>add ns acl forward-acl-host ALLOW -srcIP 20.20.20.1 Done >add forwardingSession fs -aclname forward-acl-host Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

MAC Address Wildcard Mask for ACLs

Jan 31, 2011

A wildcard mask parameter has been introduced for extended ACLs and ACL6s and is used with the source MAC address parameter to define a range of MAC addresses to be match against the source MAC address of incoming packets.

Wild card masks specify which hexadecimal digits of the MAC address are used and which hexadecimal digits are ignored. The wildcard mask parameter specifies a series of ones and zeroes and has a length of 12 digits. Each digit is a mask for the corresponding hexadecimal digit of the MAC address. A zero digit in the wildcard mask indicates that the corresponding hexadecimal digit of the MAC address must be considered and a one digit indicates that the corresponding hexadecimal digit to be ignored.

The wildcard mask should meet the following conditions:

- Has only one series of zeroes
- Has only one series of ones
- Start with a series of zeroes

The following are some of the examples of valid wildcard masks:

- 000000111111
- 000000011111
- 000011111111

The following are some of the examples of invalid wildcard masks:

- 000000111100
- 111110000000
- 010101010101

For an ACL, a wildcard mask of 000000111111 for MAC address 96:fa:95:1d:67:4a defines the MAC address range 96:FA:95:00:00:00 - 96:FA:95:FF:FF:FF. This MAC address range is matched against the source MAC address of the incoming packets.

To specify a range of source MAC addresses in an ACL rule by using the NetScaler command line

At the command prompt, type:

- add ns acl <name> <action> -srcMac <mac_addr> -srcMacMask <string>
- show ns acl <aclname>

Example

```
> add ns acl ACL-1 ALLOW –protocol TCP –srcport 2000-3000 -srcMac 96:fa:95:1d:67:4a
 –srcMacMask 000000111111
Done
```

To specify a range of source MAC addresses in an ACL6 rule by using the NetScaler command line

At the command prompt, type:

- add ns acl6 <name> <action> -srcMac <mac_addr> -srcMacMask <string>
- show ns acl6 <acl6name>

Example

```
> add ns acl6 ACL6-1 ALLOW -destIPv6 2001::45 -srcMac 96:fa:90:1d:67:4a
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

–srcMacMask 000000001111
Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Blocking Traffic on Internal Ports

Jan 31, 2011

The NetScaler appliance does not block traffic that matches an ACL rule if the traffic is destined to the appliance's NSIP address, or one of its SNIP addresses, and a port in the 3008-3011 range.

This behavior is now specified by the default setting of the new Implicit ACL Allow (implicitACLAllow) parameter (of the L3 param command). You can disable this parameter if you want to block traffic to ports in the 3008-3011 range. An appliance in a high availability configuration makes an exception for its partner (primary or secondary) node. It does not block traffic from that node.

To disable or enable this parameter by using the command line interface

At the command prompt, type:

set l3param -implicitACLAllow [ENABLED | DISABLED]

Note: The parameter implicitACLAllow is enabled by default.

Example

```
> set l3param -implicitACLAllow DISABLED

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

IP Routing

Sep 06, 2013

NetScaler appliances support both dynamic and static routing. Because simple routing is not the primary role of a NetScaler, the main objective of running dynamic routing protocols is to enable route health injection (RHI), so that an upstream router can choose the best among multiple routes to a topographically distributed virtual server.

Most NetScaler implementations use some static routes to reduce routing overhead. You can create backup static routes and monitor routes to enable automatic switchover in the event that a static route goes down. You can also assign weights to facilitate load balancing among static routes, create null routes to prevent routing loops, and configure IPv6 static routes. You can configure policy based routes (PBRs), for which routing decisions are based on criteria that you specify.

This document includes the following information:

- Configuring Dynamic Routes
- Configuring Static Routes
- Route Health Injection Based on Virtual Server Settings
- Configuring Policy-Based Routes
- Troubleshooting Routing Issues

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Dynamic Routes

Apr 26, 2017

When a dynamic routing protocol is enabled, the corresponding routing process monitors route updates and advertises routes. Routing protocols enable an upstream router to use the equal cost multipath (ECMP) technique to load balance traffic to identical virtual servers hosted on two standalone NetScaler appliances. Dynamic routing on a NetScaler appliance uses three routing tables. In a high-availability setup, the routing tables on the secondary appliance mirror those on the primary.

For command reference guides and unsupported commands on dynamic routing protocol, see Dynamic Routing Protocol Command Reference Guides and Unsupported Commands.

The NetScaler supports the following protocols:

- Routing Information Protocol (RIP) version 2
- Open Shortest Path First (OSPF) version 2
- Border Gateway Protocol (BGP)
- Routing Information Protocol next generation (RIPng) for IPv6
- Open Shortest Path First (OSPF) version 3 for IPv6
- ISIS Protocol

You can enable more than one protocol simultaneously.

Routing Tables in the NetScaler

In a NetScaler appliance, the NetScaler kernel routing table, the FreeBSD kernel routing table, and the NSM FIB routing table each hold a different set of routes and serve a different purpose. They communicate with each other by using UNIX routing sockets. Route updates are not automatically propagated from one routing table to another. You must configure propagation of route updates for each routing table.

NS Kernel Routing Table

The NS kernel routing table holds subnet routes corresponding to the NSIP and to each SNIP and MIP. Usually, no routes corresponding to VIPs are present in the NS kernel routing table. The exception is a VIP added by using the add ns ip command and configured with a subnet mask other than 255.255.255.255. If there are multiple IP addresses belonging to the same subnet, they are abstracted as a single subnet route. In addition, this table holds a route to the loopback network (127.0.0.0) and any static routes added through the command line interface (CLI). The entries in this table are used by the NetScaler in packet forwarding. From the NetScaler CLI, they can be inspected with the show route command.

FreeBSD Routing Table

The sole purpose of the FreeBSD routing table is to facilitate initiation and termination of management traffic (telnet, ssh, etc.). In a NetScaler appliance, these applications are tightly coupled to FreeBSD, and it is imperative for FreeBSD to have the necessary information to handle traffic to and from these applications. This routing table contains a route to the NSIP subnet and a default route. In addition, FreeBSD adds routes of type WasCloned(W) when the NetScaler establishes connections to hosts on local networks. Because of the highly specialized utility of the entries in this routing table, all other route updates from the NS kernel and NSM FIB routing tables bypass the FreeBSD routing table. Do not modify it with the route command. The FreeBSD routing table can be inspected by using the netstat command from any UNIX shell.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Network Services Module (NSM) FIB

The NSM FIB routing table contains the advertisable routes that are distributed by the dynamic routing protocols to their peers in the network. It may contain:

Connected routes

IP subnets that are directly reachable from the NetScaler. Typically, routes corresponding to the NSIP subnet and subnets over which routing protocols are enabled are present in NSM FIB as connected routes.

Kernel routes

All the VIP addresses on which the -hostRoute option is enabled are present in NSM FIB as kernel routes if they satisfy the required RHI Levels. In addition, NSM FIB contains any static routes configured on the NetScaler CLI that have the -advertise option enabled. Alternatively, if the NetScaler is operating in Static Route Advertisement (SRADV) mode, all static routes configured on the NetScaler CLI are present in NSM FIB. These static routes are marked as kernel routes in NSM FIB, because they actually belong to the NS kernel routing table.

Static routes

Normally, any static route configured in VTYSH is present in NSM FIB. If administrative distances of protocols are modified, this may not always be the case. An important point to note is that these routes can never get into the NS kernel routing table.

Learned routes

If the NetScaler is configured to learn routes dynamically, the NSM FIB contains routes learned by the various dynamic routing protocols. Routes learned by OSPF, however, need special processing. They are downloaded to FIB only if the fib-install option is enabled for the OSPF process. This can be done from the router-config view in VTYSH.

High Availability Setup

In a high availability setup, the primary node runs the routing process and propagates routing table updates to the secondary node. The routing table of the secondary node mirrors the routing table on the primary node.

Non-Stop Forwarding

After failover, the secondary node takes some time to start the protocol, learn the routes, and update its routing table. But this does not affect routing, because the routing table on the secondary node is identical to the routing table on the primary node. This mode of operation is known as non-stop forwarding.

Black Hole Avoidance Mechanism

After failover, the new primary node injects all its VIP routes into the upstream router. However, that router retains the old primary node's routes for 180 seconds. Because the router is not aware of the failover, it attempts to load balance traffic between the two nodes. During the 180 seconds before the old routes expire, the router sends half the traffic to the old, inactive primary node, which is, in effect, a black hole.

To prevent this, the new primary node, when injecting a route, assigns it a metric that is slightly lower than the one specified by the old primary node.

Interfaces for Configuring Dynamic Routing

To configure dynamic routing, you can use either the configuration utility or a command-line interface. The NetScaler supports two independent command-line interfaces: the NetScaler CLI and the Virtual Teletype Shell (VTYSH). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler CLI is the appliance's native shell. VTYSH is exposed by ZebOS. The NetScaler routing suite is based on ZebOS, the commercial version of GNU Zebra.

Note: Citrix recommends that you use VTYSH for all commands except those that can be configured only on the NetScaler CLI. Use of the NetScaler CLI should generally be limited to commands for enabling the routing protocols, configuring host route advertisement, and adding static routes for packet forwarding.

Dynamic Routing Protocol Command Reference Guides and Unsupported Commands

The following table lists command reference guide links, for various dynamic routing protocols, and unsupported commands on the NetScaler appliance:

Dynamic Routing Protocol	Command Reference Guide	Unsupported Commands
OSPF	OSPF Command Reference	• Domain-id command • OSPF-TE related commands • OSPF-VPN related commands • CSPF-TE related commands • ip ospf resync-timeout command • capability opaque command • enable ext-ospf-multi-inst command • Graceful restart related commands
IPv6 OSPF (OSPFv3)	OSPF Command Reference	• OSPF-TE related commands • Graceful restart related commands
BGP	BGP Command Reference	• VPN/VRF related commands • MPLS related commands • 6PE commands (IPv6 provider edge) • MD5 authentication related commands • Multicast options • set-overload-bit command • Graceful restart related commands
IS-IS	IS-IS Command Reference	• capability cspf command • enable-cspf command • mpls traffic-eng command • mpls traffic-eng router-id command • multi-topology for ipv6 address family related commands • Graceful restart related commands
RIP and IPv6 RIP (RIPng)	-	• neighbor command

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring RIP

Mar 20, 2012

Routing Information Protocol (RIP) is a Distance Vector protocol. The NetScaler supports RIP as defined in RFC 1058 and RFC 2453. RIP can run on any subnet.

After enabling RIP, you need to configure advertisement of RIP routes. For troubleshooting, you can limit RIP propagation. You can display RIP settings to verify the configuration.

Enabling and Disabling RIP

Updated: 2013-08-30

Use either of the following procedures to enable or disable RIP. After you enable RIP, the NetScaler appliance starts the RIP process. After you disable RIP, the appliance stops the RIP process.

To enable or disable RIP routing by using the command line interface

At the command prompt, enter one of the following commands to enable or disable RIP:

- enable ns feature RIP
- disable ns feature RIP

To enable or disable RIP routing by using the configuration utility

1. Navigate to System > Settings, in Modes and Features group, click Change advanced features.
2. Select or clear the RIP Routing option.

Advertising Routes

Updated: 2013-08-30

RIP enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone NetScaler appliances. Route advertisement enables an upstream router to track network entities located behind the NetScaler.

To configure RIP to advertise routes by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router rip	Start the RIP routing process and enter configuration mode for the routing process.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command	Specifies
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
```

Limiting RIP Propagations

Updated: 2013-08-30

If you need to troubleshoot your configuration, you can configure listen-only mode on any given interface.

To limit RIP propagation by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router rip	Start the RIP routing process and enter configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# passive-interface VLAN0
```

Verifying the RIP Configuration

Updated: 2013-08-30

You can display the routing table and other RIP settings.

To view the RIP settings by using the VTYSH command line

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands in the following order:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh rip	Display updated RIP routing table.
sh rip interface <vlan_name>	Displays RIP information for the specified VLAN.

Example

```
NS# VTYSH
NS# sh rip
NS# sh rip interface VLAN0
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring OSPF

Feb 07, 2014

The NetScaler supports Open Shortest Path First (OSPF) Version 2 (RFC 2328). The features of OSPF on the NetScaler are:

- If a vserver is active, the host routes to the vserver can be injected into the routing protocols.
- OSPF can run on any subnet.
- Route learning advertised by neighboring OSPF routers can be disabled on the NetScaler.
- The NetScaler can advertise Type-1 or Type-2 external metrics for all routes.
- The NetScaler can advertise user-specified metric settings for VIP routes. For example, you can configure a metric per VIP without special route maps.
- You can specify the OSPF area ID for the NetScaler.
- The NetScaler supports not-so-stubby-areas (NSSAs). An NSSA is similar to an OSPF stub area but allows injection of external routes in a limited fashion into the stub area. To support NSSAs, a new option bit (the N bit) and a new type (Type 7) of Link State Advertisement (LSA) area have been defined. Type 7 LSAs support external route information within an NSSA. An NSSA area border router (ABR) translates a type 7 LSA into a type 5 LSA that is propagated into the OSPF domain. The OSPF specification defines only the following general classes of area configuration:
  - Type 5 LSA: Originated by routers internal to the area are flooded into the domain by AS boarder routers (ASBRs).
  - Stub: Allows no type 5 LSAs to be propagated into/throughout the area and instead depends on default routing to external destinations.

After enabling OSPF, you need to configure advertisement of OSPF routes. For troubleshooting, you can limit OSPF propagation. You can display OSPF settings to verify the configuration.

Enabling and Disabling OSPF

Updated: 2013-09-05

To enable or disable OSPF, you must use either the command line interface or the configuration utility. When OSPF is enabled, the NetScaler starts the OSPF process. When OSPF is disabled, the NetScaler stops the OSPF routing process.

To enable or disable OSPF routing by using the command line interface

At the command prompt, type one of the following commands:

1. enable ns feature OSPF
2. disable ns feature OSPF

To enable or disable OSPF routing by using the configuration utility

1. Navigate to System > Settings, in Modes and Features group, click Change advanced features.
2. Select or clear the OSPF Routing option.

Advertising OSPF Routes

Updated: 2013-08-30

OSPF enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone NetScaler appliances. Route advertising enables an upstream router to track network entities located behind the NetScaler.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure OSPF to advertise routes by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enters global configuration mode.
router OSPF	Start OSPF routing process and enter configuration mode for the routing process.
network A.B.C.D/M area <0-4294967295>	Enable routing on an IP network.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# network 10.102.29.0/24 area 0
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
```

Limiting OSPF Propagations

Updated: 2013-08-30

If you need to troubleshoot your configuration, you can configure listen-only mode on any given VLAN.

To limit OSPF propagation by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router OSPF	Start OSPF routing process and enters configuration mode for the routing process.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command	Specifies
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# passive-interface VLAN0
```

Verifying the OSPF Configuration

Updated: 2013-08-30

You can display current OSPF neighbors, and OSPF routes.

To view the OSPF settings by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh OSPF neighbor	Displays current neighbors.
sh OSPF route	Displays OSPF routes.

Example

```
>VTYSH
NS# sh ip OSPF neighbor
NS# sh ip OSPF route
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring BGP

Mar 20, 2012

The NetScaler appliance supports BGP (RFC 4271). The features of BGP on the NetScaler are:

- The NetScaler advertises routes to BGP peers.
- The NetScaler injects host routes to virtual IP addresses (VIPs), as determined by the health of the underlying virtual servers.
- The NetScaler generates configuration files for running BGP on the secondary node after failover in an HA configuration.
- This protocol supports IPv6 route exchanges.
- As-Override Support in Border Gateway Protocol

After enabling BGP, you need to configure advertisement of BGP routes. For troubleshooting, you can limit BGP propagation. You can display BGP settings to verify the configuration.

Prerequisites for IPv6 BGP

Before you begin configuring IPv6 BGP, do the following:

- Make sure that you understand the IPv6 BGP protocol.
- Install the IPv6PT license on the NetScaler appliance.
- After installing the IPv6PT license, enable the IPv6 feature.

Enabling and Disabling BGP

Updated: 2013-09-05

To enable or disable BGP, you must use either the command line interface or the configuration utility. When BGP is enabled, the NetScaler appliance starts the BGP process. When BGP is disabled, the appliance stops the BGP process.

To enable or disable BGP routing by using the command line interface

At the command prompt, type one of the following commands:

- enable ns feature BGP
- disable ns feature BGP

To enable or disable BGP routing by using the configuration utility

1. Navigate to System > Settings, in Modes and Features group, click Change advanced features.
2. Select or clear the BGP Routing option.

Advertising IPv4 Routes

Updated: 2013-08-30

You can configure the NetScaler appliance to advertise host routes to VIPs and to advertise routes to downstream networks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure BGP to advertise IPv4 routes by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter. Possible values: 1 to 4,294,967,295.
Neighbor < IPv4 address> remote-as < as-number>	Update the IPv4 BGP neighbor table with the link local IPv4 address of the neighbor in the specified autonomous system.
Address-family ipv4	Enter address family configuration mode.
Neighbor < IPv4 address> activate	Exchange prefixes for the IPv4 router family between the peer and the local node by using the link local address.
redistribute kernel	Redistribute kernel routes.
redistribute static	Redistribute static routes.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor a1bc::102 remote-as 100
NS(config-router)# Address-family ipv4
NS(config-router-af)# Neighbor 10.102.29.170 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
```

Advertising IPv6 BGP Routes

Updated: 2013-08-30

Border Gateway Protocol (BGP) enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone NetScaler appliances. Route advertising enables an upstream router to track network entities located behind the NetScaler.

To configure BGP to advertise IPv6 routes by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter. Possible values: 1 to 4,294,967,295.
Neighbor < IPv6 address> remote-as < as-number>	Update the IPv6 BGP neighbor table with the link local IPv6 address of the neighbor in the specified autonomous system.
Address-family ipv6	Enter address family configuration mode.
Neighbor < IPv6 address> activate	Exchange prefixes for the IPv6 router family between the peer and the local node by using the link local address.
redistribute kernel	Redistribute kernel routes.
redistribute static	Redistribute static routes.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor a1bc::102 remote-as 100
NS(config-router)# Address-family ipv6
NS(config-router-af)# Neighbor a1bc::102 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
```

Verifying the BGP Configuration

Updated: 2013-08-30

You can use VTYSH to display BGP settings.

To view the BGP settings using the VTYSH command line

At the command prompt, type:

```
VTYSH
```

You are now in the VTYSH command prompt. An output similar to the following appears:

```
NS170#
```

At the VTYSH command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
NS170# sh ip BGP
NS170# sh BGP
NS170# sh ip BGP neighbors
NS170# sh ip BGP summary
NS170# sh ip BGP route-map <map-tag>
```

As-Override Support in Border Gateway Protocol

As a part of BGP loop prevention functionality, if a router receives a BGP packet containing the router's Autonomous System Number (ASN) in the Autonomous Systems (AS) path, the router drops the packet. The assumption is that the packet originated from the router and has reached the place from where it originated.

If an enterprise has several sites with a same ASN, BGP loop prevention causes the sites with an identical ASN to not get linked by another ASN. Routing updates (BGP packets) are dropped when another site receives them.

To solve this issue, BGP AS-Override functionality has been added to the ZebOS BGP routing module of the NetScaler.

With AS-Override enabled for a peer device, when the NetScaler appliance receives a BGP packet for forwarding to the peer, and the ASN of the packet matches that of the peer, the appliance replaces the ASN of the BGP packet with its own ASN number before forwarding the packet.

You can enable AS-Override for a specific neighbor or a group of neighbors (peer group) by using the vtysh command line.

To configure BGP AS-Override for a IPv4 neighbor by using the VTYSH command line

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor < IPv4 address> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <IPv4 address> as-override	Enable BGP as-override for the specified neighbor.

```
> VTYSH NS# configure terminal
NS(config)# router BGP 200
NS(config-router)# Neighbor 192.0.2.100 remote-as 100
NS(config-router)# Neighbor 10.102.29.100 as-override
```

To configure BGP AS-Override for a IPv4 BGP peer group by using the VTYSH command line

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Neighbor <peer group name> **peer-group**	Create a BGP peer group.
Neighbot <IPv4 address> **peer-group** <peer group name>	Associate neighbors to the specified peer group.
Neighbor <peer group name> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <peer group name> as-override	Enable BGP as-override for all the neighbors that are associated with the specified peer group.

```
> VTYSH NS# configure terminal
NS(config)# router BGP 200
NS(config-router)# neighbor external-peers-1 peer-group
NS(config-router)# neighbor 192.0.2.101 peer-group external-peers-1
NS(config-router)# neighbor 192.0.2.102 peer-group external-peers-1
NS(config-router)# neighbor 192.0.2.103 peer-group external-peers-1
NS(config-router)# Neighbor external-peers-1 remote-as 100
NS(config-router)# Neighbor external-peers-1 as-override
```

To configure BGP AS-Override for an IPv6 neighbor by using the VTYSH command line

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor < IPv6 address> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <IPv6 address> as-override	Enable BGP as-override for the specified neighbor.
Address-family ipv6	Enter address family configuration mode.
Neighbor < IPv6 address> activate	Exchange prefixes for the IPv6 router family between the specified neighbor and the NetScaler by using the link local address.
Neighbor <IPv6 address> as-override	Enable BGP as-override for the specified neighbor.

```
> VTYSH NS# configure terminal
NS(config)# router BGP 200
NS(config-router)# Neighbor a1bc::102 remote-as 100
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

NS(config-router)# Neighbor a1bc::102 as-override
NS(config-router)# Address-family ipv6
NS(config-router-af)# Neighbor a1bc::102 activate
NS(config-router)# Neighbor a1bc::102 as-override

To configure BGP AS-Override for IPv6 peer group by using the VTYSH command line

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor <peer group name> **peer-group**	Create a BGP peer group.
Neighbor <IPv6 address> **peer-group** <peer group name>	Associate a neighbor with the specified peer group.
Neighbor <peer group name> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <peer group name> as-override	Enable BGP as-override for all the neighbors that are associated with the specified peer group.
Address-family ipv6	Enter address family configuration mode.
Neighbor <peer group name> activate	Exchange prefixes for the IPv6 router family between the neighbors of the specified peer group and the NetScaler by using the link local address.
Neighbor <peer group name> as-override	
	Enable BGP as-override for all the neighbors that are associated with the specified peer group.

> VTYSH NS# configure terminal
NS(config)# router BGP 200
NS(config-router)# neighbor external-peers-2 peer-group
NS(config-router)# neighbor 2001::1 peer-group external-peers-2
NS(config-router)# neighbor 2001::2 peer-group external-peers-2
NS(config-router)# Neighbor external-peers-2 remote-as 100
NS(config-router)# Neighbor external-peers-2 as-override
NS(config-router)# Address-family ipv6
NS(config-router-af)# Neighbor external-peers-2 activate
NS(config-router)# Neighbor external-peers-2 as-override

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring IPv6 RIP

Mar 20, 2012

IPv6 Routing Information Protocol (RIP) or RIPng is a Distance Vector protocol. This protocol is an extension of RIP to support IPv6. After enabling IPv6 RIP, you need to configure advertisement of IPv6 RIP routes. For troubleshooting, you can limit IPv6 RIP propagation. You can display IPv6 RIP settings to verify the configuration.

Prerequisites for IPv6 RIP

Before you begin configuring IPv6 RIP, do the following:

- Make sure that you understand the IPv6 RIP protocol.
- Install the IPv6PT license on the NetScaler appliance.
- Enable the IPv6 feature.

Enabling IPv6 RIP

Updated: 2013-08-30

You can enable or disable IPv6 RIP by using VTYSH. After you enable IPv6 RIP, the NetScaler starts the IPv6 RIP daemon. After you disable IPv6 RIP, the NetScaler stops the RIP daemon.

To enable IPv6 RIP by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
ns IPv6-routing	Start IPv6 dynamic routing daemon.
interface < vlan_name>	Enter VLAN configuration mode.
router ipv6 RIP	Start IPv6 RIP routing process on the VLAN.

Example

```
> VTYSH
NS# configure terminal
NS(config)# ns IPv6-routing
NS(config)# interface vlan0
NS(config-if)# router ipv6 RIP
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Advertising IPv6 RIP Routes

Updated: 2013-08-30

IPv6 RIP enables an upstream router to load balance traffic between two identical vservers hosted on two standalone NetScaler devices. Route advertisement enables an upstream router to track network entities located behind the NetScaler.

To configure IPv6 RIP to advertise IPv6 routes by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 rip	Start IPv6 RIP routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
```

Limiting IPv6 RIP Propagations

Updated: 2013-08-30

If you need to troubleshoot your configuration, you can configure the listen-only mode on any given interface.

To limit IPv6 RIP propagation by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command	Specifies
router ipv6 rip	Start IPv6 RIP routing process and enter configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# passive-interface VLAN0
```

Verifying the IPv6 RIP Configuration

Updated: 2013-08-30

You can use VTYSH to display the IPv6 RIP routing table and IPv6 RIP information for a specified VLAN.

To view the IPv6 RIP settings by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
sh ipv6 rip	Display updated IPv6 RIP routing table.
sh ipv6 rip interface <vlan_name>	Display IPv6 RIP information for the specified VLAN.

Example

```
NS# VTYSH
NS# sh ipv6 rip
NS# sh ipv6 rip interface VLAN0
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring IPv6 OSPF

Mar 20, 2012

IPv6 OSPF or OSPF version 3 (OSPF v3) is a link state protocol that is used to exchange IPv6 routing information. After enabling IPv6 OSPF, you need to configure advertisement of IPv6 OSPF routes. For troubleshooting, you can limit IPv6 OSPF propagation. You can display IPv6 OSPF settings to verify the configuration.

Prerequisites for IPv6 OSPF

Before you begin configuring IPv6 OSPF, do the following:

- Make sure that you understand the IPv6 OSPF protocol.
- Install the IPv6PT license on the NetScaler appliance.
- Enable the IPv6 feature.

Enabling IPv6 OSPF

Updated: 2013-08-30

To enable IPv6 OSPF, you must use the VTYSH command line. When IPv6 OSPF is enabled, the NetScaler appliance starts the IPv6 OSPF daemon. When IPv6 OSPF is disabled, the appliance stops the IPv6 OSPF daemon.

To enable IPv6 OSPF by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
ns IPv6-routing	Start IPv6 dynamic routing process.
interface < vlan_name>	Enter the VLAN configuration mode.
ipv6 router OSPF area <area-id>	Start IPv6 OSPF routing process on a VLAN.

Example

```
>VTYSH
NS# configure terminal
NS(config)# ns IPv6-routing
NS(config)# interface vlan0
NS(config-if)# ipv6 router OSPF area 3
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Advertising IPv6 Routes

Updated: 2013-08-30

IPv6 OSPF enables an upstream router to load balance traffic between two identical vservers hosted on two standalone NetScaler devices. Route advertising enables an upstream router to track network entities located behind the NetScaler.

To configure IPv6 OSPF to advertise IPv6 routes by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 OSPF	Start IPv6 OSPF routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
```

Limiting IPv6 OSPF Propagations

Updated: 2013-08-30

If you need to troubleshoot your configuration, you use VTYSH to configure listen-only mode on any given VLAN.

To limit IPv6 OSPF propagation by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Commands	Specifies
router ipv6 OSPF	Start IPv6 OSPF routing process and enter configuration mode for the routing process.
passive-interface < vlan_name >	Suppress routing updates on interfaces bound to the specified VLAN.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# passive-interface VLAN0
```

Verifying the IPv6 OSPF Configuration

Updated: 2013-08-30

You use VTYSH to display IPv6 OSPF current neighbors and IPv6 OSPF routes.

To view the IPv6 OSPF settings by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh ipv6 OSPF neighbor	Display current neighbors.
sh ipv6 OSPF route	Display IPv6 OSPF routes.

Example

```
>VTYSH
NS# sh ipv6 OSPF neighbor
NS# sh ipv6 OSPF route
```

OSPFv3 Authentication

Updated: 2015-06-15

To ensure the integrity, data origin authentication, and data confidentiality of OSPFv3 packets, OSPFv3 authentication must be configured on OSPFv3 peers.

The NetScaler appliance supports OSPFv3 authentication and is partially compliant with RFC 4552. OSPFv3 authentication is based on the two IPSec protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP). The NetScaler appliance supports only the AH protocol for OSPFv3 authentication.

OSPFv3 authentication uses manually defined IPSec Security Associations (SAs) between the OSPFv3 peers and does not

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

rely on IKE protocol for forming dynamic SAs. Manual SAs define the security parameter Index (SPI) values, algorithms, and keys to be used between the peers. Manual SAs require no negotiation between the peers; therefore, the same SA must be defined on both the peers.

You can configure OSPFv3 authentication on a VLAN or for an OSPFv3 area. When you configure for a VLAN, the settings are applied to all the interfaces that are members of the VLAN. When you configure OSPFv3 authentication for an OSPF area, the settings are applied to all the VLANs in that area. The settings are in turn applied to all the interfaces that are members of these VLANs. These settings do not apply to member VLANs on which you have configured OSPFv3 authentication directly.

Consider the following points and limitations before configuring OSPFv3 authentication on a NetScaler appliance:

- Make sure that you understand the different components of OSPFv3 authentication, described in RFC 4552.
- Only Authentication Header protocol is supported for OSPFv3 authentication. Encapsulating Security Payload (ESP) is not supported.
- You must define an SA with the same setting on the peer interface.
- Rekeying of manual keys is not supported.

To configure OSPFv3 authentication on a VLAN by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display the VTYSH command prompt.
configure terminal	Enter global configuration mode.
interface < vlan_name>	Enter the VLAN configuration mode.

ipv6 ospf authentication ipsec spi <spi> [**MD5** | **SHA1**] <string>	Set the following OSPFv3 authentication parameters.
	• **SPI**— Configures the Security Parameter Index (SPI) for the manual IPSec SA. Possible values: 256 to 4294967295.
	• **Authentication Algorithms and Key**— MD5 or SHA1 algorithms are used by the AH protocol for OSPFv3 authentication between the OSPFv3 peers. The string specifies the key for the selected authentication algorithm:
	  • An MD5 key specifies a 32-character or 16-byte hexadecimal string.
	  • A SHA1 key specifies a 40-character or 20-byte hexadecimal string.

Example

```
> VTYSH NS# configure terminal
NS(config)# interface vlan2
NS(config-if)# ipv6 ospf authentication ipsec spi 256 md5 123456789ABCDEF0123456789ABCDEF0
```

To configure OSPFv3 authentication on an OSPF area by using the VTYSH command line:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 ospf <process tag>	Enter IPv6 OSPF configuration mode.

area <areaid> **authentication ipsec spi** <spi> [**MD5** | **SHA1**] <string>	Set the following OSPFv3 authentication parameters for the specified OSPF area.
	• **SPI**— Configures the Security Parameter Index (SPI) for the manual IPSec SA. Possible values: 256 to 4294967295.
	• **Authentication Algorithms and Key**— MD5 or SHA1 algorithms are used by the AH protocol for OSPFv3 authentication between the OSPFv3 peers. The string specifies the key for the selected authentication algorithm:
	  • An MD5 key specifies a 32-character or 16-byte hexadecimal string.
	  • A SHA1 key specifies a 40-character or 20-byte hexadecimal string.

Example

```
> VTYSH NS# configure terminal
ns(config)#router ipv6 ospf 30
ns(config-router)# area 1 authentication ipsec spi 256 md5123456789ABCDEF0123456789ABCDEF0
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring ISIS

Mar 02, 2015

The NetScaler appliance supports the Intermediate System-to-Intermediate System (IS-IS or ISIS) dynamic routing protocol. This protocol supports IPv4 as well as IPv6 route exchanges. IS-IS is a link state protocol and is therefore less prone to routing loops. With the advantages of faster convergence and the ability to support larger networks, ISIS can be very useful in Internet Service Provider (ISP) networks.

Prerequisites for configuring ISIS

Before you begin configuring ISIS, do the following:

- Make sure that you understand the ISIS protocol.
- For IPV6 routes, enable:
  - IPv6 protocol translation feature.
  - IPv6 Dynamic Routing option on the VLANs on which you want to run ISIS protocol.

Enabling ISIS

Updated: 2013-08-30

Use either of the following procedures to enable the ISIS routing feature on the NetScaler appliance.

To enable ISIS routing by using the command line interface

At the command prompt, type:

```
enable ns feature ISIS
```

To enable ISIS routing by using the configuration utility

1. Navigate to System > Settings, in Modes and Features group, click Change advanced features.
2. Select or clear the ISIS Routing option.

Creating an ISIS Routing Process and Starting It on a VLAN

Updated: 2013-08-30

To create an ISIS routing process, you must use the VTYSH command line.

At the command prompt, type the following commands, in the order shown:

Command	Description
VTYSH	Displays VTYSH command prompt.
configure terminal	Enters the global configuration mode.
router ISIS [tag]	Creates an ISIS routing process and configuration mode for the routing process.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command	Description
net XX....XXXX.YYYY.YYYY.YYYY.00	Specifies a NET value for the routing process, where: ● · XX. .. .XXXX is the Area Address (can be 1-13 bytes) ● · YYYY.YYYY.YYYY is the System ID (6 bytes) ● · 00 is the N-selector (1 byte) A NET value can be 8 to 20 bytes in length. The last byte is always the n-selector, and must be zero. The n-selector indicates that there is no transport entity and means that the packet is for the routing software of the appliance. The six bytes directly preceding the n-selector are the system ID. The system ID length is fixed and cannot be changed. The system ID must be unique throughout each area (Level 1) and throughout the backbone (Level 2). The bytes preceding the system ID are the area ID, which can be from 1 to 13 bytes in length. A maximum of three NETs per routing process are allowed with different area ID, but the system ID should be the same for all NETs.
is-type (level-1 | level-1-2 | level-2-only)	Sets the ISIS routing process to the specified level of routing. Default: level-1-2.
ns IPv6-routing	Starts the IPv6 dynamic routing daemon.
interface <vlan_name>	Enters the VLAN configuration mode.
ip router ISIS	Enables the ISIS routing process on the VLAN for IPv4 route exchanges.
ipv6 router ISIS	Enables the ISIS routing process on the VLAN for IPv6 route exchanges.

Example

```
> VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# net 15.aabb.ccdd.0097.00
NS(config-router)# is-type level-1
NS(config-router)# exit
NS(config)# ns IPv6-routing
NS(config)# interface vlan0
NS(config-if)#  ip router isis 11
NS(config-if)# ipv6 router isis 11
```

Advertising Routes

Updated: 2013-08-30

Route advertisement enables an upstream router to track network entities located behind the NetScaler appliance.

To configure ISIS to advertise routes by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Description

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Command	Description
VTYSH	Displays the VTYSH command prompt.
configure terminal	Enters the global configuration mode.
router ISIS [tag]	Starts the ISIS routing instance and enter configuration mode for the routing process.
redistribute connected (level-1 | level-1-2 | level-2)	Redistributes connected routes, where • **level-1** : Redistribute connected routes into Level-1. • **level-1-2** : Redistribute connected routes into Level-1 and Level-2. • **level-2** : Redistribute connected routes into Level-2.
redistribute kernel(level-1 | level-1-2 | level-2)	Redistributes kernel routes, where: • **level-1** : Redistribute kernel routes into Level-1. • **level-1-2** : Redistribute kernel routes into Level-1 and Level-2. • **level-2** : Redistribute kernel routes into Level-2.

Example

```
>VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# redistribute connected level-1
NS(config-router)# redistribute kernel level-1
```

Limiting ISIS Propagations

Updated: 2013-08-30

If you need to troubleshoot your configuration, you can configure the listen-only mode on any given VLAN.

To limit ISIS propagation by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Command	Description
VTYSH	Displays the VTYSH command prompt.
configure terminal	Enters the global configuration mode.
router isis [tag]	Enters the configuration mode for the routing process.
passive-interface <vlan_name>	Suppresses routing updates on interfaces bound to the specified VLAN.

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# passive-interface VLAN0
```

Verifying the ISIS Configuration

Updated: 2013-08-30

You can use VTYSH to display the ISIS routing table and ISIS information for a specified VLAN.

To view the ISIS settings by using the VTYSH command line

At the command prompt, type the following commands, in the order shown:

Commands	Description
VTYSH	Displays the VTYSH command prompt.
show ip isis route	Displays updated IPv4 ISIS routing table.
show ipv6 isis route	Displays updated IPv6 ISIS routing table.
sh isis interface <vlan_name>	Displays IPv6 ISIS information for the specified VLAN.

Example

```
NS# VTYSH
NS#  show ip isis route
NS#  show ipv6 isis route
NS# sh isis interface VLAN0
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing Routes to the NetScaler Routing Table

Aug 30, 2013

The NetScaler appliance can use routes learned by various routing protocols after you install the routes in the appliance's routing table.

To install various routes to the internal routing table by using the VTYSH command line

At the command prompt, type the following commands as appropriate for the routes that you want to install:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
ns route-install Default	Install IPv4 default routes to the internal routing table.
ns route-install RIP	Install IPv4 RIP specific routes to the internal routing table.
ns route-install BGP	Install IPv4 BGP specific routes to the internal routing table.
ns route-install OSPF	Install IPv4 OSPF specific routes to the internal routing table.
ns route-install IPv6 Default	Install IPv6 default routes to the internal routing table.
ns route-install IPv6 RIP	Install IPv6 RIP specific routes to the internal routing table.
ns route-install IPv6 BGP	Install IPv6 BGP specific routes to the internal routing table.
ns route-install IPv6 OSPF	Install IPv6 OSPF specific routes to the internal routing table.

Example

```
>VTYSH
NS# configure terminal
NS# ns route-install Default
NS(config)# ns route-install RIP
NS(config)# ns route-install BGP
NS(config)# ns route-install OSPF
NS# ns route-install IPv6 Default
NS(config)# ns route-install IPv6 RIP
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
NS(config)# ns route-install IPv6 BGP
NS(config)# ns route-install IPv6 OSPF
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Static Routes

Sep 06, 2013

Static routes are manually created to improve the performance of your network. You can monitor static routes to avoid service disruptions. Also, you can assign weights to ECMP routes, and you can create null routes to prevent routing loops.

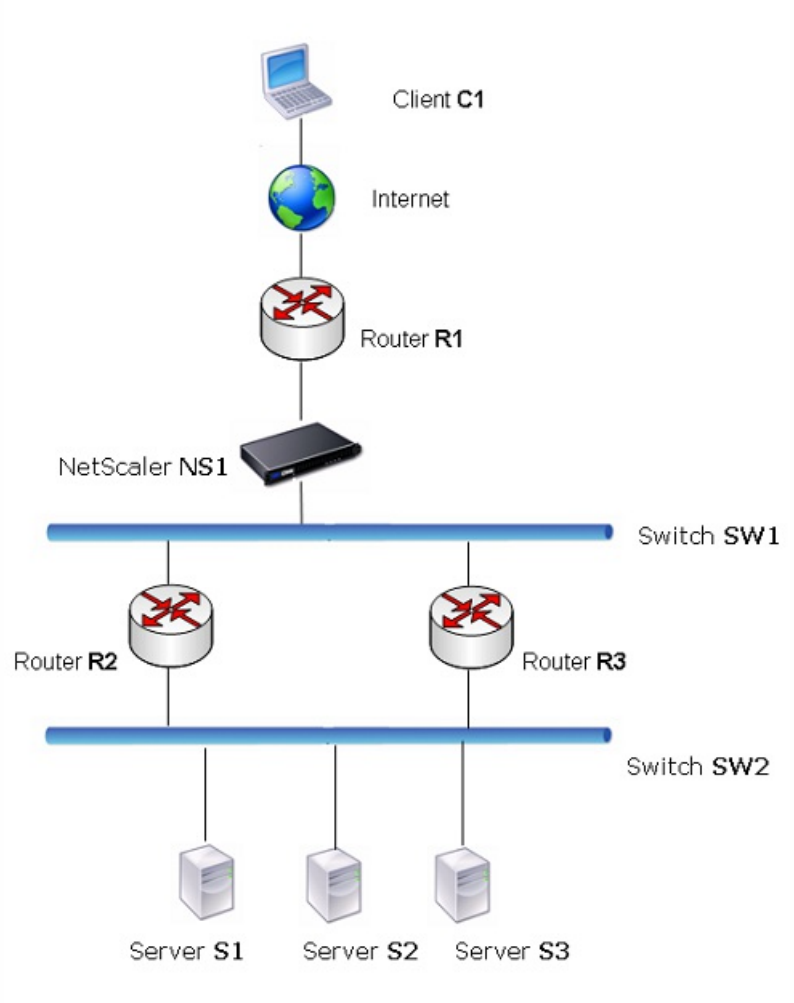
Monitored Static Routes

If a manually created (static) route goes down, a backup route is not automatically activated. You must manually delete the inactive primary static route. However, if you configure the static route as a monitored route, the NetScaler appliance can automatically activate a backup route.

Static route monitoring can also be based on the accessibility of the subnet. A subnet is usually connected to a single interface, but it can be logically accessed through other interfaces. Subnets bound to a VLAN are accessible only if the VLAN is up. VLANs are logical interfaces through which packets are transmitted and received by the NetScaler. A static route is marked as DOWN if the next hop resides on a subnet that is unreachable.

Note: In a high availability (HA) setup, the default value for monitored state routes (MSRs) on the secondary node is UP. The value is set to avoid a state transition gap upon failover, which could result in dropping packets on those routes.
Consider the following simple topology, in which a NetScaler is load balancing traffic to a site across multiple servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Router R1 moves traffic between the client and the NetScaler appliance. The appliance can reach servers S1 and S2 through routers R2 or R3. It has two static routes through which to reach the servers' subnet, one with R2 as the gateway and another with R3 as the gateway. Both these routes have monitoring enabled. The administrative distance of the static route with gateway R2 is lower than that of the static route with gateway R3. Therefore, R2 is preferred over R3 to forward traffic to the servers. Also, the default route on the NetScaler points to R1 so that all Internet traffic exits properly.

If R2 fails while monitoring is enabled on the static route, which uses R2 as the gateway, the NetScaler marks it as DOWN. The NetScaler now uses the static route with R3 as the gateway and forwards the traffic to the servers through R3.

The NetScaler supports monitoring of IPv4 and IPv6 static routes. You can configure the NetScaler to monitor an IPv4 static route either by creating a new ARP or PING monitor or by using existing ARP or PING monitors. You can configure the NetScaler to monitor an IPv6 static route either by creating a new Neighbor discovery for IPv6 (ND6) or PING monitor or by using the existing ND6 or PING monitors.

Weighted Static Routes

When the NetScaler appliance makes routing decisions involving routes with equal distance and cost, that is, Equal Cost Multi-Path (ECMP) routes, it balances the load between them by using a hashing mechanism based on the source and destination IP addresses. For an ECMP route, however, you can configure a weight value. The NetScaler then uses both the weight and the hashed value for balancing the load.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Null Routes

If the route chosen in a routing decision is inactive, the NetScaler appliance chooses a backup route. If all the backup routes become inaccessible, the appliance might reroute the packet to the sender, which could result in a routing loop leading to network congestion. To prevent this situation, you can create a null route, which adds a null interface as a gateway. The null route is never the preferred route, because it has a higher administrative distance than the other static routes. But it is selected if the other static routes become inaccessible. In that case, the appliance drops the packet and prevents a routing loop.

This section includes the following details:

- Configuring IPv4 Static Routes
- Configuring IPv6 Static Routes

Configuring IPv4 Static Routes

Updated: 2013-09-06

You can add a simple static route or a null route by setting a few parameters, or you can set additional parameters to configure a monitored or monitored and weighted static route. You can change the parameters of a static route. For example, you might want to assign a weight to an unweighted route, or you might want to disable monitoring on a monitored route.

To create a static route by using the command line interface

At the command prompt, type the following commands to create a static route and verify the configuration:

- add route <network> <netmask> <gateway>[-cost <positive_integer>] [-advertise (DISABLED | ENABLED)]
- show route [<network> <netmask> [<gateway>]] [<routeType>] [-detail]

Example

```
> add route 10.102.29.0 255.255.255.0 10.102.29.2 -cost 2 -advertise ENABLED
 Done
```

To create a monitored static route by using the command line interface

At the command prompt, type the following commands to create a monitored static route and verify the configuration:

- add route <network> <netmask> <gateway> [-distance <positive_integer>] [-weight <positive_integer>][-msr (ENABLED | DISABLED) [-monitor <string>]]
- show route [<network> <netmask> [<gateway>]] [<routeType>] [-detail]

Example

```
> add route 10.102.29.0 255.255.255.0 10.102.29.3 -distance 5 -weight 6 -msr ENBLED -monitor PING
 Done
```

To create a null route by using the command line interface

At the command prompt type:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- add route <network> <netmask> null
- show route <network> <netmask>

Example

```
> add route 10.102.29.0 255.255.255.0 null
 Done
```

To remove a static route by using the command line interface

At the command prompt, type:

rm route <network> <netmask> <gateway>

Example

```
> rm route 10.102.29.0 255.255.255.0 10.102.29.3
 Done
```

To configure a static route by using the configuration utility

Navigate to System > Network > Routes and, on the Basic tab, add a new static route, or edit an existing static route.

To remove a route by using the configuration utility

Navigate to System > Network > Routes and, on the Basic tab, delete the static route.

Configuring IPv6 Static Routes

Updated: 2013-09-06

You can configure a maximum of six default IPv6 static routes. IPv6 routes are selected on the basis of whether the MAC address of the destination device is reachable. This can be determined by using the IPv6 Neighbor Discovery feature. Routes are load balanced and only source/destination-based hash mechanisms are used. Therefore, route selection mechanisms such as round robin are not supported. The next hop address in the default route need not belong to the NSIP subnet.

To create an IPv6 route by using the command line interface

At the command prompt, type the following commands to create an IPv6 route and verify the configuration:

- add route6 <network> <gateway> [-vlan <positive_integer>]
- show route6 [<network> [<gateway>]

Example

```
> add route6 ::/0 FE80::67 -vlan 5
 Done
```

To create a monitored IPv6 static route by using the command line interface

At the command prompt, type the following commands to create a monitored IPv6 static route and verify the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

configuration:

- add route6 <network> <gateway> [-msr (ENABLED | DISABLED) [-monitor <string>]
- show route6 [<network> [<gateway>]

Example

```
> add route6 ::/0 2004::1 -msr ENABLED -monitor PING
 Done
```

To remove an IPv6 route by using the command line interface

At the command prompt, type:

rm route6 <network> <gateway>

Example

```
> rm route6 ::/0 FE80::67
 Done
```

To configure an IPv6 route by using the configuration utility

Navigate to System > Network > Routes and, on the IPV6 tab, add a new IPv6 route, or edit an existing IPv6 route.

To remove an IPv6 route by using the configuration utility

Navigate to System > Network > Routes and, on the IPV6 tab, delete the IPv6 route.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Route Health Injection Based on Virtual Server Settings

Apr 11, 2014

The following option and parameter are introduced for controlling the Route Health Injection (RHI) functionality of the NetScaler appliance for advertising the route of a VIP address.

- **VSVR_CNTRLD**. It is an option for the (Vserver RHI Level) parameter of a VIP address. When this option is set to the Vserver RHI Level parameter, the RHI behavior for advertising the route of the VIP address depends on the RHI STATE parameter setting on all the associated virtual servers of the VIP address along with their states.
- **RHI STATE**. It is a parameter of virtual server. You can set the RHI STATE parameter to either PASSIVE or ACTIVE. By default, the RHI STATE parameter is set to PASSIVE.

For a VIP address, when RHI (Vserver RHI Level) parameter is set to VSVR_CNTRLD, the following are different RHI behaviours for the VIP address on the basis of RHI STATE settings on the virtual servers associated with the VIP address:

- If you set RHI STATE to PASSIVE on all virtual servers, the NetScaler ADC always advertises the route for the VIP address.
- If you set RHI STATE to ACTIVE on all virtual servers, the NetScaler ADC advertises the route for the VIP address if at least one of the associated virtual servers is in UP state.
- If you set RHI STATE to ACTIVE on some and PASSIVE on others, the NetScaler ADC advertises the route for the VIP address if at least one of the associated virtual servers, whose RHI STATE set to ACTIVE, is in UP state.

Following table displays the sample RHI behaviour for a VIP address on the basis of RHI STATE settings on the virtual servers associated with the VIP address. The NetScaler appliance has two virtual servers V1 and V2 associated with the VIP address:

Associated virtual servers for a VIP	**State 1**	**State 2**	**State 3**	**State 4**
RHI State set to PASSIVE on all virtual servers				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Advertise the route for this VIP address?	Yes	Yes	Yes	Yes
RHI State set to ACTIVE on all virtual servers				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Advertise the route for this VIP address?	Yes	Yes	Yes	No

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Associated virtual servers for a VIP RHI State set to ACTIVE on one virtual server and PASSIVE on the other	State 1	State 2	State 3	State 4
V1 (RHI State = ACTIVE)	UP	UP	DOWN	DOWN
V2 (RHI State = PASSIVE)	UP	DOWN	UP	DOWN
Advertise the route for this VIP address?	Yes	Yes	No	No

To configure RHI for a VIP address, to be based on the RHI (RHI State) parameter setting of the associated virtual servers, perform the following steps:

- Set the RHI (Vserver RHI Level) parameter to VSVR_CNTRLD for the VIP address.
- Set the RHI State parameter for each virtual server associated with the VIP address.

To set the vServer RHI Level for a VIP address by using command line interface

At the command prompt, type:

- set ns ip <IPAddress> [-vserverRHILevel <vserverRHILevel>]

To set the RHI State parameter of a virtual server by using command line interface

At the command prompt, type:

- set lb vserver <name> [-RHIstate (PASSIVE | ACTIVE)]

To set the vServer RHI Level for a VIP address by using configuration utility

1. Navigate to System > Network > IPs.
2. Select a VIP address, and then click Edit.
3. Set the Vserver RHI Level parameter to VSVR_CNTRLD, and then click OK.

To set the RHI State parameter of a virtual server by using configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Select a load balancing virtual server, and then click Edit.
3. Set the RHI State parameter, and then click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Policy-Based Routes

Jun 11, 2012

Policy-based routing bases routing decisions on criteria that you specify. A policy-based route (PBR) specifies criteria for selecting packets and, typically, a next hop to which to send the selected packets. For example, you can configure the NetScaler appliance to route outgoing packets from a specific IP address or range to a particular next hop router. Each packet is matched against each configured PBR, in the order determined by the specified priorities, until a match is found. If no match is found, or if the matching PBR specifies a DENY action, the NetScaler applies the routing table for normal destination-based routing.

A PBR bases routing decisions for the data packets on parameters such as source IP address, source port, destination IP address, destination port, protocol, and source MAC address. A PBR defines the conditions that a packet must satisfy for the NetScaler to route the packet. These actions are known as "processing modes." The processing modes are:

- ALLOW - The NetScaler sends the packet to the designated next-hop router.
- DENY - The NetScaler applies the routing table for normal destination-based routing.

You can create PBRs for outgoing IPv4 and IPv6 traffic.

Many users begin by creating PBRs and then modifying them. To activate a new PBR, you must apply it. To deactivate a PBR, you can either remove or disable it. You can change the priority number of a PBR to give it a higher or lower precedence.

This document includes the following information:

- Configuring a Policy-Based Routes (PBR) for IPv4 Traffic
- Configuring a Policy-Based Routes (PBR6) for IPv6 Traffic

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Policy-Based Routes (PBR) for IPv4 Traffic

Mar 20, 2012

Configuring PBRs involves the following tasks:

- Create a PBR.
- Apply PBRs.
- (Optional) Disable or enable a PBR.
- (Optional) Renumber the priority of the PBR.

Creating or Modifying a PBR

Updated: 2013-10-31

You cannot create two PBRs with the same parameters. If you attempt to create a duplicate, an error message appears.

You can configure the priority of a PBR. The priority (an integer value) defines the order in which the NetScaler appliance evaluates PBRs. When you create a PBR without specifying a priority, the NetScaler automatically assigns a priority that is a multiple of 10.

If a packet matches the condition defined by the PBR, the NetScaler performs an action. If the packet does not match the condition defined by the PBR, the NetScaler compares the packet against the PBR with the next highest priority.

Instead of sending the selected packets to a next hop router, you can configure the PBR to send them to a link load balancing virtual server to which you have bound multiple next hops. This configuration can provide a backup if a next hop link fails.

Consider the following example. Two PBRs, p1 and p2, are configured on the NetScaler and automatically assigned priorities 20 and 30. You need to add a third PBR, p3, to be evaluated immediately after the first PBR, p1. The new PBR, p3, must have a priority between 20 and 30. In this case, you can specify the priority as 25.

To create a PBR by using the command line interface

At the command prompt, type:

- add ns pbr <name> <action> [-srcIP [<operator>] <srcIPVal>] [-srcPort [<operator>] <srcPortVal>] [-destIP [<operator>] <destIPVal>] [-destPort [<operator>] <destPortVal>] [-nextHop <nextHopVal>] [-srcMac <mac_addr>] [-protocol <protocol> | -protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-state (ENABLED | DISABLED)]
- show ns pbr

Example

```
> add ns pbr pbr1  allow -srcip 10.102.37.252 -destip 10.10.10.2 -nexthop 10.102.29.77
 Done
```

To modify the priority of a PBR by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to modify the priority and verify the configuration:

- set ns pbr <name> [-action (ALLOW | DENY)] [-srcIP [<operator>] <srcIPVal>] [-srcPort [<operator>] <srcPortVal>] [-destIP [<operator>] <destIPVal>] [-destPort [<operator>] <destPortVal>] [-nextHop <nextHopVal>] [-srcMac <mac_addr>] [-protocol <protocol> | -protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-state (ENABLED | DISABLED)]
- show ns pbr [<name>]

Example

```
> set ns pbr pbr1 -priority 23
 Done
```

To remove one or all PBRs by using the command line interface

At the command prompt, type one of the following commands:

- rm ns pbr <name>
- clear ns pbrs

Example

```
> rm ns pbr pbr1
 Done
> clear ns PBRs
 Done
```

To create a PBR by using the configuration utility

Navigate to System > Network > PBRs, on the PBRs tab, add a new PBR, or edit an existing PBR.

To remove one or all PBRs by using the configuration utility

Navigate to System > Network > PBRs, on the PBRs tab, delete the PBR.

Applying a PBR

Updated: 2013-08-30

You must apply a PBR to activate it. The following procedure reapplies all PBRs that you have not disabled. The PBRs constitute a memory tree (lookup table). For example, if you create 10 PBRs (p1 - p10), and then you create another PBR (p11) and apply it, all of the PBRs (p1 - p11) are freshly applied and a new lookup table is created. If a session has a DENY PBR related to it, the session is destroyed.

You must apply this procedure after every modification you make to any PBR. For example, you must follow this procedure after disabling a PBR.

Note: PBRs created on the NetScaler appliance do not work until they are applied.

To apply a PBR by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

apply ns PBRs

To apply a PBR by using the configuration utility

1. Navigate to System > Network > PBRs.
2. On the PBRs tab, select the PBR, in the Action list, select Apply.

Enabling or Disabling PBRs

Updated: 2013-08-30

By default, the PBRs are enabled. This means that when PBRs are applied, the NetScaler appliance automatically compares incoming packets against the configured PBRs. If a PBR is not required in the lookup table, but it needs to be retained in the configuration, it must be disabled before the PBRs are applied. After the PBRs are applied, the NetScaler does not compare incoming packets against disabled PBRs.

To enable or disable a PBR by using the command line interface

At the command prompt, type one of the following commands:

- enable ns pbr <name>
- disable ns pbr <name>

Examples

```
> enable ns PBR pbr1
 Done
> show ns PBR pbr1
1)     Name: pbr1
       Action: ALLOW                    Hits: 0
       srcIP = 10.102.37.252
       destIP = 10.10.10.2
       srcMac:                          Protocol:
       Vlan:                            Interface:
       Active Status: ENABLED           Applied Status: APPLIED
       Priority: 10
       NextHop: 10.102.29.77

 Done

> disable ns PBR pbr1
Warning: PBR modified, use 'apply pbrs' to commit this operation
> apply pbrs
 Done
> show ns PBR pbr1
1)     Name: pbr1
       Action: ALLOW                    Hits: 0
       srcIP = 10.102.37.252
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    destIP = 10.10.10.2
    srcMac:                    Protocol:
    Vlan:                      Interface:
    Active Status: DISABLED          Applied Status: NOTAPPLIED
    Priority: 10
    NextHop: 10.102.29.77
Done
```

To enable or disable a PBR by using the configuration utility

1. Navigate to System > Network > PBRs.
2. On the PBRs tab, select the PBR, in the Action list, select Enable or Disable.

Renumbering PBRs

Updated: 2013-08-30

You can automatically renumber the PBRs to set their priorities to multiples of 10.

To renumber PBRs by using the command line interface

At the command prompt, type:

```
renumber ns pbrs
```

To renumber PBRs by using the configuration utility

Navigate to System > Network > PBRs, on the PBRs tab, in the Action list, select Renumber Priority (s).

Use Case - PBR with Multiple Hops

Updated: 2013-08-30

Consider a scenario in which two PBRs, PBR1 and PBR2, are configured on NetScaler appliance NS1. PBR1 routes all the outgoing packets, with source IP address as 10.102.29.30, to next hop router R1. PBR2 routes all the outgoing packets, with source IP address as 10.102.29.90, to next hop router R2. R3 is another next hop router connected to NS1.

If router R1 fails, all the outgoing packets that matched against PBR1 are dropped. To avoid this situation, you can specify a link load balancing (LLB) virtual server in the next hop field while creating or modifying a PBR. Multiple next hops are bound to the LLB virtual server as services (for example R1, R2, and R3). Now, if R1 fails, all the packets that matched against PBR1 are routed to R2 or R3 as determined by the LB method configured on the LLB virtual server.

The NetScaler appliance throws an error if you attempt to create a PBR with an LLB virtual server as the next hop in the following cases:

- Adding another PBR with the same LLB virtual server.
- Specifying a nonexistent LLB virtual server.
- Specifying an LLB virtual server for which the bound services are not next hops.
- Specifying an LLB virtual server for which the LB method is not set to one of the following:
  - LEASTPACKETS
  - LEASTBANDWIDTH
  - DESTIPHASH
  - SOURCEIPHASH

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- WEIGHTDRR
- SRCIPDESTIP_HASH
- LTRM
- CUSTOM LOAD
- Specifying an LLB virtual server for which the LB persistence type is not set to one of the following:
  - DESTIP
  - SOURCEIP
  - SRCDSTIP

The following table lists the names and values of the entities configured on the NetScaler appliance:

Table 1. Sample Values for Creating Entities

Entity Type	Name	IP Address
Link load balancing virtual server	LLB1	NA
Services (next hops)	Router1	1.1.1.254
	Router2	2.2.2.254
	Router3	3.3.3.254
PBRs	PBR1	NA
	PBR2	NA

To implement the configuration described above, you need to:

1. Create services Router1, Router2, and Router3 that represent next hop routers R1, R2, and R3.
2. Create link load balancing virtual server LLB1 and bind services Router1, Router2, and Router3 to it.
3. Create PBRs PBR1 and PBR2, with next hop fields set as LLB1 and 2.2.2.254 (IP address of the router R2), respectively.

To create a service by using the command line interface

At the command prompt, type:

- add service <name> <IP> <serviceType> <port>
- show service <name>

Example

```
> add service Router1 1.1.1.254 ANY *
 Done
> add service Router2 2.2.2.254 ANY *
 Done
> add service Router3 3.3.3.254 ANY *
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done
```

To create a service by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, and create a service.

To create a link load balancing virtual server and bind a service by using the command line interface

At the command prompt, type:

- add lb vserver <name> <serviceType>
- bind lb vserver < name> <serviceName>
- show lb vserver < name>

Example

```
> add lb vserver LLB1 ANY
 Done
> bind lb vserver LLB1 Router1 Router2 Router3
 Done
```

To create a link load balancing virtual server and bind a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and create a virtual server for link load balancing. Specify **ANY** in the **Protocol** field.
   Note: Make sure that **Directly Addressable** is unchecked.
2. Under the **Services** tab, in the **Active** column, select the check box for the service that you want to bind to the virtual server.

To create a PBR by using the command line interface

At the command prompt, type:

- add ns pbr <name> <action> [-srcIP [<operator>] <srcIPVal>] [-nextHop <nextHopVal>]
- show ns pbr

Example

```
> add pbr PBR1 ALLOW -srcIP 10.102.29.30 -nextHop LLB1
 Done
> add pbr PBR2 ALLOW -srcIP 10.102.29.90 -nextHop 2.2.2.254
 Done
```

To create a PBR by using the configuration utility

Navigate to System > Network > PBRs, on the PBRs tab, add a new PBR.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Policy-Based Routes (PBR6) for IPv6 Traffic

Mar 20, 2012

Configuring PBR6s involves the following tasks:

- Create a PBR6.
- Apply PBR6s.
- (Optional) Disable or enable a PBR6.
- (Optional) Renumber the priority of the PBR6.

Creating or Modifying a PBR6

Updated: 2013-09-06

You cannot create two PBR6s with the same parameters. If you attempt to create a duplicate, an error message appears.

You can configure the priority of a PBR6. The priority (an integer value) defines the order in which the NetScaler appliance evaluates PBR6s. When you create a PBR6 without specifying a priority, the NetScaler automatically assigns a priority that is a multiple of 10.

If a packet matches the condition defined by the PBR6, the NetScaler performs an action. If the packet does not match the condition defined by the PBR6, the NetScaler compares the packet against the PBR6 with the next highest priority.

To create a PBR6 by using the command line interface

At the command prompt, type:

- add ns pbr6 <name> <action> [-srcIPv6 [<operator>] <srcIPv6Val>] [-srcPort [<operator>] <srcPortVal>] [-destIPv6 [<operator>] <destIPv6Val>] [-destPort [<operator>] <destPortVal>] [-srcMac <mac_addr>] [-protocol <protocol> |-protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-state (ENABLED | DISABLED)] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-nextHop <nextHopVal>] [-nextHopVlan <positive_integer>]
- show ns pbr

To modify or remove a PBR6 by using the command line interface

To modify a PBR6, type the set pbr6 <name> command and the parameters to be changed, with their new values.

To remove one or all PBR6s by using the command line interface

At the command prompt, type one of the following commands:

- rm ns pbr6 <name>
- clear ns pbr6

To create or modify a PBR6 by using the configuration utility

Navigate to System > Network > PBRs and, on the PBR6s tab, add a new PBR6, or edit an existing PBR6.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To remove one or all PBR6s by using the configuration utility

Navigate to System > Network > PBRs and, on the PBR6s tab, delete the PBR6.

Applying PBR6s

Updated: 2013-08-30

You must apply a PBR6 to activate it. The following procedure reapplies all PBR6s that you have not disabled. The PBR6s constitute a memory tree (lookup table). For example, if you create 10 PBR6s (p6_1 - p6_10), and then you create another PBR6 (p6_11) and apply it, all of the PBR6s (p6_1 - p6_11) are freshly applied and a new lookup table is created. If a session has a DENY PBR6 related to it, the session is destroyed.

You must apply this procedure after every modification you make to any PBR6. For example, you must follow this procedure after disabling a PBR6.

Note: PBR6s created on the NetScaler appliance do not work until they are applied.

To apply PBR6s by using the command line interface

At the command prompt, type:

apply ns PBR6

To apply PBR6s by using the configuration utility

1. Navigate to System > Network > PBRs.
2. On the PBR6s tab, select the PBR6, in the Action list, select Apply.

Enabling or Disabling a PBR6

Updated: 2013-08-30

By default, the PBR6s are enabled. This means that when PBR6s are applied, the NetScaler appliance automatically compares outgoing IPv6 packets against the configured PBR6s. If a PBR6 is not required in the lookup table, but it needs to be retained in the configuration, it must be disabled before the PBR6s are applied. After the PBR6s are applied, the NetScaler does not compare incoming packets against disabled PBR6s.

To enable or disable a PBR6 by using the command line interface

At the command prompt, type one of the following commands:

- enable ns pbr <name>
- disable ns pbr <name>

To enable or disable a PBR6 by using the configuration utility

1. Navigate to System > Network > PBRs.
2. On the PBR6s tab, select the PBR6, in the Action list, select Enable or Disable.

Renumbering PBR6s

Updated: 2013-08-30

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You can automatically renumber the PBR6s to set their priorities to multiples of 10.

To renumber PBR6s by using the command line interface

At the command prompt, type:

```
renumber ns pbr6
```

To renumber PBR6s by using the configuration utility

Navigate to System > Network > PBRs, on the PBR6s tab, in the Action list, select Renumber Priority (s).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

MAC Address Wildcard Mask for PBRs

Sep 06, 2015

A wildcard mask parameter has been introduced for extended PBRs and PBR6s and is used with the source MAC address parameter to define a range of MAC addresses to be match against the source MAC address of outgoing packets.

Wild card masks specify which hexadecimal digits of the MAC address are used and which hexadecimal digits are ignored. The wildcard mask parameter specifies a series of ones and zeroes and has a length of 12 digits. Each digit is a mask for the corresponding hexadecimal digit of the MAC address. A zero digit in the wildcard mask indicates that the corresponding hexadecimal digit of the MAC address must be considered and a one digit indicates that the corresponding hexadecimal digit to be ignored.

The wildcard mask should meet the following conditions:

- Has only one series of zeroes
- Has only one series of ones
- Start with a series of zeroes

The following are some of the examples of valid wildcard masks:

- 000000111111
- 000000011111
- 000011111111

The following are some of the examples of invalid wildcard masks:

- 000000111100
- 111110000000
- 010101010101

For an PBR rule, a wildcard mask of 000000111111 for MAC address 96:fa:95:1d:67:4a defines the MAC address range 96:FA:95:00:00:00 - 96:FA:95:FF:FF:FF. This MAC address range is matched against the source MAC address of the outgoing packets.

To specify a range of source MAC addresses in a PBR rule by using the NetScaler command line

At the command prompt, type:

- **add ns pbr** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show ns pbr** <pbrname>

Example COPY

```
> add ns pbr PBR-1 ALLOW -srcip 192.0.2.34 -srcMac 96:fa:95:1d:67:4a  –srcMacMask 000000111111 -nexthop 198.51.100.1

 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To specify a range of source MAC addresses in an PBR6 rule by using the NetScaler command line

At the command prompt, type:

- **add ns pbr6** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show pbr6** <pbr6name>

Example COPY

```
> add ns pbr6 PBR6-1 ALLOW –srcipv6 2001:db8:0::7 -srcMac 96:fa:95:1d:67:4a  –srcMacMask 000000001111 -nexthop 2001:db8:0::1

 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting Routing Issues

May 11, 2012

To make your troubleshooting process as efficient as possible, begin by gathering information about your network. You need to obtain the following information about the NetScaler appliance and other systems in the Network:

- Complete Topology diagram, including interface connectivity and intermediate switch details.
- Running Configuration. You can use the show running command to get the running configuration for ns.conf and ZebOS.conf.
- Output of the History command, to determine whether any configuration changes were made when the issue arose.
- Output of the Top and ps -ax commands, to determine whether any routing daemon is over utilizing the CPU or is misbehaving.
- Any routing related core files in /var/core - nsm, bgpd, ospfd, or ripd. Check the time stamp to see if they are relevant.
- dr_error.log and dr_info.log files from /var/log.
- Output of the date command and time details for all relevant systems. Print dates across all devices one after another, so that the times on the log messages can be correlated with various events.
- Relevant ns.log, newnslog files.
- Configuration files, log files and command history details from upstream and downstream routers.

This document includes the following information:

- Generic Routing FAQs
- Troubleshooting OSPF-Specific Issues

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Generic Routing FAQs

Mar 20, 2012

Users typically have the following questions about how to troubleshoot generic routing issues:

- How do I save the config files?

  The write command from VTYSH saves only ZebOS.conf. Run the save ns config command from NetScaler CLI to save both ns.conf and ZebOS.conf files.
- If I have configured both a static default route and a dynamically learned default route, which is the preferred default route?

  The dynamically learned route is the preferred default route. This behavior is unique to default routes. However, in case of the Network Services Module (NSM), unless the administrative distances are modified, a statically configured route in the RIB is preferred over a dynamic route. The route that is downloaded to the NSM FIB is the static route.

- How do I block the advertisement of default routes?

  After release 7.0, the default route is not injected into ZebOS.

  However, if you are working with 7.0 or an earlier release, you must apply a suitable route map with the

  However, if you are working with 7.0 or an earlier release, you must apply a suitable route map with the redistribute kernel command for each protocol to block default route advertisement. For example:

```
ns(config)#access-list 1 deny 0.0.0.0
ns(config)#access-list 2 permit any
ns(config)#route-map redist-kernel permit 5
ns(config-route-map)#match ip address 1
ns(config)#route-map redist-kernel permit 10
ns(config-route-map)#match ip address 2
ns(config-route-map)#q
ns(config)#router ospf 1
ns(config-router)#redistribute kernel route-map redist-kernel
ns(config-router)#q
ns(config)#q
ns#show route-map
route-map redist-kernel, permit, sequence 5
  Match clauses:
    ip address 1
  Set clauses:
route-map redist-kernel, permit, sequence 10
  Match clauses:
    ip address 2
  Set clauses:
ns#show access-list
Standard IP access list 1
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
   deny 0.0.0.0
Standard IP access list 2
   permit any
ns#
```

- How do I view the debug output of networking daemons?

  You can write debugging output from networking daemons to a file by entering the following log file command from the global configuration view in VTYSH:

  ```
  ns(config)#log file /var/ZebOS.log
  ```

  With release 8.1, you can direct debug output to the console by entering the terminal monitor command from VTYSH user view:

  ```
  ns#terminal monitor
  ```

- How do I collect cores of running daemons?

  You can use the gcore utility to collect cores of running daemons for processing by gdb. This might be helpful in debugging misbehaving daemons without bringing the whole routing operation to a standstill.

  ```
  gcore [-s] [-c core] [executable] pid
  ```

  The -s option temporarily stops the daemon while gathering the core image. This is a recommended option, because it guarantees that the resulting image shows the core in a consistent state.

  ```
  root@ns#gcore -s -c nsm.core /netscaler/nsm 342
  ```

- How do I run a batch of ZebOS commands?

  You can run a batch of ZebOS commands from a file by entering the VTYSH -f <file-name> command. This does not replace the running configuration, but appends to it. However, by including commands to delete the existing configuration in the batch file and then add those for the new, desired configuration, you can use this mechanism to replace a specific configuration:

  ```
  !
  router bgp 234
  network 1.1.1.1 255.255.255.0
  !
  route-map bgp-out2 permit 10
   set metric 9900
   set community 8602:300
  !
  ```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting OSPF-Specific Issues

Mar 20, 2012

Before you start debugging any OSPF specific issue, you must collect information from the NetScaler appliance and all systems in the affected LAN, including upstream and downstream routers. To begin, enter the following commands:

1. show interface from both nscli and VTYSH
2. show ip ospf interface
3. show ip ospf neighbor detail
4. show ip route
5. show ip ospf route
6. show ip ospf database summary
   1. If there are only few LSAs in the database, then enter show ip ospf database router, show ip ospf database A. network, show ip ospf database external, and other commands to get the full details of LSAs.
   2. If there are a large number of LSAs in the database, enter the show ip ospf database self-originated command.
7. show ip ospf
8. show ns ip. This ensures that the details of all VIPs of interest are included.
9. Get the logs from peering devices and run the following command:

   ```
   gcore -s -c xyz.core /netscaler/ospfd <pid>
   ```

Note: The gcore command is non-disruptive.

Collect additional information from the NetScaler as follows:

1. Enable logging of error messages by entering the following command from the global configuration view in VTYSH:

   ```
   ns(config)#log file /var/ospf.log
   ```

2. Enable debugging ospf events and log them by using the following command:

   ```
   ns(config)#log file /var/ospf.log
   ```

   Enable debug ospf lsa packet only if the number of LSAs in the database is relatively small (< 500).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Internet Protocol version 6 (IPv6)

Mar 20, 2012

A NetScaler appliance supports both server-side and client-side IPv6 and can therefore function as an IPv6 node. It can accept connections from IPv6 nodes (both hosts and routers) and from IPv4 nodes, and can perform Protocol Translation (RFC 2765) before sending traffic to the services. You have to license the IPv6 feature before you can implement it.

The following table lists some of the IPv6 features that the NetScaler appliance supports.

Table 1. Some Supported IPv6 Features

IPv6 features
IPv6 addresses for SNIPs (NSIP6, VIP6, and SNIP6)
Neighbor Discovery (Address Resolution, Duplicated Address Detection, Neighbor Unreachability Detection, Router Discovery)
Management Applications (ping6, telnet6, ssh6)
Static Routing and Dynamic routing (OSPF, BGP, RIPng, and ISIS)
Port Based VLANs
Access Control Lists for IPv6 addresses (ACL6)
IPv6 Protocols (TCP6, UDP6, ICMP6)
Server Side Support (IPv6 addresses for vservers, services)
USIP (Use source IP) and DSR (Direct Server Return) for IPv6
SNMP and CVPN for IPv6
HA with native IPv6 node address
IPv6 addresses for MIPs
Path-MTU discovery for IPv6

The following table lists NetScaler components that support IPv6 addresses and provides references to the topics that document the components.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 2. NetScaler Components That Support IPv6 Addresses and the Corresponding Documentation

NetScaler component	Topic that documents IPv6 support
Network	Adding, Customizing, Removing, Removing all, and Viewing routes.
SSL Offload	Creating IPv6 vservers for SSL Offload
SSL Offload	Specifying IPv6 SSL Offload Monitors
SSL Offload	Creating IPv6 SSL Offload Servers
Load Balancing	Creating IPv6 vservers for Load Balancing
Load Balancing	Specifying IPv6 Load Balancing Monitors
Load Balancing	Creating IPv6 Load Balancing Servers
DNS	Creating AAAA Records

You can configure IPv6 support for the above features after implementing the IPv6 feature on your NetScaler appliance. You can configure both tagged and prefix-based VLANs for IPv6. You can also map IPv4 addresses to IPv6 addresses.

Implementing IPv6 Support

IPv6 support is a licensed feature, which you have to enable before you can use or configure it. If IPv6 is disabled, the NetScaler does not process IPv6 packets. It displays the following warning when you run an unsupported command:

"Warning: Feature(s) not enabled [IPv6PT]"

The following message appears if you attempt to run IPv6 commands without the appropriate license:

"ERROR: Feature(s) not licensed"

After licensing the feature, use either of the following procedures to enable or disable IPv6.

To enable or disable IPv6 by using the command line interface

At the command prompt, type one of the following commands:

- enable ns feature ipv6pt
- disable ns feature ipv6pt

To enable or disable IPv6 by using the configuration utility

1. Navigate to System > Settings, in the Modes and Features group, click Configure Advanced Features.
2. Select or clear the IPv6 Protocol Translation option.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

VLAN Support

Updated: 2013-08-30

If you need to send broadcast or multicast packets without identifying the VLAN (for example, during DAD for NSIP, or ND6 for the next hop of the route), you can configure the NetScaler appliance to send the packet on all the interfaces with appropriate tagging. The VLAN is identified by ND6, and a data packet is sent only on the VLAN.

For more information about ND6 and VLANs, see "Configuring Neighbor Discovery."

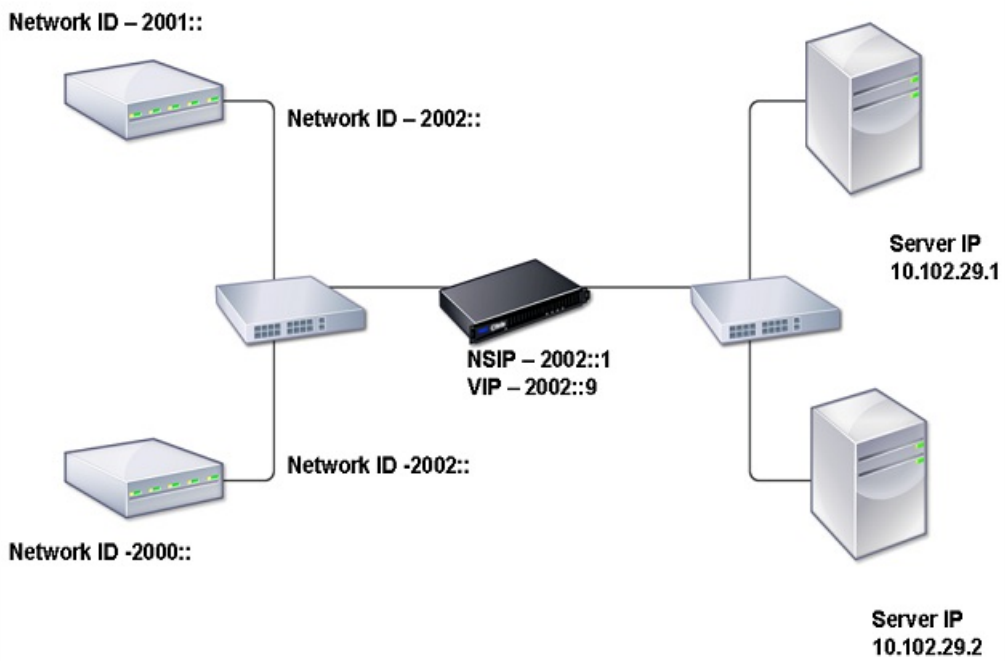
Port-based VLANs are common for IPv4 and IPv6. Prefix-based VLANs are supported for IPv6.

Simple Deployment Scenario

Updated: 2013-08-30

Following is an example of a simple load balancing set-up consisting of an IPv6 vserver and IPv4 services, as illustrated in the following topology diagram.

Figure 1. IPv6 Sample Topology



The following table summarizes the names and values of the entities that must be configured on the NetScaler.

Table 3. Sample Values for Creating Entities

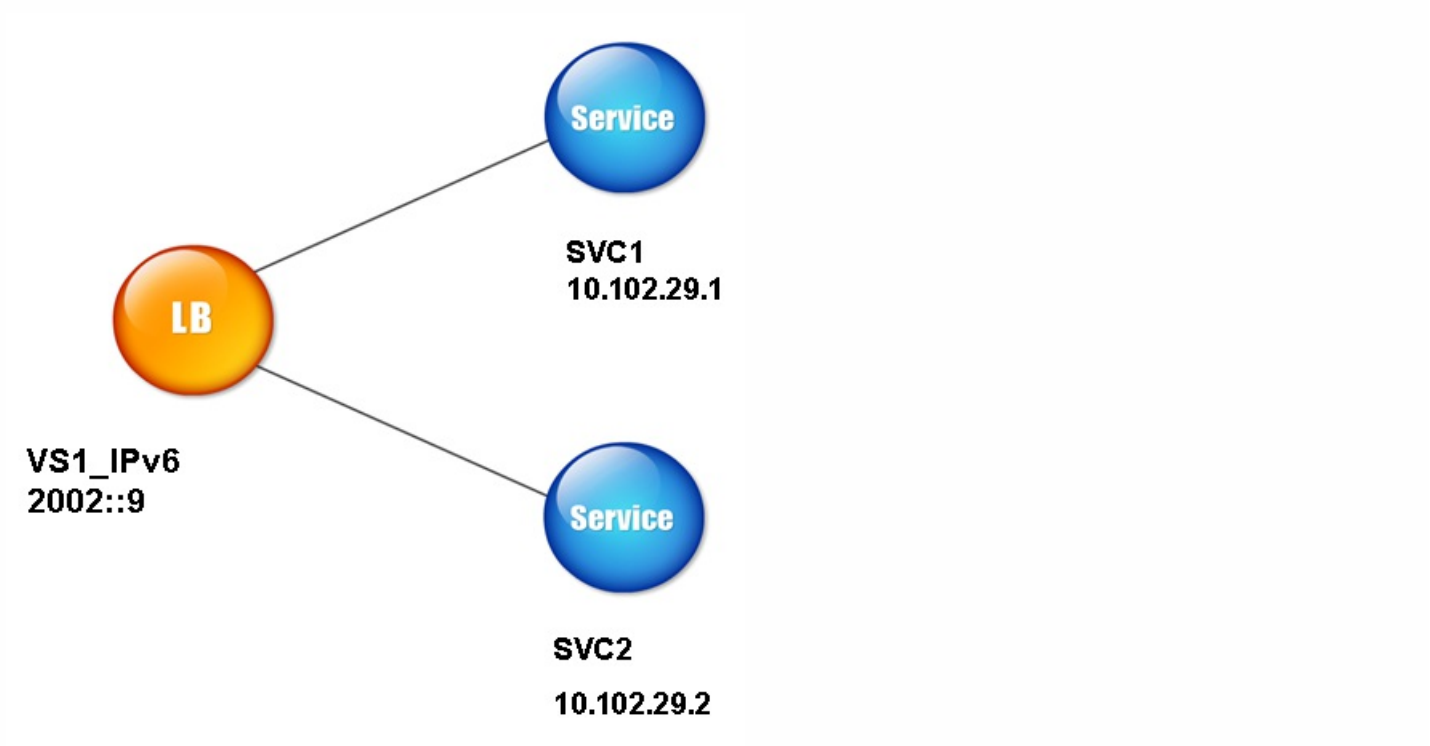
Entity type	Name	Value
LB Vserver	VS1_IPv6	2002::9
Services	SVC1	10.102.29.1
	SVC2	10.102.29.2

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity type	Name	Value

The following figure shows the entities and values of the parameters to be configured on the NetScaler.

Figure 2. IPv6 Entity Diagram



To configure this deployment scenario, you need to do the following:

1. Create an IPv6 service.
2. Create an IPv6 LB vserver.
3. Bind the services to the vserver.

To create IPv4 services by using the command line interface

At the command prompt, type:

add service <Name> <IPAddress> <Protocol> <Port>

Example

```
add service SVC1 10.102.29.1 HTTP 80
add service SVC2 10.102.29.2 HTTP 80
```

To create IPv4 services by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, click Add, and then set the following parameters:

- Service Name
- IP Address
- Protocol
- Port

To create IPv6 vserver by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

add lb vserver <Name> <IPAddress> <Protocol> <Port>

Example

add lb vserver VS1_IPv6 2002::9 HTTP 80

To create IPv6 vserver by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, click Add, and select the IPv6 check box.
2. Set the following parameters:
   - Name
   - Protocol
   - IP Address Type
   - IP Address
   - Port

To bind a service to an LB vserver by using the command line interface

At the command prompt, type:

bind lb vserver <name> <service>

Example

bind lb vserver VS1_IPv6 SVC1

The vservers receive IPv6 packets and the NetScaler performs Protocol Translation (RFC 2765) before sending traffic to the IPv4-based services.

To bind a service to an LB vserver by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the Load Balancing Virtual Servers page, select the vserver for which you want to bind the service (for example, VS1_IPv6).
3. Click Open.
4. In the Configure Virtual Server (Load Balancing) dialog box, on the Services tab, select the Active check box corresponding to the service that you want to bind to the vserver (for example, SVC1).
5. Click OK.
6. Repeat Steps 1-4 to bind the service (for example, SVC2 to the vserver).

Host Header Modification

Updated: 2013-08-30

When an HTTP request has an IPv6 address in the host header, and the server does not understand the IPv6 address, you must map the IPv6 address to an IPv4 address. The IPv4 address is then used in the host header of the HTTP request sent to the vserver.

To change the IPv6 address in the host header to an IPv4 address by using the command line interface

At the command prompt, type:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

set ns ip6 <IPv6Address> -map <IPAddress>
Example

```
set ns ip6 2002::9 -map 200.200.200.200
```

To change the IPv6 address in the host header to an IPv4 address by using the configuration utility

1. Navigate to System > Network > IPs and, on the IPV6s tab, select the IP address for which you want to configure a mapped IP address, for example, 2002:0:0:0:0:0:0:9, and click Edit.
2. In the Mapped IP text box, type the mapped IP address that you want to configure, for example, 200.200.200.200.

VIP Insertion

Updated: 2013-08-30

If an IPv6 address is sent to an IPv4-based server, the server may not understand the IP address in the HTTP header, and may generate an error. To avoid this, you can map an IPv4 address to the IPv6 VIP and enable VIP insertion.

To configure a mapped IPv6 address by using the command line interface

At the command prompt, type:

set ns ip6 <IPv6Address> -map <IPAddress>
Example

```
> set ns ip6 2002::9 -map 200.200.200.200
 Done
```

To configure a mapped IPv6 address by using the configuration utility

1. Navigate to System > Network > IPs, on the IPV6s tab, select the IP address for which you want to configure a mapped IP address, for example, 2002:0:0:0:0:0:0:9, and click Edit.
2. In the Mapped IP text box, type the mapped IP address that you want to configure, for example, 200.200.200.200.

Use either of the following procedures to enable insertion of an Ipv4 VIP address and port number in the HTTP requests sent to the servers.

To enable VIP insertion by using the command line interface

At the command prompt, type:

set lb vserver <name> -insertVserverIPPort <Value>
Example

```
> set lb vserver VS1_IPv6 -insertVserverIPPort ON
 Done
```

To enable VIP insertion by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, select the vserver that you want to enable port

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

insertion, and click Edit.

2. In the Advanced tab, under Traffic Settings, in the Vserver IP Port Insertion drop-down list box, select VIPADDR.
3. In the Vserver IP Port Insertion text box, type the vip header.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Traffic Domains

Jun 17, 2018

> **Important**
>
> Citrix recommends that you use Admin Partitioning instead of using Traffic Domains. For more information about Admin Partitions, see Admin Partitioning page.

Traffic domains are a way to segment network traffic for different applications. You can use traffic domains to create multiple isolated environments within a NetScaler appliance. An application belonging to a specific traffic domain communicates with entities and processes traffic within that domain. The traffic belonging to one traffic domain cannot cross the boundary of another traffic domain.

Benefits of using Traffic Domains

The main benefits of using traffic domains on a NetScaler appliance are the following:

- **Use of duplicate IP addresses in a Network**. Traffic domains allow you to use duplicate IP address on the network. You can assign the same IP address or network address to multiple devices on a network, or multiple entities on a NetScaler appliance, as long as each of the duplicate address belongs to a different traffic domain.
- **Use of Duplicate entities on the NetScaler appliance**. Traffic domains also allow you to use duplicate NetScaler feature entities on the appliance. You can create entities with the same settings as long as each entity is assigned to a separate traffic domain.
  Note: Duplicate entities with same name is not supported.
- **Multitenancy**. Using traffic domains, you can provide hosting services for multiple customers by isolating each customer's type of application traffic within a defined address space on the network.

A traffic domain is uniquely identified by an identifier, which is an integer value. Each traffic domain needs a VLAN or a set of VLANs. The isolation functionality of the traffic domain depends on the VLANs bound to the traffic domain. More than one VLAN can be bound to a traffic domain, but the same VLAN cannot be a part of multiple traffic domains. Therefore, the maximum number of traffic domains that can be created depends on the number of VLANS configured on the appliance.

Default Traffic Domain

A NetScaler appliance has a preconfigured traffic domain, called the *default traffic domain*, which has an ID of 0. All factory settings and configurations are part of the default traffic domain. You can create other traffic domains and then segment traffic between the default traffic domain and each of the other traffic domains. You cannot remove the default traffic domain from the NetScaler appliance. Any feature entity that you create without setting the traffic domain ID is automatically associated with the default traffic domain.
Note: Some features and configurations are supported only in the default traffic domain. They do not work in nondefault traffic domains. For a list of the features supported in all traffic domains, see Supported NetScaler Features in Traffic Domains.

This section includes the following details:

- How Traffic Domains Work
- Supported NetScaler Features in Traffic Domains

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Configuring Traffic Domains

How Traffic Domains Work

As an illustration of traffic domains, consider an example in which two traffic domains, with IDs 1 and 2, are configured on NetScaler appliance NS1.

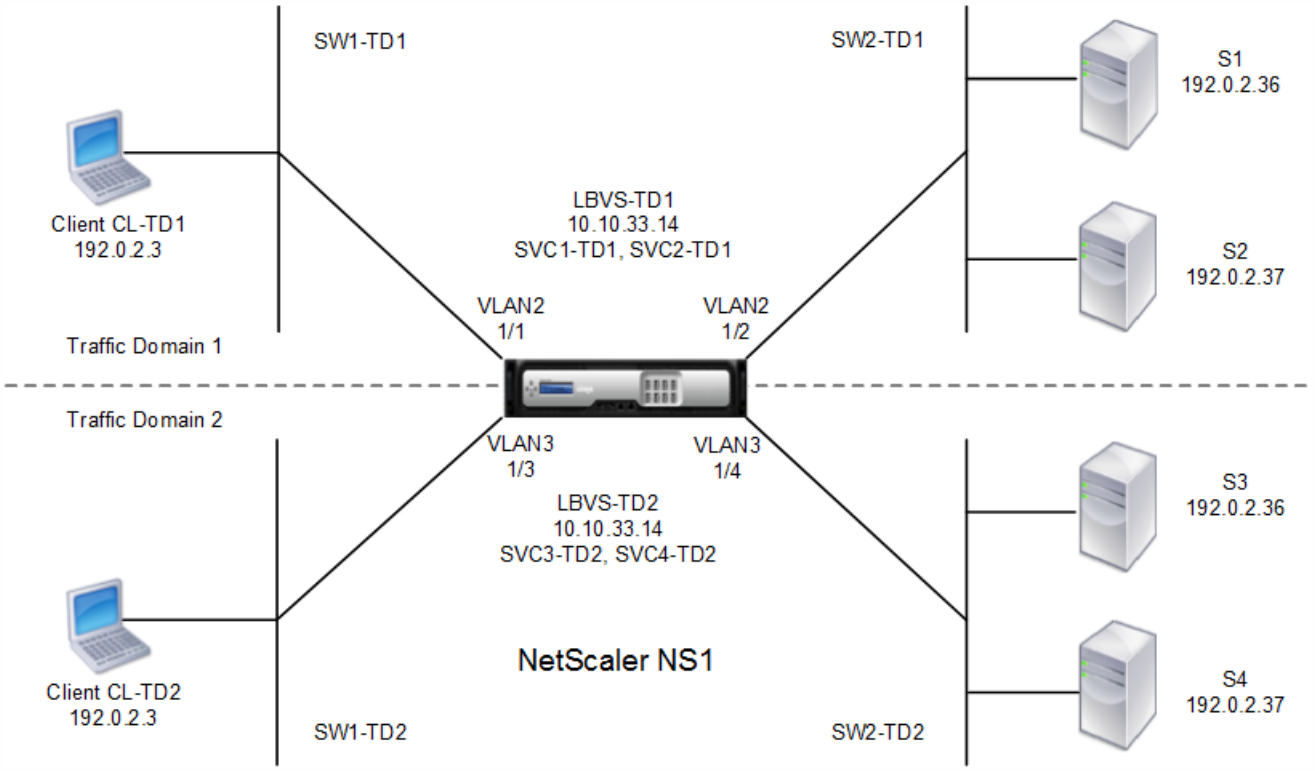
In traffic domain 1, load balancing virtual server LBVS-TD1 is configured to load balance traffic across servers S1 and S2. On the NetScaler appliance, servers S1 and S2 are represented by services SVC1-TD1 and SVC2-TD1, respectively. Servers S1 and S2 are connected to NS1 through L2 switch SW2-TD1. Client CL-TD1 is on a private network connected to NS1 through L2 switch SW1-TD1. SW1-TD1 and SW2-TD1 are connected to VLAN 2 of NS1. VLAN 2 is bound to traffic domain 1, which means that client CL-TD1 and servers S1 and S2 are part of traffic domain 1.

Similarly in traffic domain 2, load balancing virtual server LBVS-TD2 is configured to load balance traffic across S3 and S4. On the NetScaler appliance, servers S3 and S4 are represented by services SVC3-TD2 and SVC4-TD2, respectively. Servers S3 and S4 are connected to NS1 through L2 switch SW2-TD2. Client CL-TD2 is on a private network connected to NS1 through L2 switch SW1-TD2. SW1-TD2 and SW2-TD2 are connected to VLAN 3 of NS1. VLAN 3 is bound to traffic domain 2, which means that client CL-TD2 and servers S3 and S4 are part of traffic domain 2.

On the NetScaler appliance, entities LBVS-TD1 and LBVS-TD2 share the same settings, including the IP address. The same is true for SVC1-TD1 and SVC3-TD2, and for SVC2-TD1 and SVC4-TD2. This is possible because these entities are in different traffic domains.

Similarly, servers S1 and S3, S2 and S4 share the same IP address, and clients CL-TD1 and CL-TD2 each have the same IP address.

Figure 1. How traffic domains work



The following table lists the settings used in the example.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity	Name	Details
Settings in traffic domain 1		
VLANs bound to traffic domain 1	VLAN 2	VLAN Id: 2 Interfaces bound: 1/1, 1/2
Client connected to TD1	CL-TD1 (for reference purposes only)	IP address: 192.0.2.3
Load balancing virtual server in TD1	LBVS-TD1	IP address: 192.0.2.15
Service bound to virtual server LBVS-TD1	SVC1-TD1	IP address: 192.0.2.36
	SVC2-TD1	IP address: 192.0.2.37
SNIP	SNIP-TD1 (for reference purposes only)	IP address: 192.0.2.27
Settings in traffic domain 2		
VLAN bound to traffic domain 2	VLAN 3	VLAN Id: 3 Interfaces bound: 1/3, 1/4
Client connected to TD2	CL-TD2 (for reference purposes only)	IP address: 192.0.2.3
Load balancing virtual server in TD2	LBVS-TD2	IP address: 192.0.2.15
Service bound to virtual server LBVS-TD2	SVC3-TD2	IP address: 192.0.2.36
	SVC4-TD2	IP address: 192.0.2.37
SNIP in TD2	SNIP-TD2 (for reference purposes only)	IP address: 192.0.2.29

Following is the traffic flow in traffic domain 1:

1. Client CL-TD1 broadcasts an ARP request for the IP address of 192.0.2.15.
2. The ARP request reaches NS1 on interface 1/1, which is bound to VLAN 2. Because VLAN 2 is bound to traffic domain 1, NS1 updates the ARP table of traffic domain 1 for the IP address of client CL-TD1.
3. Because the ARP request is received on traffic domain 1, NS1 looks for an entity configured on traffic domain 1 that has an IP address of 192.0.2.15. NS1 finds that a load balancing virtual server LBVS-TD1 is configured on traffic domain 1 and has the IP address 192.0.2.15.
4. NS1 sends an ARP response with the MAC address of interface 1/1.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. The ARP reply reaches CL-TD1. CL-TD1 updates its ARP table for the IP address of LBVS-TD1 with the MAC address of interface 1/1 of NS1.
6. Client CL-TD1 sends a request to 192.0.2.15. The request is received by LBVS-TD1 on port 1/1 of NS1.
7. LBVS-TD1's load balancing algorithm selects server S2, and NS1 opens a connection between a SNIP in traffic domain 1 (192.0.2.27) and S2.
8. S2 replies to SNIP 192.0.2.27 on NS1.
9. NS1 sends S2's reply to client CL-TD1.

Following is the traffic flow in traffic domain 2:

1. Client CL-TD2 broadcasts an ARP request for the IP address of 192.0.2.15.
2. The ARP request reaches NS1 on interface 1/3, which is bound to VLAN 3. Because VLAN 3 is bound to traffic domain 2, NS1 updates traffic-domain 2's ARP-table entry for the IP address of client CL-TD2, even though an ARP entry for the same IP address (CL-TD1) is already present in the ARP table of traffic domain 1.
3. Because the ARP request is received in traffic domain 2, NS1 searches traffic domain 2 for an entity that has an IP address of 192.0.2.15. NS1 finds that load balancing virtual server LBVS-TD2 is configured in traffic domain 2 and has the IP address 192.0.2.15. NS1 ignores LBVS-TD1 in traffic domain 1, even though it has the same IP address as LBVS-TD2.
4. NS1 sends an ARP response with the MAC address of interface 1/3.
5. The ARP reply reaches CL-TD2. CL-TD2 updates its ARP table entry for the IP address of LBVS-TD2 with the MAC address of interface 1/3 of NS1.
6. Client CL-TD2 sends a request to 192.0.2.15. The request is received by LBVS-TD2 on interface 1/3 of NS1.
7. LBVS-TD2's load balancing algorithm selects server S3, and NS1 opens a connection between a SNIP in traffic domain 2 (192.0.2.29) and S3.
8. S2 replies to SNIP 192.0.2.29 on NS1.
9. NS1 sends S2's reply to client CL-TD2.

Supported NetScaler Features in Traffic Domains

The NetScaler features in the following list are supported in all traffic domains.

> **Important**
>
> Any NetScaler feature not listed below is supported only in the default traffic domain.

• ARP table • ND6 table • Bridge table • All types of IPv4 and IPv6 addresses • IPv4 and IPv6 routes • ACL and ACL6 • PBR & PBR6 • INAT • RNAT • RNAT6 • MSR • MSR6 • Net profiles • SNMP MIBs • Fragmentation • Monitors (Scriptable monitors are not supported) • Content Switching	• Persistency • Service (Domain based services are not supported) • Servicegroup (Domain based service groups are not supported) • Policies (*) • PING • TRACEROUTE • PMTU • High Availability (connection mirroring is not supported) • Cluster (Supported on L2 clusters. Not supported on L3 clusters) • Cookie Persistency • MSS • Logging • Priority Queuing • Surge Protection • HTTP DOSP (**) • Load balancing (The following types are not supported: • TFTP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

• Cache Redirection	• RTSP • Diameter • SIP • SMPP) • NAT46 • NAT64 • DNS64 • Forwarding Session Rules • SNMP

* Policies do not have global binding points for traffic domains. However, policies can be bound to a specific load balancing virtual server of a traffic domain.

** HTTP DOSP policies do not have global binding points for traffic domains. However, HTTP DOSP policies can be bound to a specific load balancing service of a traffic domain.

> **Note**
>
> Global Server Loading Balancing (GSLB) and ADNS features in NetScaler are not aware of Traffic Domains. If the GSLB configuration needs to be shared across all traffic domains then GSLB methods Static Proximity and Round Trip Time (RTT) do not work. As a workaround in this scenario you can use GSLB methods other than RTT and Static Proximity. For more information, see http://support.citrix.com/article/CTX202277.

Configuring Traffic Domains

Configuring a traffic domain on the NetScaler appliance consists of the following tasks:

- **Add VLANs.** Create VLANs and bind specified interfaces to them.
- **Create a traffic domain entity and bind VLANs to it.** This involves the following two tasks:
  - Create a traffic domain entity uniquely identified by an ID, which is an integer value.
  - Bind the specified VLANs to the traffic domain entity. All the interfaces that are bound to the specified VLANs are associated with the traffic domain. More than one VLAN can be bound to a traffic domain, but a VLAN cannot be a part of multiple traffic domains.
- **Create feature entities on the traffic domain**. Create the required feature entities in the traffic domain. The CLI commands and configuration dialog boxes of all the supported features in a nondefault traffic domain include a parameter called a *traffic domain identifier* (td). When configuring a feature entity, if you want the entity to be associated with a particular traffic domain, you must specify the td. Any feature entity that you create without setting the td is automatically associated with the default traffic domain.

To give you an idea of how feature entities are associated with a traffic domain, this topic covers the procedures for configuring all the entities mentioned in the figure titled How Traffic Domains Work.

The command line interface has two commands for these two tasks, but the configuration utility combines them in a single dialog box.

To create a VLAN and bind interfaces to it by using the command line interface

At the command prompt, type:

- add vlan <id>
- bind vlan <id> -ifnum <slot/port>
- show vlan <id>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To create a traffic domain entity and bind VLANs to it by using the command line interface

At the command prompt, type:

- add ns trafficdomain <td>
- bind ns trafficdomain <td> -vlan <id>
- show ns trafficdomain <td>

To create a service by using the command line interface

At the command prompt, type:

- add service <name> <IP> <serviceType> <port> -td <id>
- show service <name>

To create a load balancing virtual server and bind services to it by using the command line interface

At the command prompt, type:

- add lb vserver <name> <serviceType> <IPAddress> <port> -td <id>
- bind lb vserver <name> <serviceName>
- show lb vserver <name>

To create a VLAN by using the configuration utility

Navigate to System > Network > VLANs, click Add, and set the parameters.

To create a traffic domain entity by using the configuration utility

Navigate to System > Network > Traffic Domains, click Add, and in the Create Traffic Domain dialog box, set the parameters.

To create a service by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, click Add, and set the parameters.

To create a load balancing virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers, click Add, and set the parameters.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Inter Traffic Domain Entity Bindings

Jun 17, 2014

You can bind services in one traffic domain to a virtual server in another traffic domain. All the services to be bound to a virtual server in a different traffic domain must reside in the same traffic domain.

You configure this support by using the existing bind lb vserver command or the related configuration utility procedure.

This capability can facilitate interaction between different traffic domains. In an enterprise, servers can be grouped in different traffic domains. Virtual servers are created in a traffic domain that faces the internet. A virtual server from this traffic domain can be configured to load balance servers in another traffic domain. This virtual server receives connection requests from the Internet to be forwarded to the bound servers.

When a NetScaler ADC is used in a cloud infrastructure, each tenant can be assigned a separate traffic domain, and all the resources (including servers) for a tenant can be grouped together in the tenant's traffic domain. For each tenant, a virtual server is created for load balancing servers in its traffic domain. All of these virtual servers are grouped together in a single traffic domain that faces the Internet.

Consider an example of in which cloud service provider Example-Cloud-A has three traffic domains, with IDs 10, 20, and 30, configured on NetScaler appliance NS1.

Example-Org-A and Example-Org-B are tenants of Example-Cloud-A. Tenant A is assigned traffic domain 20, and tenant B is assigned domain 30. Servers S1 and S2 reside in traffic domain 20 and servers S3 and S4 reside in traffic domain 30.

Traffic domain 10 faces the internet. Virtual servers LBVS-1 and LBVS-2 are created in traffic domain 10. LBVS-1, in traffic domain 10, is configured to load balance servers S1 and S2, which are in traffic domain 20. LBVS-2, in traffic domain 10, is configured to load balance servers S3 and S4, which are in traffic domain 30.

Therefore, these virtual servers accept Internet connection requests for servers that are in a different traffic domain than that of the virtual servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VMAC Based Traffic Domains

Jan 31, 2011

You can associate a traffic domain with a VMAC address instead of with VLANs. The NetScaler ADC then sends the traffic domain's VMAC address in all responses to ARP queries for network entities in that domain. As a result, the ADC can segregate subsequent incoming traffic for different traffic domains on the basis of the destination MAC address, because the destination MAC address is the VMAC address of a traffic domain. After creating entities on a traffic domain, you can easily manage and monitor them by performing traffic domain level operations.

Following are points to consider before you configure VMAC based traffic domain:

1. VMAC based traffic domains are easiest way to achieve network traffic segregation.
2. Because VMAC based traffic domains segregate network traffic based on VMAC addresses and not VLANS, you cannot create duplicate IP addresses on different VMAC based traffic domains on a NetScaler ADC.
3. VMAC based traffic domains do not work when the NetScaler is deployed only in L2 Mode.
4. Both VLAN and VMAC based traffic domains can coexist on a NetScaler ADC. VMAC based traffic domains actually runs on all VLANs that are not bound to any VLAN based traffic domain.

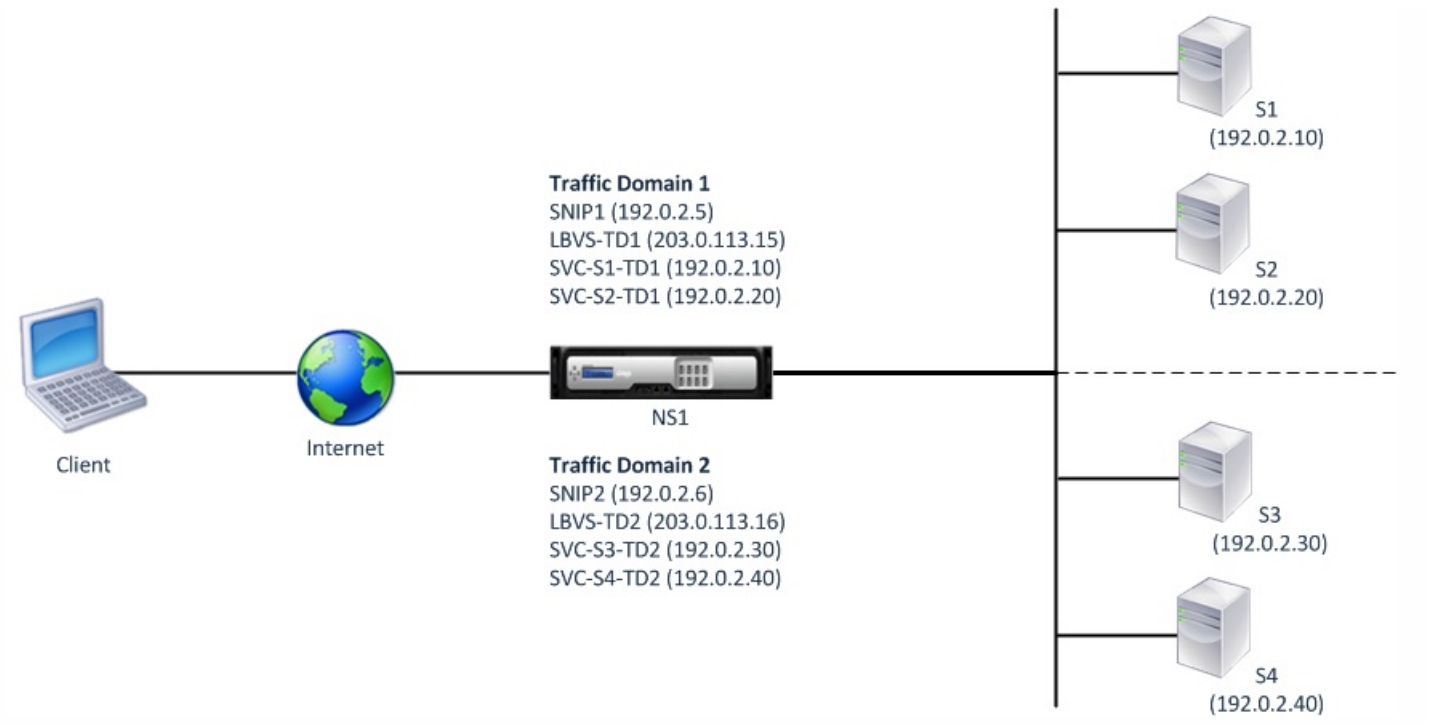
Consider an example in which two traffic domains, with IDs 1 and 2, are configured on NetScaler appliance NS1. The NetScaler creates a VMAC address VMAC1 and associates it with traffic domain 1. Similarly, the NetScaler created another VMAC address VMAC2 and associates with traffic domain 2.

In traffic domain 1, load balancing virtual server LBVS-TD1 is configured to load balance traffic across servers S1 and S2. On the NetScaler appliance, servers S1 and S2 are represented by services SVC1-TD1 and SVC2-TD1, respectively. A subnet IP address (SNIP) SNIP1 is configured for enabling the NetScaler to communicate with S1 and S2. Because VMAC1 is associated with traffic domain 1, the NetScaler sends VMAC1 as the MAC address in all ARP announcements and ARP responses for LBVS-TD1 and SNIP1.

Similarly in traffic domain 2, load balancing virtual server LBVS-TD2 is configured to load balance traffic across S3 and S4. On the NetScaler appliance, servers S3 and S4 are represented by services SVC3-TD2 and SVC4-TD2, respectively. A SNIP address SNIP2 is configured for enabling the NetScaler to communicate with S3 and S4. Because VMAC2 is associated with traffic domain 2, the NetScaler sends VMAC2 as the MAC address in all ARP announcements and ARP responses for LBVS-TD2 and SNIP2.

The NetScaler segregate subsequent incoming traffic for traffic domains 1 or 2 on the basis of the destination MAC address, if the destination MAC address is VMAC1 or VMAC2.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The following table lists the settings used in the example.

Entity	Name	Details
Settings in traffic domain 1		
VMAC Address	VMAC1 (for reference purposes only)	NS1 automatically creates VMAC1 and associates with traffic domain 1
SNIP address	SNIP1 (for reference purposes only)	192.0.2.5
Services on NS1 representing servers S1 and S2	SVC-S1-TD1	• IP address: 192.0.2.10 • Protocol: HTTP • Port: 80
	SVC-S2-TD1	• IP address: 192.0.2.20 • Protocol: HTTP • Port: 80
Load balancing virtual server	LBVS-TD1	• IP address: 203.0.113.15 • Protocol: HTTP • Port: 80 • Bound services: SVC-S1, SVC-S2
Settings in traffic domain 2		

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity	Name	Details
VMAC Address	VMAC2 (for reference purposes only)	NS1 automatically creates VMAC2 and associates with traffic domain 2
SNIP address	SNIP2(for reference purposes only)	192.0.2.6
Services on NS1 representing servers S1 and S2	SVC-S3-TD2	• IP address: 192.0.2.30 • Protocol: HTTP • Port: 80
	SVC-S4-TD2	• IP address: 192.0.2.40 • Protocol: HTTP • Port: 80
Load balancing virtual server	LBVS-TD2	• IP address: 203.0.113.16 • Protocol: HTTP • Port: 80 • Bound services: SVC-S3, SVC-S4

Configuration Steps

Configuring a VMAC based traffic domain on a NetScaler appliance consists of the following tasks:

- Create a traffic domain entity and enable the VMAC option. Create a traffic domain entity uniquely identified by an ID, which is an integer value, and then enable the VMAC option. After creating the traffic domain entity, the NetScaler ADC creates a virtual MAC address and then associates it to the traffic domain entity.
- Create feature entities on the traffic domain. Create the required feature entities in the traffic domain by specifying the traffic domain identifier (td) when configuring these feature entities. NetScaler owned network entities created in a VMAC based traffic domain are associated with the VMAC address, which is associated with the traffic domain. The NetScaler ADC then sends the traffic domain's VMAC address in ARP announcements and ARP responses for these network entities.

To create a VMAC based traffic domain by using the command line interface

At the command prompt, type:

- add ns trafficDomain <td> [-vmac (ENABLED | DISABLED)]
- show ns trafficdomain <td>

To configure a SNIP address by using the command line interface

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type SNIP –td <id>
- show ns ip <IPAddress> -td <id>

To create a service by using the command line interface

At the command prompt, type:

- add service <name> <IP> <serviceType> <port> -td <id>
- show service <name> -td <id>

To create a load balancing virtual server and bind services to it by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

- add lb vserver <name> <serviceType> <IPAddress> <port> -td <id>
- bind lb vserver <name> <serviceName>
- show lb vserver <name> -td <id>

Example

```
> add ns trafficDomain 1 -vmac ENABLED
 Done
> add ns trafficDomain 2 -vmac ENABLED
 Done

> add ns ip 192.0.2.5 255.255.255.0 -type -SNIP -td 1
 Done
> add service  SVC-S1-TD1  192.0.2.10 HTTP 80 -td 1
 Done
> add service  SVC-S2-TD1  192.0.2.20 HTTP 80 -td 1
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.15 80 -td 1
 Done
> bind lb vserver  LBVS-TD1 SVC-S1-TD1
 Done
> bind lb vserver  LBVS-TD1 SVC-S2-TD1
 Done

> add ns ip 192.0.2.6 255.255.255.0 -type -SNIP -td 2
 Done
> add service  SVC-S3-TD2  192.0.2.30 HTTP 80 -td 2
 Done
> add service  SVC-S4-TD2  192.0.2.40 HTTP 80 -td 2
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.16 80 -td 1
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
```

To create a VMAC based traffic domain by using the configuration utility

1. Navigate to System > Network > Interfaces.
2. In the details pane, click Add.
3. On the Create Traffic Domain page, set the following parameters:
   - Traffic Domain ID*
   - Enable Mac
4. Click Create.

To configure a SNIP address by using the configuration utility

1. Navigate to System > Network > IPs > IPv4

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Navigate to Network > IPs > IPv4
3. In the details pane, click Add
4. In the Create IP page, set the following parameters. For a description of a parameter, hover the mouse cursor over the corresponding field.
   - IP Address
   - Netmask
   - IP Type
   - Traffic Domain ID
5. Click Create.

To create a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Basic Settings Page, set the following parameters. For a description of a parameter, hover the mouse cursor over the corresponding field.
   - Service Name
   - Server
   - Protocol
   - Port
   - Traffic Domain ID
4. Click Continue, and click Done.
5. Repeat steps 2-4 to create another service.
6. Click Close.

To create a load balancing virtual server and bind services to it by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the Load Balancing Virtual Servers pane, click Add.
3. In the Create Virtual Servers (Load Balancing) dialog box, set the following parameters. For a description of a parameter, hover the mouse cursor over the corresponding field.
   - Name
   - IP Address
   - Protocol
   - Port
   - Traffic Domain ID
4. Click Continue, on the Service Pane, click >.
5. On the Service page, click Insert, and then select the check box for the services that you want to bind to the virtual server.
6. Click Continue, and click Done.
7. Repeat steps 2-5 to create another virtual server

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VXLAN

Aug 07, 2014

NetScaler appliances support Virtual eXtensible Local Area Networks (VXLANs). A VXLAN overlays Layer 2 networks onto a layer 3 infrastructure by encapsulating Layer-2 frames in UDP packets. Each overlay network is known as a VXLAN Segment and is identified by a unique 24-bit identifier called the *VXLAN* Network Identifier (VNI). Only network devices within the same VXLAN can communicate with each other.

VXLANs provide the same Ethernet Layer 2 network services that VLANs do, but with greater extensibility and flexibility. The two main benefits of using VXLANs are the following:

- **Higher scalability.** Server virtualization and cloud computing architectures have dramatically increased the demand for isolated Layer 2 networks in a datacenter. The VLAN specification uses a 12-bit VLAN ID to identify a Layer 2 network, so you cannot scale beyond 4094 VLANs. That number can be inadequate when the requirement is for thousands of isolated Layer 2 networks. The 24-bit VNI accommodates up to 16 million VXLAN segments in the same administrative domain.
- **Higher flexibility.** Because VXLAN carries Layer 2 data frames over Layer 3 packets, VXLANs extend L2 networks across different parts of a datacenter and across geographically separated datacenters. Applications that are hosted in different parts of a datacenter and in different datacenters but are part of the same VXLAN appear as one contiguous network.

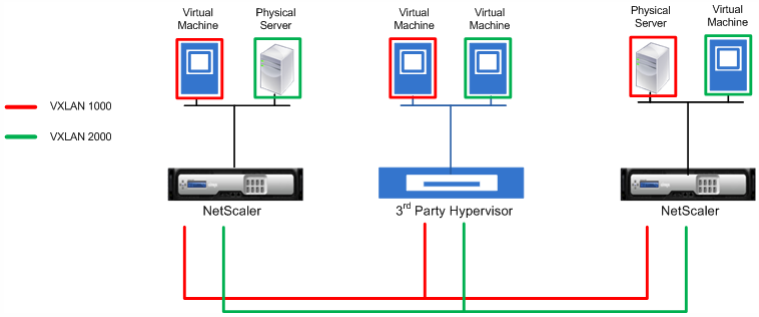
This section includes the following details:

- How VXLANs Work
- VXLAN Use Case: Load Balancing across Datacenters
- Points to Consider for Configuring VXLANs
- Configuration Steps
- Support of IPv6 Dynamic Routing Protocols on VXLANs

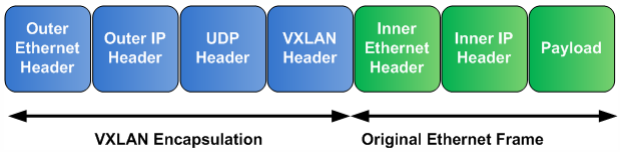
How VXLANs Work

VXLAN Segments are created between VXLAN Tunnel End Points (VTEPs). VTEPs support the VXLAN protocol and perform VXLAN encapsulation and decapsulation. You can think of a VXLAN segment as a tunnel between two VTEPs, where one VTEP encapsulates a Layer2 frame with a UDP header and an IP header and sends it through the tunnel. The other VTEP receives and decapsulates the packet to get the Layer 2 frame. A NetScaler ADC is one example of a VTEP. Other examples are third-party hypervisors, VXLAN aware virtual machines, and VXLAN capable switches.

The following illustration displays virtual machines and physical servers connected through VXLAN tunnels.



The following illustration displays the format of a VXLAN packet.



VXLANs on a NetScaler ADC use a Layer 2 mechanism for sending broadcast, multicast, and unknown unicast frames. A VXLAN supports the following modes for sending these L2 frames.

- Unicast mode: In this mode, you specify the IP addresses of VTEPs while configuring a VXLAN on a NetScaler ADC. The NetScaler ADC sends broadcast, multicast, and unknown unicast frames over Layer 3 to all VTEPs of this VXLAN.
- Multicast mode: In this mode, you specify a multicast group IP address while configuring a VXLAN on a NetScaler ADC. NetScaler ADCs do not support Internet Group Management Protocol (IGMP) protocol. NetScaler ADCs rely on the upstream router to join a multicast group, which shares a common multicast group IP address. The NetScaler ADC sends broadcast, multicast, and unknown unicast frames over Layer 3 to the multicast group IP address of this VXLAN.

Similar to a Layer 2 bridge table, NetScaler ADCs maintain VXLAN mapping tables based on the inner and outer header of the received VXLAN packets. This table maps of remote host MAC addresses to VTEP IP addresses for a particular VXLAN. The NetScaler ADC uses the VXLAN mapping table to look up the destination MAC address of a Layer 2 frame. If an entry for this MAC address is present in the VXLAN table, the NetScaler ADC sends the Layer 2 frame over Layer 3, using the VXLAN protocol, to the mapped VTEP IP address specified in the mapping entry for a VXLAN.

On a NetScaler ADC, you configure a VXLAN by creating a VXLAN tunnel and a VXLAN entity, and then bind the VXLAN tunnel to the VXLAN entity. A VXLAN tunnel is an IP tunnel with VXLAN as the underlying protocol. The VXLAN tunnel also specifies the local VTEP IP address and the remote VTEP IP address. The local VTEP IP address can be one of the configured subnet IP addresses on the NetScaler ADC. The remote IP address can be the IP address of a VTEP or the IP address of a multicast group. The VXLAN entity specifies the desired VXLAN Network Identifier (VNI).

Because VXLANs function similarly to VLANs, most of the NetScaler features that support VLAN as a classification parameter support VXLAN. These features include an optional VXLAN parameter setting, which specifies the VXLAN VNI.

In a high availability (HA) configuration, the VXLAN configuration is propagated or synchronized to the secondary node.

VXLAN Use Case: Load Balancing across Datacenters

To understand the VXLAN functionality of a NetScaler ADC, consider an example in which Example Corp hosts a site at www.example.com. To ensure application availability, the site is hosted on three servers, S0, S1, and S2. A load balancing virtual server, LBVS, on NetScaler ADC NS-ADC is used to load balance these servers. S0, S1, and S2 reside in datacenters DC0, DC1, and DC2, respectively. In DC0, server S0 is connected to NS-ADC.

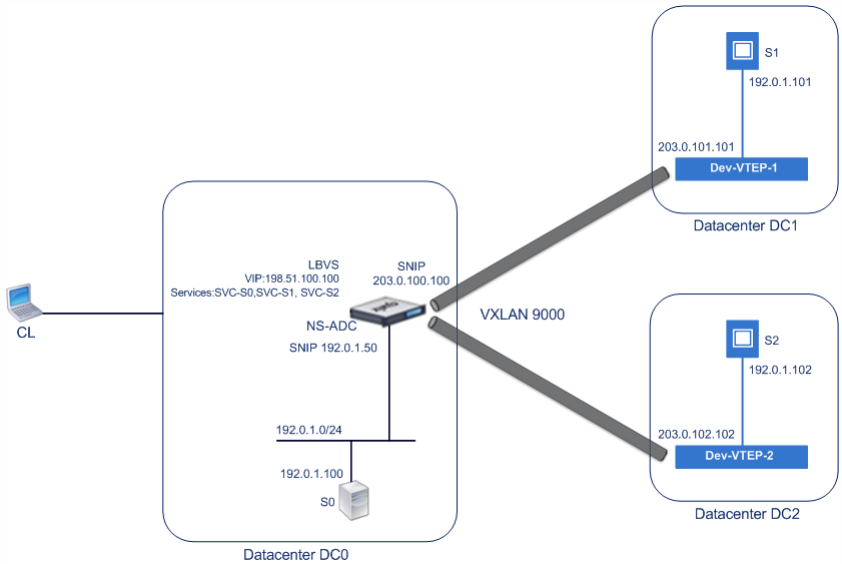
S0 is a physical server, and S1 and S2 are virtual machines (VMs). S1 runs on virtualization host device Dev-VTEP-1 in datacenter DC1, and S2 runs on host device Dev-VTEP-2 in DC2. NS-ADC, Dev-VTEP-1, and Dev-VTEP-2 support the VXLAN protocol.

S0, S1, and S2 are part of the same private subnet, 192.0.1.0/24. For enabling NS-ADC, S0, S1, and S2 be part of a common broadcast domain, VXLAN 9000 is configured on NS-ADC, Dev-VTEP-1, and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Dev-VTEP-2. Servers S1 and S2 are made part of VXLAN9000 on Dev-VTEP-1 and Dev-VTEP-2, respectively.

On NS-ADC, VXLAN 9000 configuration consists of a VXLAN entity, with an ID (VNI) of 9000, and two IP tunnels, VXLAN-9000-Tunnel-1 and VXLAN-9000-Tunnel-2. Both tunnels use VXLAN as the tunnel protocol. VXLAN-9000-Tunnel-1 is a VXLAN tunnel between a SNIP address (SNIP-VTEP-0) of NS-ADC and the IP address of Dev-VTEP-1. VXLAN-9000-Tunnel-2 is a VXLAN tunnel between a SNIP address (SNIP-VTEP-0) of NS-ADC and the IP address of Dev-VTEP-2. Both VXLAN tunnels are bound to VXLAN 9000.



The following table lists the settings used in this example.

Entity	Name	Details
Settings on NetScaler ADC NS-ADC in Datacenter DC-0		
Subnet IP address used by NS-ADC for communicating with servers S0, S1, and S2	SNIP-for-Servers (for reference purposes only)	IP address: 192.0.1.50
Local VTEP IP address, of type SNIP, for VXLAN 9000	SNIP-VTEP-0 (for reference purposes only)	IP address: 203.0.100.100
IP Tunnels of type VXLAN	VXLAN-9000-Tunnel-1	• Remote IP: 203.0.101.101 • Local IP: 203.0.100.100 • Tunnel protocol: VXLAN
	VXLAN-9000-Tunnel-2	• Remote IP: 203.0.102.102 • Local IP: 203.0.100.100 • Tunnel protocol: VXLAN
VXLAN Entity	VXLAN-9000 (For reference purposes only)	• ID (VNI): 9000 • UDP Port: 4789 • Bound VXLAN tunnels: VXLAN-9000-Tunnel-1, • VXLAN-9000-Tunnel-2
Server S0		• IP address: 192.0.1.100
Services on NS-ADC representing servers S0, S1, and S2	SVC-S0	• IP address: 192.0.1.100 • Protocol: HTTP • Port: 80
	SVC-S1	• IP address: 192.0.1.101 • Protocol: HTTP • Port: 80
	SVC-S2	• IP address: 192.0.1.102 • Protocol: HTTP • Port: 80 • Bound services: SVC-S1, SVC-S2
Settings on datacenter DC-1		

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity	Name	Details
VXLAN Settings on Dev-VTEP-2		• ID (VNI): 9000 • UDP Port : 4789 • Remote VTEP IP address: 203.0.100.100 • Local VTEP IP address: 203.0.101.101
Server S1		• IP address: 192.0.1.101
Settings on datacenter DC-2		
VXLAN Settings on Dev-VTEP-2		• ID (VNI): 9000 • UDP Port : 4789 • Remote VTEP IP address: 203.0.100.100 • Local VTEP IP address: 203.0.102.102
Server S2		• IP address: 192.0.1.102

Services SVC-S0, SVC-S1, and SVC-S2 on NS-ADC represent S0, S1, and S2. As soon as these services are configured, NS-ADC broadcasts ARP requests for S0, S1, and S2 to resolve IP-to-MAC mapping. These ARP requests are also sent over VXLAN 9000 to Dev-VTEP-1 and Dev-VTEP-2.

Following is the traffic flow for resolving the ARP request for S2:

1. NS-ADC broadcasts an ARP request for S2 to resolve IP-to-MAC mapping. This packet has:
   - Sourced IP address = Subnet IP address SNIP-for-Servers (192.0.1.50)
   - Source MAC address = MAC address of the NS-ADC's interface from which the packet is sent out = NS-MAC-1
2. NS-ADC prepares the ARP packet to be sent over the VXLAN 9000 by encapsulating the packet with following headers:
   - VXLAN header with an ID (VNI) of 9000
   - Standard UDP header, UDP checksum set to 0×0000, and destination port set to 4789.
3. NS-ADC sends the resulting encapsulated packet to Dev-VTEP-1 and Dev-VTEP-2 on tunnels VXLAN-9000-Tunnel-1 and VXLAN-9000-Tunnel-2, respectively. The encapsulated packet has:
   - Source IP address = SNIP-VTEP-0 (203.0.100.100).
4. Dev-VTEP-2 receives the UDP packet and decapsulates the UDP header, from which Dev-VTEP-2 learns that the packet is a VXLAN related packet. Dev-VTEP-2 then decapsulates the VXLAN header and learns the VXLAN ID of the packet. The resulting packet is the ARP request packet for S2, which is same as in step 1.
5. From the inner and outer header of the VXLAN packet, Dev-VTEP-2 makes an entry in its VXLAN mapping table that shows the mapping of MAC address (NS-MAC-1) and SNIP-VTEP-0 (203.0.100.100) for VXLAN9000.
6. Dev-VTEP-2 sends the ARP packet to S2. S2's response packet reaches Dev-VTEP-2. Dev-VTEP-2 performs a lookup in its VXLAN mapping table and gets a match for the destination MAC address NS-MAC-1. The Dev-VTEP-2 now knows that NS-MAC-1 is reachable through SNIP-VTEP-0 (203.0.100.100) over VXLAN 9000.
7. S2 responds with its MAC address (MAC-S2). The ARP response packet has:
   - Destination IP address = Subnet IP address SNIP-for-Servers (192.0.1.50)
   - Destination MAC address = NS-MAC-1
8. S2's response packet reaches Dev-VTEP-2. Dev-VTEP-2 performs a lookup in its VXLAN mapping table and gets a match for the destination MAC address NS-MAC-1. The Dev-VTEP-2 now knows that NS-MAC-1 is reachable through SNIP-VTEP-0 (203.0.100.100) over VXLAN 9000. Dev-VTEP-2 encapsulates the ARP response with VXLAN and UDP headers, and sends the resultant packet to SNIP-VTEP-0 (203.0.100.100) of NS-ADC.
9. NS-ADC on receiving the packet, decapsulates the packet by removing the VXLAN and UDP headers. The resultant packet is S2's ARP response. NS-ADC updates it VXLAN mapping table for S2's MAC address (MAC-S2) with Dev-VTEP-2's IP address (203.0.102.102) for VXLAN 9000. NS-ADC also updates it ARP table for S2's IP address (192.0.1.102) with S2's MAC address (MAC-S2).

Following is the traffic flow for load balancing virtual server LBVS in this example:

1. Client CL sends a request packet to LBVS of NS-ADC. The request packet has:
   - Source IP address = IP address of client CL (198.51.100.90)
   - Destination IP address = IP address (VIP) of LBVS = 198.51.110.100
2. LBVS of NS-ADC receives the request packet, and its load balancing algorithm selects server S2 of datacenter DC2.
3. NS-ADC processes the request packet, changing its destination IP address to the IP address of S2 and its source IP address to one of the Subnet IP (SNIP) addresses configured on NS-ADC. The request packet has:
   - Source IP address = Subnet IP address on NS-ADC= SNIP-for-Servers (192.0.1.50)
   - Destination IP address = IP address of S2 (192.0.1.102)
4. NS-ADC finds a VXLAN mapping entry for S2 in its bridge table. This entry indicates that S2 is reachable through Dev-VTEP-2 over VXLAN 9000.
5. NS-ADC prepares the packet to be sent over the VXLAN 9000 by encapsulating the packet with following headers:
   - VXLAN header with an ID (VNI) of 9000
   - Standard UDP header, UDP checksum set to 0×0000, and destination port set to 4789.
6. NS-ADC sends the resulting encapsulated packet to Dev-VTEP-2. The request packet has:
   - Source IP address = SNIP address = SNIP-VTEP-0 (203.0.100.100)
   - Destination IP address = IP address of Dev-VTEP-2 (203.0.102.102)
7. Dev-VTEP-2 receives the UDP packet and decapsulates the UDP header, from which Dev-VTEP-2 learns that the packet is a VXLAN related packet. Dev-VTEP-2 then decapsulates the VXLAN header and learns the VXLAN ID of the packet. The resulting packet is the same packet as in step 3.
8. Dev-VTEP-2 then forwards the packet to S2.
9. S2 processes the request packet and sends the response to the SNIP address of NS-ADC. The response packet has:
   - Source IP address = IP address of S2 (192.0.1.102)
   - Destination IP address = Subnet IP address on NS-ADC= SNIP-for-Servers (192.0.1.50)
10. Dev-VTEP-2 encapsulates the response packet in the same way that NS-ADC encapsulated the request packet in steps 4 and 5. Dev-VTEP-2 then sends the encapsulated UDP packet to SNIP address SNIP-for-Servers (192.0.1.50) of NS-ADC.
11. NS-ADC, upon receiving the encapsulated UDP packet, decapsulates the packet by removing the UDP and VXLAN headers in the same way that Dev-VTEP-2 decapsulated the packet in step 7. The resultant packet is the same response packet as in step 9.
12. NS-ADC then uses the session table for load balancing virtual server LBVS, and forwards the response packet to client CL. The response packet has:
    - Source IP address = IP address of client CL (198.51.100.90)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Destination IP address = IP address (VIP) of LBVS (198.51.110.100)

Points to Consider for Configuring VXLANs

Consider the following points before configuring VXLANs on a NetScaler ADC:

- A maximum of 2048 VXLANs can be configured on a NetScaler ADC.
- VXLANs are not supported in a cluster.
- Link-local IPv6 addresses cannot be configured for each VXLAN.
- NetScaler ADCs do not support Internet Group Management Protocol (IGMP) protocol to form a multicast group. NetScaler ADCs rely on the IGMP protocol of its upstream router to join a multicast group, which share a common multicast group IP address. You can specify a multicast group IP address while creating a VXLAN tunnel but the multicast group must be configured on the upstream router. The NetScaler ADC sends broadcast, multicast, and unknown unicast frames over Layer 3 to the multicast group IP address of this VXLAN. The upstream router then forwards the packet to all the VTEPs that are part of the multicast group.
- VXLAN encapsulation adds an overhead of 50 bytes to each packet:
  Outer Ethernet Header (14) + UDP header (8) + IP header (20) + VXLAN header (8) = 50 bytes

  To avoid fragmentation and performance degradation, you must adjust the MTU settings of all network devices in a VXLAN pathway, including the VXLAN VTEP devices, to handle the 50 bytes of overhead in the VXLAN packets.

  Important: Jumbo frames are not supported on the NetScaler VPX virtual appliances, NetScaler SDX appliances, and NetScaler MPX 15000/17000 appliances. These appliances support an MTU size of only 1500 bytes and cannot be adjusted to handle the 50 bytes overhead of VXLAN packets. VXLAN traffic might be fragmented or suffer performance degradation, if one of these appliances is in the VXLAN pathway or acts as a VXLAN VTEP device.
- On NetScaler SDX appliances, VLAN filtering does not work for VXLAN packets.
- IPv6 Dynamic Routing is not supported on VXLAN.
- You cannot set a MTU value on a VXLAN.
- You cannot bind interfaces to a VXLAN.

Configuration Steps

Configuring a VXLAN on a NetScaler appliance consists of the following tasks.

- **Create an IP tunnel of type VXLAN**. Create an IP tunnel with VXLAN as the protocol for the IP tunnel. You specify one of the configured SNIP address for the local IP address of the tunnel. For the remote IP address you can specify either a multicast group IP address or a unicast address of a VTEP device. If you specify a multicast group IP address, you must configure IGMP on the upstream router of the NetScaler ADC to join the multicast group. Also, you can specify the configured VLAN through which the NetScaler ADC sends VXLAN packets to the multicast group IP address. In other words, the upstream router must be available on this VLAN. The upstream router forwards the VXLAN packets, from NS-ADC, to all the VTEPs that are part of the multicast group.
- **Create a VXLAN entity**. Create a VXLAN entity uniquely identified by a positive integer, which is also called the VXLAN Network Identifier (VNI). In this step, you can also specify the destination UDP port of remote VTEP on which the VXLAN protocol is running. By default, the destination UDP port parameter is set to 4789 for the VXLAN entity. This UDP port setting must match the settings on all remote VTEPs for this VXLAN. In this step, you can also bind VLANs to this VXLAN. The traffic (which includes broadcasts, multicasts, unknown unicasts) of all bound VLANs are allowed over this VXLAN. If no VLANs are bound to the VXLAN, the NetScaler ADC allows traffic of all VLANs, on this VXLAN, that are not part of any other VXLANs.
- **Bind the VXLAN tunnel to the VXLAN entity**. Bind the desired IP tunnels, of type VXLAN, to the VXLAN entity. More than one VXLAN tunnels can be bound to the VXLAN entity. These VXLAN tunnels form the broadcast domain for the VXLAN identified by its VNI.
- **(Optional) Bind different feature entities to the configured VXLAN**. VXLANs function similarly to VLANs, most of the NetScaler ADC features that support VLAN as a classification parameter also support VXLAN. These features include an optional VXLAN parameter setting, which specifies the VXLAN VNI.
- **(Optional) Display the VXLAN mapping table**. Display the VXLAN mapping table, which includes mapping entries for remote host MAC address to VTEP IP address for a particular VXLAN. In other words, a VXLAN mapping states that a host is reachable through the VTEP on a particular VXLAN. The NetScaler ADC learns VXLAN mappings and updates its mapping table from the VXLAN packets it receives. The NetScaler ADC uses the VXLAN mapping table to lookup for the destination MAC address of a Layer 2 frame. If an entry for this MAC address is present in the VXLAN table, the NetScaler ADC sends the Layer 2 frame over Layer 3, using the VXLAN protocol, to the mapped VTEP IP address specified in the mapping entry for a VXLAN.

Configuration Using the Command Line Interface

To create a VXLAN tunnel by using the command line interface

At the command prompt, type:

- add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol VXLAN [-vlan <positive_integer>]
- show ipTunnel <name>

To create a VXLAN entity by using the command line interface

At the command prompt, type:

- add vxlan <id> [-vlan <positive_integer>] [-port <port>]
- show vxlan <id>

To bind a VXLAN tunnel to a VXLAN entity by using the command line interface

At the command prompt, type:

- bind vxlan <id> -tunnel <string>
- show vxlan <id>

To display the VXLAN forwarding table by using the command line interface

At the command prompt, type:

show bridgetable

Example

```
> add ipTunnel VXLAN-9000-Tunnel-1 203.0.101.101 255.255.255.255 203.0.100.100 -protocol VXLAN
 Done

> add ipTunnel VXLAN-9000-Tunnel-2 203.0.102.102 255.255.255.255 203.0.100.100 -protocol VXLAN
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> add vxlan 9000
Done

> bind vxlan 9000 -tunnel VXLAN-9000-Tunnel-1
Done

> bind vxlan 9000 -tunnel VXLAN-9000-Tunnel-2
Done
```

Configuration Using the Configuration Utility

To create a VXLAN tunnel by using the configuration utility

Navigate to System > Network > IP Tunnels, and add a tunnel of type VXLAN.

To create a VXLAN entity and bind a VXLAN tunnel by using the configuration utility

1. Navigate to System > Network > VXLANs, and add a new VXLAN entity.
2. Open a VXLAN entity, in the VXLAN IP Tunnel Bindings pane, bind a VXLAN tunnel to the VXLAN entity.

To display the VXLAN forwarding table by using the configuration utility

Navigate to System > Network > Bridge Table.

Support of IPv6 Dynamic Routing Protocols on VXLANs

The NetScaler appliance supports IPv6 dynamic routing protocols for VXLANs. You can configure various IPv6 Dynamic Routing protocols (for example, OSPFv3, RIPng, BGP) on VXLANs from the VTYSH command line. An option IPv6 Dynamic Routing Protocol has been added to VXLAN command set for enabling or disabling IPv6 dynamic routing protocols on a VXLAN. After enabling IPv6 dynamic routing protocols on a VXLAN, processes related to the IPv6 dynamic routing protocols are required to be started on the VXLAN by using the VTYSH command line.
To enable IPv6 Dynamic routing protocols on a VXLAN by using the NetScaler command line

- add vxlan <ID> [-ipv6DynamicRouting (ENABLED | DISABLED)]
- show vxlan

Example

In the following sample configuration, VXLAN-9000 is created and has IPv6 dynamic routing protocols enabled on it. Then, using the VTYSH command line, process for the IPv6 OSPF protocol is start > add ipTunnel VXLAN-9000-Tunnel-1 203.0.101.101 255.255.255.255 203.0.100.100 -protocol VXLAN Done > add vxlan 9000 -ipv6DynamicRouting ENABLED Done > bind vxlan 9000 -tunnel VXLA

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppExpert

Sep 14, 2016

The following topics provide a conceptual reference and configuration instructions for the AppExpert and other features of the NetScaler appliance.

Action Analytics	Collects run-time statistics on the basis of pre-defined criteria. When used with policies, the feature also provides you with the infrastructure for automatic, real-time traffic optimization.
AppExpert Applications and Templates	Simplify configuration steps for the Citrix® NetScaler® appliance by using applications, application templates, NetScaler Gateway applications, and entity templates.
Entity Templates	Describes how to use entity templates to set up and configure individual NetScaler entities, such as a policy or virtual server. An entity template provides a specification and a set of defaults for the object.
AppQoE	Application level Quality of Experience (AppQoE) integrates several existing policy-based security features of the NetScaler appliance into a single integrated feature that takes advantage of a new queuing mechanism, fair queuing.
HTTP Callouts	An HTTP request that the NetScaler appliance generates and sends to an external application when certain criteria are met during policy evaluation.
Pattern Sets	Allow string matching during the evaluation of a default syntax policy.
Policies and Expressions	Rules that determine the operations that the NetScaler appliance must perform.
Rate Limiting	Defines the maximum load for a given network entity or virtual entity on the NetScaler appliance.
Responder	Bases responses on who sends the request, where it is sent from, and other criteria with security and system management implications.
Rewrite	Rewrites information in the requests or responses handled by the NetScaler appliance.
String Maps	Perform pattern matching in all NetScaler features that use the default policy syntax.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Action Analytics

Sep 16, 2013

The performance of your website or application depends on how well you optimize the delivery of the most frequently requested content. Techniques such as caching and compression help accelerate the delivery of services to clients, but you need to be able to identify the resources that are requested most frequently, and then cache or compress those resources. You can identify the most frequently used resources by aggregating real-time statistics about website or application traffic. Statistics such as how frequently a resource is accessed relative to other resources and how much bandwidth is consumed by those resources help you determine whether those resources need to be cached or compressed to improve server performance and network utilization. Statistics such as response times and the number of concurrent connections to the application help you determine whether you must enhance server-side resources.

If the website or application does not change frequently, you can use products that collect statistical data, and then manually analyze the statistics and optimize the delivery of content. However, if you do not want to perform manual optimizations, or if your website or application is dynamic in nature, you need infrastructure that can not only collect statistical data but can also automatically optimize the delivery of resources on the basis of the statistics. On the NetScaler appliance, this functionality is provided by the action analytics feature. The feature operates on a single NetScaler appliance and collects run-time statistics on the basis of criteria that you define. When used with NetScaler policies, the feature also provides you with the infrastructure that you need for automatic, real-time traffic optimization.

When configuring the action analytics feature, you specify the request attributes for which you want to collect statistical data (for example, URLs and HTTP methods) by configuring default syntax expressions in an entity called a selector. Then, you configure an identifier to configure settings such as the sampling interval and sample count. You also configure a policy that enables the appliance to evaluate traffic as specified by the selector-identifier pair. Finally, you bind the policy to a bind point to begin collecting statistics.

The appliance also provides you with a set of built-in selectors, identifiers, and responder policies that you can use to get started with the feature.

The appliance aggregates the following statistics:

- The number of requests.
- The bandwidth consumed by the requests.
- The response time.
- The number of concurrent connections.

You can configure the feature to perform run-time sorting of the records on an attribute of your choice. You can view the statistical data by using either the command-line interface or the Stream Sessions tool in the configuration utility.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Selector

Sep 16, 2013

A selector is a filter for identifying requests. It consists of up to five individual default syntax expressions that identify request attributes such as the client IP address and the URL in the request. Each expression is a non-compound default syntax expression and is considered to be in an AND relationship with the other expressions. Following are some examples of selector expressions:

- HTTP.REQ.URL
- CLIENT.IP.SRC
- HTTP.RES.BODY(1000).AFTER_STR(\"<string>\").BEFORE_STR(\"<string>\")"
- CLIENT.IP.SRC.SUBNET(24)

Selectors are used in rate limiting and action analytics configurations. A selector is optional in a rate limiting configuration, but is required in a action analytics configuration.

The order in which you specify parameters is significant. For example, if you configure an IP address and a domain (in that order) in one selector, and then specify the domain and the IP address (in the reverse order) in another selector, the NetScaler considers these values to be unique. This can lead to the same transaction being counted twice. Also, if multiple policies invoke the same selector, the NetScaler, again, can count the same transaction more than once.

If you modify an expression in a selector, you may get an error if any policy that invokes it is bound to a new policy label or bind point. For example, suppose that you create a selector named myLimitSelector1, invoke it from myLimitID1, and invoke the identifier from a DNS policy named dnsRateLimit1. If you change the expression in myLimitSelector1, you might receive an error when binding dnsRateLimit1 to a new bind point. The workaround is to modify these expressions before creating the policies that invoke them.

The NetScaler appliance provides the following built-in selectors for some of the most common use cases:

Table 1. Built-in Selectors

Selector	Selector Expressions
Top_URL	HTTP.REQ.URL
Top_CLIENTS	CLIENT.IP.SRC
Top_URL_CLIENTS_LBVSERVER	1. HTTP.REQ.URL 2. CLIENT.IP.SRC 3. HTTP.REQ.LB_VSERVER.NAME
Top_URL_CLIENTS_CSVSERVER	1. HTTP.REQ.URL 2. CLIENT.IP.SRC 3. HTTP.REQ.CS_VSERVER.NAME
Top_MSSQL_QUERY_DB_LBVSERVER	1. MSSQL.REQ.QUERY.TEXT 2. MSSQL.REQ.LB_VSERVER.NAME

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Selector	Selector Expressions
Top_MYSQL_QUERY_DB_LBVSERVER	1. MYSQL.REQ.QUERY.TEXT 2. MYSQL.REQ.LB_VSERVER.NAME

You can also configure a selector with expressions that identify the request attributes of your choice. For example, you might want to create a record for a request that arrives with a specific header. To evaluate the header, you can add HTTP.REQ.HEADER("<header_name>") to the selector that you intend to use.

To configure a selector by using the command line interface

At the command prompt, type the following commands to configure a selector and verify the configuration:

- add stream selector <name> <rule> ...
- show stream selector

Example

```
> add stream selector myselector HTTP.REQ.URL CLIENT.IP.SRC
 Done
> show stream selector myselector
 Name: myselector
 Expressions:
  1) HTTP.REQ.URL
  2) CLIENT.IP.SRC
 Done
>
```

To modify or remove a selector by using the command line interface

- To modify a selector, type the set stream selector command, the name of the selector, and the rule parameter with the expressions. Enter the existing expressions that you want to retain, along with the new expressions that you want to add.
- To remove a selector, type the rm stream selector command and the name of the selector.

To configure a selector by using the configuration utility

1. Navigate to AppExpert > Action Analytics > Selectors.
2. In the details pane, do one of the following:
   - To create a selector, click Add.
   - To modify a selector, select the selector, and then click Open.
3. In the Create Limit Selector or Configure Limit Selector dialog box, set one or more of the following parameters:
   - Name
   - Expressions
     To add the expression to the selector configuration, click Add. To remove an expression from the selector configuration, in the Expression box, select the expression, and then click Remove.
     Note: In the Expressions box, enter a valid parameter. For example, enter HTTP. Then, enter a period after this parameter. A drop-down menu appears. The contents of this menu provide the keywords that can follow the initial keyword that you entered. To select the next keyword in this expression prefix, double-click the selection in the drop-down menu. The Expressions text box displays both the first and second keywords for the expression prefix, for

© 1999-2017 Citrix Systems, Inc. All rights reserved.

example, HTTP.REQ. Continue adding expression components until the complete expression is formed.

4. Click Add.
5. Continue adding up to five non-compound expressions.
6. Click Create or OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Stream Identifier

Aug 30, 2013

You configure a stream identifier to specify parameters for collecting statistical data from requests identified by a given selector. An identifier specifies the selector to be used, the statistics collection interval, the sample count, and the field on which the records are to be sorted.

The NetScaler appliance includes the following built-in stream identifiers for common use cases. All the built-in identifiers specify a sample count of 1 and an interval of 1 minute. Additionally, they sort the data on the REQUESTS attribute. They differ only in being associated with different built-in selectors. Each built-in identifier is associated with a built-in selector of the same name (for example, the built in identifier Top_URL is associated with the built-in selector Top_URL). Following are the built-in identifiers:

- Top_URL
- Top_CLIENTS
- Top_URL_CLIENTS_LBVSERVER
- Top_URL_CLIENTS_CSVSERVER
- Top_MSSQL_QUERY_DB_LBVSERVER
- Top_MYSQL_QUERY_DB_LBVSERVER

For more information about the built-in selectors, see "Configuring a Selector."

Note: The maximum length for storing string results of selectors (for example, HTTP.REQ.URL) is 60 characters. If the string (for example, URL) is 1000 characters long, of which 50 characters are enough to uniquely identify a string, use an expression to extract only the required 50 characters.
You cannot modify a built-in identifier's configuration. However, you can create an identifier with a configuration of your choice.

To configure a stream identifier by using the command line interface

At the command prompt, type the following commands to configure a stream identifier and verify the configuration:

- add stream identifier <name> <selectorName> [-interval <positive_integer>] [-SampleCount <positive_integer>] [-sort <sort>]
- show stream identifier <name>

Example

```
> add stream identifier myidentifier Top_URL -interval 10 -sampleCount 100
 Done
```

To configure a stream identifier by using the configuration utility

1. Navigate to AppExpert > Action Analytics > Stream Identifiers.
2. In the details pane, do one of the following:
   - To create a stream identifier, click Add.
   - To modify a stream identifier, select the identifier, and then click Open.
3. In the Configure Stream Identifier dialog box, set one or more of the following parameters:
   - Name
   - Selector
   - Interval

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Sample Count
- Sort

4. Click Create or OK, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing Statistics

Aug 30, 2013

You can view the collected statistics in tabular format in the command-line interface and in graphical format in the configuration utility.

The following table describes the collected statistics:

Table 1. Statistical Information Displayed for a Stream Identifier

Statistics	Column name in the output of the stat stream identifier <identifier name> command	Description
Number of requests	Req	The number of requests for which records were created in the last <interval> number of minutes.
Bandwidth consumed	BandW	The total bandwidth consumed by the requests that were received in the last <interval> number of minutes. The total bandwidth of a request is the bandwidth consumed by the request and its response. The value is rounded off to the next higher or next lower integer value. Consequently, it might differ slightly from the expected value. For example, if a request's total bandwidth consumption is 2.2 KB, one instance of the request might be shown as having consumed 2 KB and two instances might be shown as having consumed 4 KB, but three instances might be shown as having consumed 7 KB.
Response time	RspTime	The average response time for all the requests received in the last <interval> number of minutes.
Concurrent connections	Conn	The total number of concurrent connections that are currently open.

To view the statistical data collected for a stream identifier by using the command line

At the command prompt, type:

stat stream identifier <name> [<pattern> ...] [-detail] [-fullValues] [-ntimes <positive_integer>] [-logFile <input_filename>] [-sortBy <sortBy> [<sortOrder>]

Examples

Example 1 sorts the output on the BandW column, in the descending order. Example 2 sorts the output in Example 1, on the Req column, and in the ascending order

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 1

```
> stat stream identifier myidentifier -sortBy BandW Descending -fullValues
Stream Session statistics
            Req          BandW
User1       508          125924
User2       5020         12692
User3       2025         4316

          RspTime        Conn
User1     5694           0
User2     109            0
User3     3              0
 Done
```

Example 2

```
> stat stream identifier myidentifier -sortBy Req Ascending –fullValues
Stream Session statistics
            Req          BandW
User1       508          125924
User3       2025         4316
User2       5020         12692

          RspTime        Conn
User1     5694           0
User3     3              0
User2     109            0
 Done
```

To view the statistical data collected for a stream identifier by using the configuration utility

1. Navigate to AppExpert > Action Analytics > Stream Identifiers.
2. Select the stream identifier whose sessions you want to view, and then click Stream Sessions. For information about how you can group the output on the basis of the values collected for various selector expressions, see "Grouping Records on Attribute Values."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Grouping Records on Attribute Values

Sep 30, 2013

Statistical information such as the number of times a particular URL has been accessed overall and per client, and the total number of GET and POST requests per client can provide valuable insights into whether any of your resources need to be expanded to meet the demand or be optimized for delivery. To obtain such statistics, you must use an appropriate set of selector expressions, and then use the pattern parameter in the stat stream identifier command. The grouping is based on the pattern that is specified in the command. Grouping can be performed concurrently on the values of multiple expressions.

In the command-line interface, you can group the output by using patterns of your choice. In the configuration utility, the pattern depends on the choices you make when drilling down through the values of various selector expressions. For example, consider a selector that has the expressions HTTP.REQ.URL, CLIENT.IP.SRC, and HTTP.REQ.LB_VSERVER.NAME, in that order. The statistics home page displays icons for each of these expressions. If you click the icon for CLIENT.IP.SRC, the output is based on the patterns * ? *. The output displays statistics for each client IP address. If you click an IP address, the output is based on the patterns * <IP address> ? and ? <IP address> * where <IP address> is the IP address you selected. In the resulting output, if you click a URL, the pattern used is <URL> <IP address> ?.

To group the records on the values of selector expressions by using the command line interface

At the command prompt, enter the following command to group the records on the basis of a selector expression:

stat stream identifier <name> [<pattern> ...]

Examples

Each example uses a different pattern to demonstrate the effect of the pattern on the output of the stat stream identifier command. The selector expressions are HTTP.REQ.URL and HTTP.REQ.HEADER("UserHeader"), in that order. The requests contain a custom header whose name is UserHeader. Note that in the examples, a given statistical value changes as determined by the grouping, but the sum total of the values for a given field remains the same.

Example 1

In the following command, the pattern used is ? ?. The appliance groups the output on the values collected for both selector expressions. The row headers consist of the expression values separated by a question mark (?). The row with the header /mysite/mypage1.html?Ed displays statistics for requests made by user Ed for the URL /mysite/mypage1.html.

```
> stat stream identifier myidentifier ? ? -fullValues
Stream Session statistics
                                Req          BandW
/mysite/mypage2.html?Grace        1          2553
/mysite/mypage1.html?Grace        2             4
/mysite/mypage1.html?Ed           8            16
/mysite/mypage2.html?Joe          1          2554
/mysite/mypage1.html?Joe          5            10
/mysite/?Joe                  1          4

                            RspTime        Conn
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
/mysite/mypage2.html?Grace          0          0
/mysite/mypage1.html?Grace          0          0
/mysite/mypage1.html?Ed             0          0
/mysite/mypage2.html?Joe            0          0
/mysite/mypage1.html?Joe            0          0
/mysite/?Joe                        6          0
 Done
```

Example 2

In the following command, the pattern used is * ?. The appliance groups the output on the values accumulated for the second expression HTTP.REQ.HEADER("UserHeader"). The rows display statistics for all requests made by users Grace, Ed, and Joe.

```
> stat stream identifier myidentifier * ?
Stream Session statistics
            Req   BandW  RspTime    Conn
Grace         3    2557       0       0
Ed            8      16       0       0
Joe           7    2568       6       0
 Done
```

Example 3

In the following command, the pattern used is ? *, which is the default pattern. The output is grouped on the values collected for the first selector expression. Each row displays statistics for one URL.

```
> stat stream identifier myidentifier ? * -fullValues
Stream Session statistics
                          Req          BandW
/mysite/mypage2.html        2           5107
/mysite/mypage1.html       15             30
/mysite/                    1              4

                       RspTime           Conn
/mysite/mypage2.html        0              0
/mysite/mypage1.html        0              0
/mysite/                    6              0
 Done
```

Example 4

In the following command, the pattern used is * *. The appliance displays one set of collective statistics for all the requests received, with no row title.

```
> stat stream identifier myidentifier * *
Stream Session statistics
            Req   BandW  RspTime    Conn
             18    5141       6       0
 Done
```

Example 5

In the following command, the pattern is /mysite/mypage1.html *. The appliance displays one set of collective statistics for

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

all the requests received for the URL /mysite/mypage1.html, with no row title.

```
> stat stream identifier myidentifier /mysite/mypage1.html *
Stream Session statistics
        Req   BandW  RspTime    Conn
         15     30       0       0
 Done
```

To group the records on the values of selector expressions by using the configuration utility

1. Navigate to AppExpert > Action Analytics > Stream Identifiers.
2. In the details pane, click the stream identifier for which you want to view statistics, and then click Stream Sessions.
3. On the Home page, click the icon for the stream selector by which you want to group the output.
4. To return to the Home page from the statistics page for a selector expression, click Home.
5. To view statistics for the value of a given selector expression, click the value. You can repeat this step for a selector expression value in each subsequent output until you obtain the statistics you want.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Clearing a Stream Session

Aug 30, 2013

You can flush all the records that have been accumulated for a stream identifier.

To clear a stream session by using the command line interface

At the command prompt, enter the following commands to clear a stream session and verify the results:

- clear stream session <name>
- stat stream identifier <name>

Example

This example uses the stat stream identifier command first, so that a comparison can be made with the stat stream identifier command that is used for verifying the result of the clear stream session command.

```
>stat stream identifier myidentifier
Stream Session statistics
            Req   BandW  RspTime    Conn
/aed....html    2      0       0       0
/             636    303      12       0
 Done
>clear stream session myidentifier
 Done
>stat stream identifier myidentifier
 Done
```

To clear a stream session by using the configuration utility

1. Navigate to AppExpert > Action Analytics > Stream Identifiers.
2. Select the stream identifier whose sessions you want to clear, and then click Clear Sessions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Policy for Analyzing and Optimizing Traffic

Sep 16, 2013

To put the selector-identifier pair in your action analytics configuration into effect, you must associate the pair with the point in the traffic flow at which you want to collect statistics. You can do so by configuring a default syntax policy and referencing the stream identifier from the policy rule. You can use compression policies, caching policies, rewrite policies, application firewall policies, responder policies, and any other policies whose action is based on a Boolean expression.

The action analytics feature introduces a set of default syntax expressions and functions for collecting and evaluating data. The expression ANALYTICS.STREAM(<identifier_name>) is used for referencing the identifier that you want to use. The expression COLLECT_STATS is used to collect statistical data. Functions such as IS_TOP(<uint>) and IS_TOP_FREQUENTS(<uint>) are used for making automatic, real-time traffic optimization decisions.

- **IS_TOP(<number>).** Finds if a given object is in the top <number> of elements. For example, is the element among the top 10 elements. When multiple elements have the count, they are considered to be similar in nature. The sort function must be turned on to avoid an undef condition.
- **IS_TOP_FREQUENTS(<frequency>).** Finds if a given object is in the top <frequency> of the elements that are in the top elements. For example, is the element among the top 50% of all the top elements maintained. Elements having the same values are considered similar in nature. The sort function must be turned on to avoid an undef condition.

It is your policy configuration that determines whether the NetScaler appliance must only collect data from traffic or also perform an action. If the appliance must only collect statistical data, you can configure a policy with the rule ANALYTICS.STREAM(<identifier_name>).COLLECT_STATS and the action NOOP. The NOOP policy must be the policy with the highest priority at the bind point. This policy is sufficient if you are only collecting statistics. Traffic optimization decisions, such as what to compress or cache, must be based on manual, periodic evaluation of the statistical data.

If, in addition to collecting statistics, the appliance must also perform an action on the traffic, you must configure the gotoPriorityExpression parameter of the NOOP policy such that another policy that has the desired rule and action is evaluated subsequently. This second policy must have a rule that begins with the ANALYTICS.STREAM(<identifier_name>) prefix and a function that evaluates the data.

Following is an example of two responder policies that are configured and bound globally. The policy responder_stat_collection enables the appliance to collect statistics based on the identifier, myidentifier. The policy responder_notify evaluates the data that is collected.

Example

```
> add responder action send_notification respondwith '"You are in the Top 10 list for bandwidth consumption"'
 Done
> add responder policy responder_stat_collection' ANALYTICS.STREAM("myidentifier").COLLECT_STATS' NOOP
Done
> add responder policy responder_notify 'ANALYTICS.STREAM("myidentifier").BANDWIDTH.IS_TOP(10)' send_notification
 Done
> bind responder global responder_stat_collection 10 NEXT
 Done
> bind responder global responder_notify 20 END
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case: Limiting Bandwidth Consumption per User or Client Device

Sep 16, 2013

Your web site, application, or file hosting service has finite network and server resources available to it to serve all its users. One of the most important resources is bandwidth. Substantial bandwidth consumption by only a subset of the user base can result in network congestion and reduced resource availability to other users. To prevent network congestion, you might have to limit a client's bandwidth consumption by using temporary service denial techniques such as responding to a client request with an HTML page if it has exceeded a preconfigured bandwidth value over a fixed time period leading up to the request.

In general, you can regulate bandwidth consumption either per client device or per user. This use case demonstrates how you can limit bandwidth consumption per client to 100 MB over a time period of one hour. The use case also demonstrates how you can regulate bandwidth consumption per user to 100 MB over a time period of one hour, by using a custom header that provides the user name. In both cases, the tracking of bandwidth consumption over a moving time period of one hour is achieved by setting the interval parameter in the stream identifier to 60 minutes. The use cases also demonstrate how you can import an HTML page to send to a client that has exceeded the limit. Importing an HTML page not only simplifies the configuration of the responder action in these use cases, but also simplifies the configuration of all responder actions that need the same response.

To limit bandwidth consumption per user or client device by using the command line interface

In the command-line interface, perform the following tasks to configure action analytics for limiting a client's or user's bandwidth consumption. Each step includes sample commands and their output.

1. **Set up your load balancing configuration.** Configure load balancing virtual server mysitevip, and then configure all the services that you need. Bind the services to the virtual server. The following example creates ten services and binds the services to mysitevip.

```
> add lb vserver mysitevip HTTP 192.0.2.17 80
 Done
> add service service[1-10] 192.0.2.[240-249] HTTP 80
service "service1" added
service "service2" added
service "service3" added
.
.
.
service "service10" added
 Done
> bind lb vserver vserver1 service[1-10]
service "service1" bound
service "service2" bound
service "service3" bound
.
.
.
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

service "service10" bound
Done

2. **Configure the stream selector.** Configure one of the following stream selectors:
   - To limit bandwidth consumption per client, configure a stream selector that identifies the client IP address.
     > add stream selector myselector CLIENT.IP.SRC
     Done
   - To limit bandwidth consumption per user on the basis of the value of a request header that provides the user name, configure a stream selector that identifies the header. In the following example, the name of the header is UserHeader.
     > add stream selector myselector HTTP.REQ.HEADER(“UserHeader”)
     Done
3. **Configure a stream identifier.** Configure a stream identifier that uses the stream selector. Set the interval parameter to 60 minutes.
   > add stream identifier myidentifier myselector -interval 60 -sampleCount 1 -sort BANDWIDTH
   Done
4. **Configure the responder action.** Import the HTML page that you want to send to users or clients that have exceeded the bandwidth consumption limit, and then use the page in responder action crossed_limits.
   > import responder htmlpage http://192.0.2.20:80/stdpages/wait.html crossed-limits.html
   This operation may take some time, Please wait...

   Done
   > add responder action crossed_limits respondwithhtmlpage crossed-limits.html
   Done
5. **Configure the responder policies.** Configure responder policy myrespol1 with the rule ANALYTICS.STREAM("myidentifier").COLLECT_STATS and the action NOOP. Then, configure policy myrespol2 for determining whether a client or user has crossed the 100 MB limit. The policy myrespol2 is configured with the responder action crossed_limits.
   > add responder policy myrespol1 'ANALYTICS.STREAM("myidentifier").COLLECT_STATS' NOOP
   Done
   > add responder policy myrespol2 'ANALYTICS.STREAM("myidentifier").BANDWIDTH.GT(1048576)' crossed_limits
   Done
6. **Bind the responder policies to the load balancing virtual server.** The policy myrespol1, which only collects statistical data, must have the higher priority and a GOTO expression of NEXT.
   > bind lb vserver mysitevip -policyName myrespol1 -priority 1 -gotoPriorityExpression NEXT
   Done
   > bind lb vserver mysitevip -policyName myrespol2 -priority 2 -gotoPriorityExpression END
   Done
7. **Test the configuration.** Test the configuration by sending test HTTP requests, from multiple clients or users, to the load balancing virtual server and using the stat stream identifier command to view the statistics that are collected for the specified identifier. The following output displays statistics for clients.
   > stat stream identifier myidentifier -sortBy BandW –fullValues
   Stream Session statistics

	Req	BandW
192.0.2.30	5000	3761
192.0.2.31	29	2602
192.0.2.32	25	51

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
                RspTime    Conn
192.0.2.30        2         0
192.0.2.31        0         0
192.0.2.32        0         0
 Done
>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppExpert Applications and Templates

Jan 04, 2016

An AppExpert application is a collection of configuration information that you set up on the Citrix NetScaler appliance for securing and optimizing traffic for a Web application, such as Microsoft SharePoint. Managing AppExpert applications is simplified by a graphical user interface (GUI) that allows you to specify application traffic subsets and a distinct set of security and optimization policies for processing each traffic subset. Additionally, it consolidates all deployment tasks in one view, so you can quickly configure target IP addresses for clients and specify host servers.

Prebuilt application templates for widely used Web applications, such as Microsoft Outlook Web Access and Microsoft SharePoint, are available on the AppExpert Templates page of the Citrix Community website at "http://community.citrix.com/display/ns/AppExpert+Templates."

Each prebuilt template provides you with an initial configuration for managing the associated Web application. You can customize prebuilt application templates for your organization. If a prebuilt application template does not suit your requirements, you can create a custom application without using a template.

Regardless of whether you use a prebuilt application template or you create a custom application, you can export the configuration to a template file. You can then share the template with other administrators or import the template to other NetScaler appliances that require a similar AppExpert application configuration.

To get started with an AppExpert application, you must first obtain the appropriate application template and import the template to the NetScaler appliance. After the AppExpert application is set up, you must verify that the application is working correctly. If required, you can customize the configuration to suit your requirements.

Periodically, you can verify and monitor the configuration by viewing the hit counters for various application components, statistics, and the Application Visualizer. You can also configure authentication, authorization, and auditing (AAA) policies for the application.

AppExpert Application Terminology

Updated: 2013-08-30

Following are the terms used in the AppExpert applications feature and the descriptions of the entities for which the terms are used:

Public Endpoint. The IP address and port combination at which the NetScaler appliance receives client requests for the associated web application. A public endpoint can be configured to receive either HTTP or secure HTTP (HTTPS) traffic. All client requests for the web application must be sent to a public endpoint. An AppExpert application can be assigned multiple endpoints. You configure public endpoints after you import a template.

Application Unit. An AppExpert application entity that processes a subset of web application traffic and load balances a set of services that host the associated content. The subset of traffic that an application unit must manage is defined by a rule. Each application unit also defines its own set of traffic optimization and security policies for the requests and responses that it manages. The NetScaler services associated with these policies are Compression, Caching, Rewrite, Responder, and application firewall.

By default, every AppExpert application with at least one application unit includes a default application unit, which cannot be deleted. The default application unit is not associated with a rule for identifying requests and is always placed last in the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

order of application units. It defines a set of policies for processing any request that does not match the rules that are configured for the other application units, thereby ensuring that all client requests are processed.

Application units and their associated rules, policies, and actions are included in AppExpert application templates.

Service. The combination of the IP address of the server that hosts the web application instance and the port to which the application is mapped on the server, in the format <IP address>:<Port>. A web application that serves a large number of requests is usually hosted on multiple servers. Each server is said to host an instance of the web application, and each such instance of the web application is represented by a service on the NetScaler appliance. Services are deployment-specific, and are therefore not included in templates. You must configure services after you import a template.

Application Unit Rule. Either a classic expression or a default syntax expression that defines the characteristics of a traffic subset for an application unit. The following example rule is a default syntax expression that identifies a traffic subset that consists of four image types:

```
HTTP.REQ.URL.SUFFIX.EQ("bmp") || HTTP.REQ.URL.SUFFIX.EQ("gif") || HTTP.REQ.URL.SUFFIX.EQ("png") ||
HTTP.REQ.URL.SUFFIX.EQ("jpg")
```

For more information about default syntax expressions and classic policy expressions, see "Policies and Expressions."

Traffic Subset. A set of client requests that require a common set of traffic optimization and security policies. A traffic subset is managed by an application unit and is defined by a rule.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How AppExpert Application Works

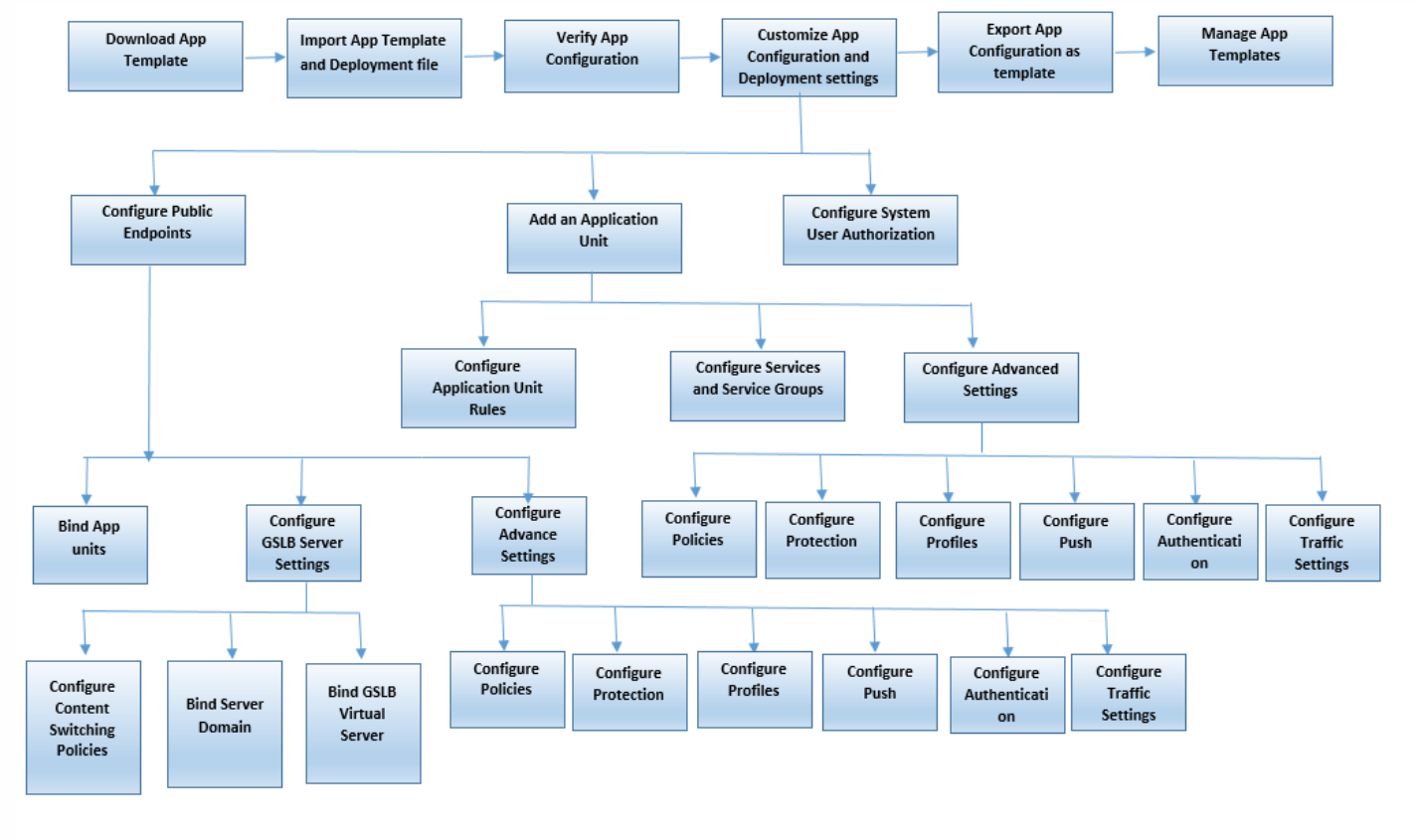
Jan 05, 2016

When the endpoint receives a client request, the NetScaler appliance evaluates the request against the rule that is configured for the topmost application unit. If the request satisfies this rule, the request is processed by the policies that are configured for the application unit, and then forwarded to a service. The choice of service depends on which services are configured for the application, and on settings such as the load balancing algorithm and persistence method configured for the application unit.

If the request does not satisfy the rule, the request is evaluated against the rule for the next topmost application unit. In this order, the request is evaluated against each application unit rule until the request satisfies a rule. If the request does not satisfy any of the configured rules, it is processed by the default application unit, which is always the last application unit.

You can configure multiple public endpoints for an AppExpert application. In such a configuration, by default, each application unit processes requests received by all the public endpoints and load balances all the services that are configured for the application. However, you can specify that an application unit processes traffic from only a subset of the public endpoints and load balances only a subset of the services that are configured for the AppExpert application.

The following flow diagram illustrates the AppExpert Application flow sequence for using a built-in application template.



If you prefer to create a customized application without using a template, do the following:

1. Create a custom application.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Configure application and deployment settings.
3. Export the configuration to new template files (optional).
4. Import the template files to other NetScaler appliances that require a similar AppExpert application configuration

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Getting Started with an AppExpert Application

Feb 05, 2016

To get started with an AppExpert application, you must first obtain an application template and import the template to a NetScaler appliance. After the AppExpert application is set up, you must verify that the application is working correctly. If required, you can customize the configuration to suit your requirements.

Periodically, you can verify and monitor the configuration by viewing the hit counters for various application components. You can also configure authentication, authorization, and auditing (AAA) policies for the application.

The process of setting up an application can be done in two ways:

1. Using a prebuilt application template
2. Creating a custom application without using a template.

If you prefer to set up the application by using a prebuilt application template, do the following:

1. Download an application template.
2. Import template files to Netscaler appliance.
3. Verify application setup.
4. Configure application and deployment settings.
5. Export the configuration to new template files (optional).
6. Import the template files to other NetScaler appliances that require a similar AppExpert application configuration.

Citrix NetScaler's video tutorials enable you to understand NetScaler features in easy and simple way. Watch https://www.youtube.com/watch?v=aqayflvCR_0 video to learn how to set up an application using AppExpert Application template.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Downloading an Application Template

Jan 07, 2016

AppExpert application setup begins with downloading an application template from the Citrix Community Web site at http://community.citrix.com/display/ns/AppExpert+Templates to your local computer or NetScaler appliance. The application templates are designed to be imported and exported, so that you can easily share application-specific configurations within an organization or across organizations. An application template includes the following set of entities:

1. Application components (for example, web pages, files, archives, and web services)
2. Traffic management entities (for example, virtual server IP addresses and associated load-balancing algorithms, and SSL offload settings) for the application components.
3. Netscaler policies used for optimizing the application traffic.

Note: Application templates are available in different versions for configuring different types of NetScaler appliances.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Importing an Application Template

Feb 05, 2016

For NetScaler software version 9.3 or later, each AppExpert template has two XML files: a Template file and a Deployment file. You must import both files from your local computer to a Netscaler appliance. You can either import the template files from your computer to the AppExpert application templates directory in Netscaler appliance or upload files to a NetScaler appliance and then import them from the appliance.

Note: When you import a template from an appliance, you have to provide the variable value available in the template. By default, the pre-configured value is displayed.

After you import the template files, the application-configuration and deployment information populates the target application automatically. The appliance imports all the configuration from the template files through the NITRO API. If you do not import the deployment file, the system generates an application populated with content switching virtual server configuration. For more information about the format of application templates and deployment files, see "Understanding Netscaler Application Templates and Deployment Files" section.

When you import a template, if you do not include a deployment file, you have to configure the public endpoints in the application that the system automatically generates from the template. One endpoint for HTTP and another endpoint for HTTPS. When configuring a public endpoint of type HTTPS, make sure you enable the SSL feature, bind the server certificate, and include the server-certificate and certificate-key files.

For more information about configuring endpoints after you import a template, see "Configuring Public Endpoints" topic.

To import AppExpert application template files to a NetScaler appliance by using the configuration utility:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click **Import Template**.
3. On the **Import** page, set the following parameters:
   1. Application Name (mandatory)
   2. Template File (mandatory)
   3. Use deployment file
4. Click **Continue** to auto populate application-configuration and deployment information into an application.

Citrix NetScaler's video tutorials enable you to understand NetScaler features in an easy and simply way. Watch https://www.youtube.com/watch?v=AR9TwSD9uJM video to learn how to import an application template.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Verifying and Testing Application Configuration

Jan 07, 2016

Verifying the Configuration

The NetScaler configuration utility includes icons that indicate the states of the entities in the AppExpert application. These icons are displayed for applications and application units and are based on the health checks that the NetScaler appliance performs periodically on services and entities. The following table lists the icons and describes their meanings.

Table 1. Descriptions of State Indicator Icons

Icon	Entity	Indicates that
	Application	At least one public endpoint is up. The application will accept client requests from the public endpoints that are up.
	Application unit	The application unit is up. The application unit is up when at least one service or service group is up.
	Application	The public endpoint is out of service (disabled). This indicator is displayed when only one public endpoint is configured for the AppExpert application.
	Application	All the endpoints that are configured for the application are out of service. This indicator is displayed only when multiple endpoints are configured for the application.
	Application unit	All the services configured for the application unit are down.

You must ensure that the icons for each application and its application units are green at all times. If the icon that is displayed for an application is not green, verify that you have configured the public endpoints correctly. If the icon that is displayed for an application unit is not green, verify that the services are configured correctly. However, note that a green indicator does not mean that the state of all associated entities is UP. It only means that the application has sufficient resources (endpoints and services) to serve client requests. To verify that the state of all associated entities is UP, check the health of all the entities on the statistics page for the application.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Customizing the Configuration

Dec 24, 2015

After you verify that the AppExpert application is working correctly, you can customize the configuration to suit your requirements.

After you verify that the AppExpert application configuration is working correctly, you can configure the application and the deployment settings to suit your requirements. When you import an application template and deployment file, the system automatically populates the target application with the available configuration settings (such as application units, application unit rules, policies, persistence settings, load balancing methods, profiles, and traffic settings). In this application, you can configure deployment settings such as public endpoints, services, and service groups for each traffic subset. If you want the AppExpert application to manage a traffic subset that is not included in the template, you can either add an application unit for a traffic subset or modify the existing application unit. After you customize the configuration, you can also specify the order of evaluation for each traffic subset that the application manages.

Configuring an AppExpert application consists of the following steps:

1. Configuring Public Endpoints
2. Configuring Application Units
3. Specifying the Order of Evaluation
4. Viewing Application Configuration using Visualizer

Also, you can configure the policies that the template provided. If the AppExpert application template does not include policies for a particular NetScaler feature, such as Rewrite or application firewall, you can configure your own policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Public Endpoints

Aug 30, 2013

If you did not specify a public endpoint when importing an AppExpert application, you can specify public endpoints after you create the application. You can configure one public endpoint of type HTTP and one public endpoint of type HTTPS for your AppExpert application.

If endpoints are already configured for the application, you can dissociate endpoints from the AppExpert application and delete any endpoints that you no longer need. Note that when you dissociate a public endpoint from the AppExpert application, the endpoint is automatically unbound from the associated application unit, but it is not deleted from the system.

To configure public endpoints for an AppExpert application

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application for which you want to configure public endpoints, and then click Configure Public Endpoints.
3. In the Choose Public Endpoints dialog box for the application, do one of the following:
   - If the endpoints you want are listed in the dialog box, click the corresponding check boxes.
   - If you want to specify all the public endpoints, click Activate All.
   - If you want to dissociate endpoints from the AppExpert application, clear the corresponding check boxes.
   - If you want to create a new public endpoint, click Add. Then, in the Create public endpoint dialog box, configure endpoint settings, and then click OK.

     In the Create public endpoint dialog box, you can specify only the name, IP address, port, and protocol for the endpoint. You can specify additional endpoint settings after you create the public endpoint. To specify additional endpoint settings, after you create the endpoint, in the Choose Public Endpoints dialog box, click the endpoint, and then click Open. Then, in the Configure Public Endpoint dialog box, provide additional settings, and then click OK.

     For more information about the parameters in the Create public endpoint and Configure Public Endpoint dialog boxes, see "Content Switching."

   - If you want to modify a public endpoint, click the endpoint, and then click Open. Then, in the Configure Public Endpoint dialog box, modify settings for the endpoint, and then click OK.

     For more information about the parameters in the Configure Public Endpoint dialog box, see "Content Switching."

4. Click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Endpoints for an Application Unit

Aug 30, 2013

When you configure multiple public endpoints for an AppExpert application, by default, all endpoints are bound to each application unit, and each application unit processes the requests received by all the endpoints. However, you can specify that a given application unit manages the traffic that is received by only a subset of the endpoints that are configured for the AppExpert application.

To configure endpoints for an application unit

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application unit for which you want to specify public endpoints, and then click Configure Public Endpoints.
3. In the Choose Public Endpoints dialog box for the application unit, do one of the following:
   - If you are specifying endpoints for the application unit for the first time, clear the check boxes that correspond to the endpoints that you do not want to be bound to the application unit.
   - If you want to specify endpoints that are listed in the dialog box but not currently bound to the application unit, click the corresponding check boxes.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Services and Service Groups

Aug 30, 2013

When you configure a service or service group, you either modify an existing service or service group, or add new services to the AppExpert application. You add services or service groups if you did not specify them when you imported the application template. You also add services and service groups when you increase the number of servers that host instances of the application. You can configure a service and service group for an application unit only after you configure the service or service group for the AppExpert application.

To configure a service or service group for the AppExpert application

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application for which you want to configure services or service groups, and then click Configure Backend Services.
3. In the Configure Backend Services dialog box, do one of the following:
   - To configure services, click the Services tab.
   - To configure service groups, click the Service Groups tab.
4. On the Service or Service Groups tab, do one of the following:
   - If the services or service groups that you want are listed on the tab, click the corresponding check boxes.
   - If you want to specify all the services or service groups, click Activate All.
   - If you want to create a new service or service group, click Add. Then, in the Create Service dialog box or Create Service Group dialog box, configure settings for the service or service group, respectively, and then click Create.
   - If you want to modify a service, click the service, and then click Open. Then, in the Configure Service dialog box or Create Service Group dialog box, configure settings for the service or service group, respectively, and then click OK.

For information about the settings in the Create Service, Configure Service, and Create Service Group dialog boxes, see "Load Balancing."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Services, Service Groups, and Load Balancing Parameters for an Application Unit

Aug 30, 2013

When you configure services and service groups for an AppExpert application, by default, all the services and service groups are bound to each application unit. However, depending on how you have configured your web application, the application resources that are managed by an application unit might be hosted on only some of the servers that are configured as services for the AppExpert application. Or, a set of servers might host content that is meant for the requests received at one or more specific public endpoints. In such scenarios, if all the services and service groups that are configured for the AppExpert application are associated with the application unit, a request that is forwarded to a server that does not host the requested content might not be served or might be served incorrect content. Therefore, you must ensure that each application unit is configured to manage only those services that can serve the requested content.

When configuring services and service groups for an application unit, you might choose to specify load balancing settings such as the weights that services must be assigned and the desired load balancing, persistence, and spillover methods. For more information about these settings, see Load Balancing.

To configure services or service groups for an application unit

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application unit for which you want to configure a service or service group, and then click Configure Backend Services.
3. In the Configure Backend Services dialog box, do one of the following:
   - To configure services, click the Services tab.
   - To configure service groups, click the Service Groups tab.
4. In the Services or Service Groups tab, do one of the following:
   - Clear the check boxes that correspond to the services or service groups that you do not want configured for the application unit. Make sure that the check boxes that correspond to the services or service groups that you want configured for the application unit are selected. Then, in the Weight column, specify the weight that you want to assign to each configured service.
   - To specify all services or service groups, click Activate All.
5. On the Method and Persistence and Advanced tabs, specify the desired parameters.
6. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating Application Units

Aug 30, 2013

You might need to add application units for traffic subsets that are either specific to your web application implementation or not defined in the template. When creating an application unit, you must configure a rule for the application unit.

To create an application unit for the AppExpert application

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application for which you want to add an application unit, and then click Add.
3. Click Create.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Application Unit Rules

Aug 30, 2013

You might want to configure an application unit rule to include or exclude certain types of traffic. When you configure the rule, you can also define the syntax of the expression.

To configure an application unit rule

1. In the navigation pane of the NetScaler configuration utility, expand AppExpert, and then click Applications.
2. In the details pane, right-click the application unit for which you want to modify the rule, and then click Open.
3. In the Configure Application Unit dialog box, do the following:
   1. To specify the format of the new expression, do one of the following:
      - To specify that you want to configure a classic expression in the Rule box, click Classic Syntax.
      - To specify that you want to configure an advanced expression in the Rule box, click Default Syntax.
   2. In the Rule box, configure the expression.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Policies for Application Units

Apr 24, 2014

For an AppExpert application, you can configure policies for Compression, Caching, Rewrite, Responder, and Application Firewall. The templates that you download from the Citrix Community web site provide you with a set of policies that fulfill the most common application management requirements. You might want to fine-tune or customize these policies. If the set of policies provided for a given application unit does not include policies for a particular feature, you can create and bind your own policies for that feature.

If you create an AppExpert application without using a template, you must configure all the policies that the web application needs.

The GUI uses various icons to indicate whether or not policies are configured for a feature. For an application unit, if a policy is configured for a given feature, an icon that represents the feature is displayed. For example, if a compression policy is configured for an application unit, a compression icon is displayed in the Compression column for the application unit. For features for which no policy is configured, an icon depicting a plus sign (+) is displayed.

Note: When configuring policies for application units, you might need to configure policies and expressions that are either in the classic or default syntax. Additionally, when you configure default syntax policies, you might need to specify parameters such as Goto expressions and invoke policy banks. For information about configuring policies and expressions in both formats, see "Policies and Expressions."

Configuring Compression Policies

You can use either classic policies or advanced policies to configure compression, but you cannot bind compression policies of both types to the same application unit.

To configure a compression policy for an application unit

1. Navigate to AppExpert > Applications.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Compression column.
3. In the Configure Compression Policies dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click Switch to Default Syntax if you want to configure a default syntax compression policy. If you want to bind or configure classic compression policies, and if you are in the default syntax view, you can click Switch to Classic Syntax to return to the classic policy view and begin modifying bound classic policies or create and bind new classic compression policies.
     Important: This setting also determines what policies are displayed when you want to insert a policy. For example, if you are in the default syntax view, when you click Insert Policy, the list that appears in the Policy Name column will include only default syntax policies. You cannot bind policies of both types to an application unit.
   - If you want to configure classic policies, click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.
     You can configure both request-time and response-time classic compression policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.
   - To modify a compression policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Compression Policy dialog box, modify the policy, and then click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For information about modifying a compression policy, see "Compression."

- To unbind a policy, click the name of the policy, and then click Unbind Policy.
- To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
- To regenerate assigned priorities, click Regenerate Priorities.
- To insert a new policy, click Insert Policy and, in the list that is displayed in the Policy Name column, click New Policy. Then, in the Create Compression Policy dialog box, configure the policy, and then click Create.

  For information about modifying a compression policy, see "Compression."

- If you are configuring a default syntax expression, do the following:
  - In the Goto Expression column, select a Goto expression.
  - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.

4. Click Apply Changes, and then click Close.

Configuring Caching Policies

You can use only default syntax policies and expressions to configure Caching policies.

To configure Caching policies for an application unit

1. Navigate to AppExpert > Applications.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Caching column.
3. In the Configure Cache Policies dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.

     You can configure both request-time and response-time Caching policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.

   - To modify a Caching policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Cache Policy dialog box, modify the policy, and then click OK.

     For information about modifying a Caching policy, see "Integrated Caching."

   - To unbind a policy, click the name of the policy, and then click Unbind Policy.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click Regenerate Priorities.
   - To insert a new policy, click Insert Policy and, in the list that is displayed in the Policy Name column, click New Policy. Then, in the Create Cache Policy dialog box, configure the policy, and then click Create.

     For information about modifying a Caching policy, see "Integrated Caching."

   - In the Goto Expression column, select a Goto expression.
   - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click Apply Changes, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Rewrite Policies

You can use only default syntax policies and expressions to configure Rewrite policies.

To configure Rewrite policies for an application unit

1. Navigate to AppExpert > Applications.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Rewrite column.
3. In the Configure Rewrite Policies dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.

     You can configure both request-time and response-time Rewrite policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.

   - To modify a Rewrite policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Rewrite Policy dialog box, modify the policy, and then click OK.

     For information about modifying a Rewrite policy, see "Rewrite."

   - To unbind a policy, click the name of the policy, and then click Unbind Policy.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click Regenerate Priorities.
   - To insert a new policy, click Insert Policy and, in the list that is displayed in the Policy Name column, click New Policy. Then, in the Create Rewrite Policy dialog box, configure the policy, and then click Create.

     For information about modifying a Rewrite policy, see "Rewrite."

   - In the Goto Expression column, select a Goto expression.
   - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click Apply Changes, and then click Close.

Configuring Responder Policies

You can use only default syntax policies and expressions to configure Responder policies.

To configure Responder policies for an application unit

1. Navigate to AppExpert > Applications.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Responder column.
3. In the Configure Responder Policies dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - To modify a Filter policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Responder Policy dialog box, modify the policy, and then click OK.

     For information about modifying a Responder policy, see "Responder."

   - To unbind a policy, click the name of the policy, and then click Unbind Policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
- To regenerate assigned priorities, click Regenerate Priorities.
- To insert a new policy, click Insert Policy and, in the list that is displayed in the Policy Name column, click New Policy. Then, in the Create Responder Policy dialog box, configure the policy, and then click Create.

  For information about modifying a Responder policy, see "Responder."

- In the Goto Expression column, select a Goto expression.
- In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.

4. Click Apply Changes, and then click Close.

Configuring Application Firewall Policies

You can configure both classic and default syntax policies and expressions for Application Firewall. However, if a policy of one type is already bound globally or to a virtual server that is configured on the appliance, you cannot bind a policy of the other type to an application unit. For example, if a default syntax policy is already bound either globally or to a virtual server, you cannot bind a classic policy to an application unit.

To configure Application Firewall policies for an application unit

1. Navigate to AppExpert > Applications.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Application Firewall column.
3. In the Configure Application Firewall Policies dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click either Classic Expression or Advanced Expression depending on the type of expression you want to configure for the Application Firewall policy.
     Important: This setting also determines what policies are displayed when you want to insert a policy. For example, if you select Advanced Expression, when you click Insert Policy, the list that appears in the Policy Name column will include only default syntax policies. You cannot bind policies of both types to an application unit. This option is not available if a policy of either type is already bound either globally or to a virtual server.
   - To modify an application firewall policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Application Firewall Policy dialog box, modify the policy, and then click OK.
     For information about modifying a application firewall policy, see "Policies."

   - To unbind a policy, click the name of the policy, and then click Unbind Policy.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click Regenerate Priorities.
   - To insert a new policy, click Insert Policy and, in the list that is displayed in the Policy Name column, click New Policy. Then, in the Create Application Firewall Policy dialog box, configure the policy, and then click Create.
     For information about modifying a application firewall policy, see "Policies."

4. Click Apply Changes, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Application Units

Feb 05, 2016

To configure an application unit by using the configuration utility

1. Navigate to **AppExpert** > **Applications** > **Application Unit** section and then click the plus icon to add a new application unit for a traffic subset.
2. In the **Application Unit** slider, set the following parameters:
   1. Name
   2. Expression

You can insert an expression either by adding the expression components manually or by using the Expression Editor link. To manually add an expression, enter a selector component and then type a period (.) to display a list from which you can select the next component. For example, type HTTP and then type a period. A drop-down menu appears. The contents of this menu provide the keywords that can follow the initial keyword that you entered. Select a component from the drop-down menu. The Expression* text box now displays the components that you have added to the expression (for example, HTTP.REQ). Continue adding components until the complete expression is formed.

If you prefer assistance to form the expression, you can use the Expression Editor link. On the Expression Editor page, you can form an expression by selecting components from the drop-down boxes. Select the components and click Done to insert the expression on the Application Unit page.

1. Click **Continue** to bind services and service groups.
2. Click the **Service** section to select or add a virtual service and bind it to the application unit.
3. Click **Continue** and click the **Service Group** section to select or add a virtual service group and bind it to the application unit.
4. Click **Bind** and **Continue** to configure Advanced Settings (such as Policies, Method, Persistence, Protection, Profiles, Push, Authentication, and Traffic Settings) for the application unit.
5. Click the plus icon in each section to set the configuration parameters.
6. Click **OK** and then **Done**.

To edit an application unit for an application by using the configuration utility

1. Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, select an entity, click the edit icon and modify the application unit settings.

Note: You cannot modify the name and rule expression for an existing application unit.

Citrix NetScaler's video tutorials enable you to understand NetScaler features in an easy and simple way. Watch https://www.youtube.com/watch?v=bJ5_i8fV2hc video to learn how configure an application unit.

To configure an application unit for an application by using the configuration utility.

1. Navigate to **AppExpert** > **Applications** > **Application Unit** section and then click the plus icon to add a new application unit for a traffic subset.

2. In the **Application Unit** slider, set the following parameters:

   a. Name

   b. Expression

You can insert an expression either by adding the expression components manually or by using the Expression Editor link. To manually add an expression, enter a selector component and then type a period (.) to display a list from which you can select the next com

If you prefer assistance to form the expression, you can use the Expression Editor link. On the Expression Editor page, you can form an expression by selecting components from the drop-down boxes. Select the components and click **Done** to insert the expression o

1. Click **Continue** to bind services and service groups.

2. Click the **Service** section to select or add a virtual service and bind it to the application unit.

3. Click **Continue** and click the **Service Group** section to select or add a virtual service group and bind it to the application unit.

4. Click **Bind** and **Continue** to configure Advanced Settings (such as Policies, Method, Persistence, Protection, Profiles, Push, Authentication, and Traffic Settings) for the application unit.

5. Click the plus icon in each section to set the configuration parameters.

6. Click **OK** and then **Done**.

To edit an application unit for an application by using the configuration utility

Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, select an entity, click the edit icon and modify the application unit settings.

Note: You cannot modify the name and rule expression for an existing application unit.

To configure an application unit for an application by using the configuration utility.

1. Navigate to **AppExpert** > **Applications** > **Application Unit** section and then click the plus icon to add a new application unit for a traffic subset.

2. In the **Application Unit** slider, set the following parameters:

   a. Name

   b. Expression

You can insert an expression either by adding the expression components manually or by using the Expression Editor link. To manually add an expression, enter a selector component and then type a period (.) to display a list from which you can select the next com

If you prefer assistance to form the expression, you can use the Expression Editor link. On the Expression Editor page, you can form an expression by selecting components from the drop-down boxes. Select the components and click **Done** to insert the expression o

1. Click **Continue** to bind services and service groups.

2. Click the **Service** section to select or add a virtual service and bind it to the application unit.

3. Click **Continue** and click the **Service Group** section to select or add a virtual service group and bind it to the application unit.

4. Click **Bind** and **Continue** to configure Advanced Settings (such as Policies, Method, Persistence, Protection, Profiles, Push, Authentication, and Traffic Settings) for the application unit.

5. Click the plus icon in each section to set the configuration parameters.

6. Click **OK** and then **Done**.

To edit an application unit for an application by using the configuration utility

Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, select an entity, click the edit icon and modify the application unit settings.

Note: You cannot modify the name and rule expression for an existing application unit.

To configure an application unit for an application by using the configuration utility.

1. Navigate to **AppExpert** > **Applications** > **Application Unit** section and then click the plus icon to add a new application unit for a traffic subset.

2. In the **Application Unit** slider, set the following parameters:

   a. Name

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

b. Expression

You can insert an expression either by adding the expression components manually or by using the Expression Editor link. To manually add an expression, enter a selector component and then type a period (.) to display a list from which you can select the next com

If you prefer assistance to form the expression, you can use the Expression Editor link. On the Expression Editor page, you can form an expression by selecting components from the drop-down boxes. Select the components and click **Done** to insert the expression o

1. Click **Continue** to bind services and service groups.

2. Click the **Service** section to select or add a virtual service and bind it to the application unit.

3. Click **Continue** and click the **Service Group** section to select or add a virtual service group and bind it to the application unit.

4. Click **Bind** and **Continue** to configure Advanced Settings (such as Policies, Method, Persistence, Protection, Profiles, Push, Authentication, and Traffic Settings) for the application unit.

5. Click the plus icon in each section to set the configuration parameters.

6. Click **OK** and then **Done**.

To edit an application unit for an application by using the configuration utility

Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, select an entity, click the edit icon and modify the application unit settings.

Note: You cannot modify the name and rule expression for an existing application unit.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Public Endpoints for an application

Feb 05, 2016

To configure public endpoints for an application by using the configuration utility

1. Navigate to **AppExpert** > **Applications**, select an application entity, and then click **Edit**.
2. In the **Public Endpoint** section, click **+** to configure a new public endpoint.
3. In the **Public Endpoint** slider, do one of the following:
   1. Click **New** to create a new endpoint.
   2. Click **Existing Public Endpoint** to select an endpoint from the drop-down list.
4. Set the following endpoint parameters:
   1. Name
   2. IP address
   3. Protocol
   4. Port
5. Click **Continue** to configure additional settings such as application units, GSLB server bindings, policies, profiles, push, traffic settings, and authentication.
6. Click **OK** and then **Done**.
7. Click **Continue** and then **Done**.

To edit a public endpoint for an application by using the configuration utility

1. Navigate to **AppExpert** > **Applications**, select an application, and click **Edit**. In the **Public Endpoint** section, select an endpoint, click the pen icon, and modify the endpoint settings.

To delete a public endpoint for an application by using the configuration utility

1. Navigate to **AppExpert** > **Applications** > **Public Endpoint,** click the pen icon to view the delete icon next to the entity.

Citrix NetScaler's video tutorials enable you to understand NetScaler features in an easy and simply way. Watch https://www.youtube.com/watch?v=z4v-edQiVpw video to learn how to configure a public endpoint.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Specifying the Order of Evaluation of Application Units

Mar 24, 2016

Application unit rules are evaluated in the order in which they are placed in the configuration utility. The rule that is configured for the topmost application unit is always configured first, followed by the rule that is configured for the second topmost application unit, and so on. The default application unit is always evaluated last.

When a request matches the rule that is configured for an application unit, the request is processed by the application unit, and no further matching is performed. Therefore, the order of evaluation of application units becomes an important factor if the traffic subsets for two or more application units overlap. If the traffic subsets for two or more application units overlap, you must specify the order in which an incoming request is matched against the application unit rules.

To specify the order of evaluation of application units

1. Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, click the **Pencil** icon and then hover the cursor over the check box to the left of the name of the application unit. Click the icon that appears next to the check box and hold down the mouse to drag the application up or down to a new location in the priority list.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Persistency Groups for Application Units

Aug 30, 2013

You can configure a persistency group for the application units in an AppExpert application. In the context of an AppExpert application, a persistency group is a group of application units that you can treat as a single entity for the purpose of applying common persistence settings. When the application is exported to an application template file, the persistency group settings are included, and they are automatically applied to the application units when you import the AppExpert application.

To configure a persistency group for an application by using the configuration utility

1. Navigate to AppExpert > Applications.
2. In the Applications View dialog box, click the name of the application for whose application units you want to configure a persistency group, and then click Configure Persistency Groups.
3. In the Configure Persistency Groups dialog box, do one of the following:
   - To add a persistency group, click Add.
   - To modify a persistency group, click Open.
4. In the Create Persistency Group or Configure Persistency Group dialog box, set the following parameters:
   - Group Name*—Name of the persistency group. For the NetScaler appliance to recognize the persistency group as part of the application's configuration, the name of the AppExpert application must be included in the name of the persistency group, as a prefix. Therefore, by default, the appliance displays the prefix in the Group Name box, and you cannot remove that prefix. Enter a name of your choice after the prefix.
   - Persistence—Type of persistence for the virtual server. If you select SOURCEIP, in the IPv4 Netmask box, enter a network mask that specifies the number of bits that the appliance must consider when creating persistence sessions. If you select COOKIEINSERT, in the Cookie Domain and Cookie Name boxes, specify a domain attribute to send in the Set-Cookie directive, and a name for the cookie, respectively.
   - Timeout—Time period for which a persistence session is in effect.
   - Backup Persistence—Type of backup persistence for the group.
   - Backup Timeout—Time period, in minutes, for which backup persistence is in effect.
   - Application Units—To add an application unit to the persistency group, in the Available Application Units box, click the application unit, and then click Add. To remove an application unit from the persistency group, in the Configured Application Units box, click the application unit, and then click Remove.
5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing AppExpert Applications and Configuring Entities by Using the Application Visualizer

Jan 07, 2016

The Visualizer feature shows you a graphical representation of an application's configuration. It includes the name of the public endpoint, application units assigned to the public endpoint, and the number of policies and services bound to the application. You can use the Visualizer to obtain a visual overview of an AppExpert application's configuration and configure some of the displayed entities. By default, the Visualizer displays application units, services, and monitors for the selected application.

To view an AppExpert application by using the Application Visualizer

1. Navigate to **AppExpert** > **Applications**, select an application entity, and click **Visualizer**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring User Authentication, Authorization, and Auditing

Jan 05, 2016

You can configure authorization for users and groups to enable then to access an AppExpert application. If the AAA user or group for which you want to configure permissions has not already been created, you can create it from AppExpert and then configure permissions for application access.

To configure AAA users and AAA user groups for an application by using the configuring utility

1. Navigate to **AppExpert** > **Applications**, select an application entity, and then click **Edit**.
2. In the **Advanced Settings** section, click **Authorization**, and configure authorized users and user groups.
3. Click the **AAA** user section to bind authorized users to the application.
4. In the **AAA User** slider, set the parameters .
5. Click **Continue**, and then click **Authorization Policies** in the **Advanced Settings s**ection.
6. In the **Authorization Policy** slider, bind an authorization policy to the application.
7. Click **Continue**, and then click the **Authorization Group** section in the **Advanced Settings** section.
8. In the **AAA Group Binding** slider, bind an authorization user group to the application.
9. Click **Continue**, and then click **Policies** in the **Advanced Settings** section.
10. In the **Policies** slider, bind an **Audit Syslog** or **Audit NSlog** policy to the application.
11. Click **Continue** and then **Done**.

To edit AAA users and AAA user groups for an application by using the configuration utility

Navigate to **AppExpert** > **Applications** > **Advanced Settings** and click **Authorization**. Then click the edit icon and specify values for user or user-group authorization settings.

To delete AAA users and AAA user groups by using the configuration utility

Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Applications** page, click **Advanced Settings** and click **Authorization**. Click the delete icon next to the entity.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring a NetScaler Application

Mar 18, 2011

After you customize the AppExpert application, you can view application statistics to make sure that the application and all its entities are working correctly. You can also use the Application Visualizer to monitor statistics associated with certain entities such as policies and virtual servers.

You can also view the hit counters for various entities at regular intervals to make sure that counters are being updated.

Viewing Application Statistics

Updated: 2013-08-30

In the Applications node, you can select an application and view the Statistics page for the application. On the Statistics page, you can monitor the health and states of public endpoints and application units, and view the following statistical information:

- Requests and responses per second for each of the public endpoints and application units.
- Bytes per second, at each endpoint, for incoming and outgoing traffic.
- Application unit hit counters and the number of client and server connections for each application unit.
- Statistics for the services that are bound to the application units.

On the Statistics page, you can also view CPU usage, memory usage, and system logs.

To view statistics for an application

1. Navigate to AppExpert > Applications.
2. In the details pane, click the application for which you want to view statistics, and then click Statistics.

Monitoring an Application by Using the Application Visualizer

Updated: 2013-08-30

You can use the Application Visualizer to monitor the number of requests received per second at a given point in time by the vservers and the number of hits per second at a given point in time for Rewrite, Responder, and Cache policies.

To view statistical information for vservers, Rewrite policies, Responder policies, and Cache policies in the Visualizer

1. Navigate to AppExpert > Applications.
2. In the details pane, select the application for which you want to view statistical information, and then click Visualizer.
3. In the Application Visualizer window, do the following:
   - To view the statistics, click Show Stats.

     The statistical information is displayed on the respective nodes in the Visualizer. This information is not updated in real time and has to be refreshed manually.

   - To refresh the statistical information, click Refresh Stats.

Viewing Hits

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2013-08-30

The hit counters that are provided for various AppExpert application entities enable you to monitor the functioning of public endpoints and application units. For an application, the Hits dialog box displays the total number of requests received by each configured public endpoint. For an application unit, the Hits dialog box displays the number of requests that the application unit processed from each of the public endpoints and the total hit count. For instructions on viewing hit counters, see "Verifying and Testing the Configuration."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Deleting an Application

Jan 05, 2016

If you no longer need an application and its application units, you can delete it. When you delete an AppExpert application, backend services are not deleted, and any public endpoints that the application used become available for use by other applications.

When deleting an application, you are also prompted to specify whether you want to delete any bound policies and actions that are not used elsewhere.

To delete an application unit for an application by using the configuration utility

1. Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, click the delete icon next to the entity

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Authentication, Authorization, and Auditing

May 21, 2015

You can configure Authentication, Authorization, and Auditing (AAA) for the applications that you configure on the appliance. An authentication policy that is configured for an application defines the type of authentication to apply when a user or group attempts to access the application. If external authentication is used, the policy also specifies the external authentication server. Authorization policies configured for an application specify whether a particular user or group can access the application. Auditing policies define the audit log type, the level at which logging is performed, and other audit server settings. Authentication and auditing policies use the classic policy format.

Authentication policies, authorization policies, and auditing policies can be configured in any order. However, before you configure AAA for an application, you must configure a public endpoint for the application.

This document includes the following deatils:

- [Configuring Authentication]
- [Configuring Authorization]
- [Configuring Auditing]
- [Disabling AAA for an Application]

Configuring Authentication

Updated: 2013-08-30

Configuring authentication for an application involves specifying an authentication FQDN, an authentication virtual server, a server certificate, and authentication and session policies. Authentication policies are automatically bound to the authentication virtual server specified for the application.

To configure authentication for an AppExpert application

1. Navigate to AppExpert > Applications.
2. In the details pane, click the name of the application for which you want to configure authentication, and then click Authentication.
3. In the Authentication Wizard, on the Introduction page, click Next.
4. Follow the instructions in the Authentication Wizard.

Configuring Authorization

Updated: 2013-08-30

You can configure authorization for users and groups to enable then to access an AppExpert application. If the AAA user or group for which you want to configure permissions has not already been created, you can create it from AppExpert and then configure permissions for application access.

To configure permissions for a AAA user or group to access an AppExpert application

1. Navigate to AppExpert > Applications.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. In the details pane, click the AppExpert application for which you want to configure user or group access, and then click Authorization.
3. Do one of the following:
   - If the AAA user or group for which you want to configure permissions is already in the Groups/Users tree, drag the user or group from the Groups/Users tree to the Users or Groups node in the application tree. Then, right-click the user or group and click Allow.
   - If the AAA user or group for which you want to configure permissions is not configured on the appliance, in the application tree, right-click Users or Groups, and then click Add. In the Create AAA Group or Create AAA User dialog box, fill in the values, click Create, and then click Close.

     The user or group is created with the permission set to Allow. To change the permission setting, right-click the group or user, and then click the permission setting.
4. Click Close.

Configuring Auditing

Updated: 2013-08-30

When you configure auditing policies for an application, you must specify the server to which the log messages must be directed, the format of the messages logged, and the log level. Optionally, you can configure other settings, such as the log facility and date format. Auditing policies are automatically bound to all the AppExpert application's public endpoints.

To configure auditing policies for an application

1. Navigate to AppExpert > Applications.
2. In the details pane, click the application for which you want to configure auditing policies, and then click Auditing.
3. In the Configure Auditing Policies dialog box, click Insert Policy.
   - To specify an existing auditing policy, under Policy Name, click the name of the policy, and then do the following:
     - To modify the priority that is assigned to the policy by default, under Priority, double-click the priority, and then type a new priority value.
     - To modify the settings of the audit server, under Server, double-click the name of the server, and then, in the Configure Auditing Server dialog box, modify the settings as appropriate. You can modify all the settings in this dialog box except the name of the audit server and the audit type. For more information about the settings in the Configure Auditing Server dialog box, see "Auditing Policies."
   - To create a new auditing policy, under Policy Name, click New Policy, and then, in the Create Auditing Policy dialog box, do the following:
     - In the Name box, type a name for the policy.
     - The Name box already contains the string that is required at the beginning of the server name. You cannot modify the string.
     - From the Auditing Type list, select the auditing type (either SYSLOG or NSLOG).
     - If the audit server you want to specify is already listed in the Server list, select the server from the list, and then, if you want to modify the server settings, click Modify. In the Configure Auditing Server dialog box, modify the settings as appropriate, and then click OK. For more information about the settings in the Configure Auditing Server dialog box, see "Auditing Policies."
     - If you want to configure a new audit server, click New, and then, in the Create Auditing Server dialog box, type a name for the server, specify the server IP address, port number, and other settings as appropriate. When finished,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

click OK.
    - Click Create.
- To change the priorities for the new auditing policies you created, under Priority, for each policy for which you want to change the priority, double-click the priority value and type new priority value.
- To regenerate priorities, click Regenerate Priorities.
- To unbind a policy, click the policy, and then click Unbind Policy.
- To modify a policy, click the policy, and then click Modify Policy.

4. Click Apply Changes, and then click Close.

Disabling AAA for an Application

Updated: 2013-08-30

After you configure AAA for an application, you can disable the AAA configuration for that application. When you disable AAA for an application, the configuration is not lost. You can enable AAA for the application when you want to reapply the configuration.

To enable or disable AAA for an application

1. Navigate to AppExpert > Applications.
2. In the details pane, click the application for which you want to enable or disable AAA, and then do one of the following:
    - To disable AAA for the application, click Turn Off AAA.
    - To enable AAA for the application, click Turn On AAA.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting Up a Custom NetScaler Application

May 21, 2015

If an AppExpert application template is not available for the Web application that you want to manage through the NetScaler appliance, or if available AppExpert application templates do not suit your requirements, you can create an AppExpert application without a template.

To create an AppExpert application without a template, you must first create an application and application units. Then, you configure public endpoints, services, and service groups. Finally, you configure the policies that determine how application traffic is evaluated and processed.

After you create the application and application units and configure policies, you must verify the configuration and test it to make sure that it is working correctly, just as you would when you configure an application by using a prebuilt AppExpert application template. Then, you must monitor the application to make sure that the application and its entities are working correctly.

This document includes the following details:

- [Creating an Application]
- [Creating Application Units]
- [Configuring Public Endpoints for an AppExpert Application]
- [Configuring Public Endpoints for an Application Unit]
- [Configuring Services and Service Groups for an AppExpert Application]
- [Configuring Services and Service Groups for an Application Unit]
- [Configuring Policies]

Creating an Application

Updated: 2013-08-30

When you create an AppExpert application, the appliance creates a container to which you can add application units. The default application unit is not created until you create the first application unit.

To create an AppExpert application

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click Applications, and then click Add.
3. In the Create Application dialog box, in Name, enter a name for the application, and then click OK.

Creating Application Units

Updated: 2013-08-30

For each subset of traffic associated with your web application, you must create an application unit.

To create an application unit for the AppExpert application

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application for which you want to add an application unit, and then click Add.
3. Click Create.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Public Endpoints for an AppExpert Application

Updated: 2013-08-30

After you have created all the application units that you require, you must configure one or more public endpoints to enable clients to access the web application through the NetScaler appliance.

To configure public endpoints for an AppExpert application

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application for which you want to configure public endpoints, and then click Configure Public Endpoints.
3. In the Choose Public Endpoints dialog box for the application, do one of the following:
   - If the endpoints you want are listed in the dialog box, click the corresponding check boxes.
   - If you want to specify all the public endpoints, click Activate All.
   - If you want to dissociate endpoints from the AppExpert application, clear the corresponding check boxes.
   - If you want to create a new public endpoint, click Add. Then, in the Create public endpoint dialog box, configure endpoint settings, and then click OK.

     In the Create public endpoint dialog box, you can specify only the name, IP address, port, and protocol for the endpoint. You can specify additional endpoint settings after you create the public endpoint. To specify additional endpoint settings, after you create the endpoint, in the Choose Public Endpoints dialog box, click the endpoint, and then click Open. Then, in the Configure Public Endpoint dialog box, provide additional settings, and then click OK.

     For more information about the parameters in the Create public endpoint and Configure Public Endpoint dialog boxes, see "Content Switching."

   - If you want to modify a public endpoint, click the endpoint, and then click Open. Then, in the Configure Public Endpoint dialog box, modify settings for the endpoint, and then click OK.

     For more information about the parameters in the Configure Public Endpoint dialog box, see "Content Switching."

4. Click Close.

Configuring Public Endpoints for an Application Unit

Updated: 2013-08-30

For an application unit, you specify public endpoints in the same way as you would specify public endpoints for an application that is created from an AppExpert application template. For more information about specifying a subset of the endpoints for an application unit, see "Configuring Endpoints for an Application Unit."

To configure endpoints for an application unit

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application unit for which you want to specify public endpoints, and then click Configure Public Endpoints.
3. In the Choose Public Endpoints dialog box for the application unit, do one of the following:
   - If you are specifying endpoints for the application unit for the first time, clear the check boxes that correspond to the endpoints that you do not want to be bound to the application unit.
   - If you want to specify endpoints that are listed in the dialog box but not currently bound to the application unit, click

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the corresponding check boxes.

4. Click OK.

Configuring Services and Service Groups for an AppExpert Application

Updated: 2013-08-30

Services and service groups are available for application units only after you configure the services and service groups for the AppExpert application. Therefore, you must configure services and service groups for the AppExpert application before you configure the services for the application units. All the services and service groups that you configure for an AppExpert application must use the same protocol (either HTTP or HTTPS). The procedure for configuring services and service groups for an AppExpert application that is not created from a template is the same as that for an application created from a template.

To configure a service or service group for the AppExpert application

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application for which you want to configure services or service groups, and then click Configure Backend Services.
3. In the Configure Backend Services dialog box, do one of the following:
   - To configure services, click the Services tab.
   - To configure service groups, click the Service Groups tab.
4. On the Service or Service Groups tab, do one of the following:
   - If the services or service groups that you want are listed on the tab, click the corresponding check boxes.
   - If you want to specify all the services or service groups, click Activate All.
   - If you want to create a new service or service group, click Add. Then, in the Create Service dialog box or Create Service Group dialog box, configure settings for the service or service group, respectively, and then click Create.
   - If you want to modify a service, click the service, and then click Open. Then, in the Configure Service dialog box or Create Service Group dialog box, configure settings for the service or service group, respectively, and then click OK.

For information about the settings in the Create Service, Configure Service, and Create Service Group dialog boxes, see "Load Balancing."

Configuring Services and Service Groups for an Application Unit

Updated: 2013-08-30

After you configure services and service groups, you must configure services and service groups for each application unit. However, this step is not necessary if each backend service hosts all the content associated with the web application. You configure services and service groups for an application unit if the content associated with the application unit is hosted on only a subset of the backend servers.

To configure services or service groups for an application unit

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application unit for which you want to configure a service or service group, and then click Configure Backend Services.
3. In the Configure Backend Services dialog box, do one of the following:
   - To configure services, click the Services tab.
   - To configure service groups, click the Service Groups tab.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

4. In the Services or Service Groups tab, do one of the following:
   - Clear the check boxes that correspond to the services or service groups that you do not want configured for the application unit. Make sure that the check boxes that correspond to the services or service groups that you want configured for the application unit are selected. Then, in the Weight column, specify the weight that you want to assign to each configured service.
   - To specify all services or service groups, click Activate All.
5. On the Method and Persistence and Advanced tabs, specify the desired parameters.
6. Click OK.

Configuring Policies

Updated: 2013-08-30

The procedures for configuring policies for an AppExpert application that is created without using a template are the same as those for an AppExpert application that was created from a template. For more information, see "Configuring Policies for Application Units."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating and Managing Template Files

Jan 04, 2016

After you set up an AppExpert application and customize it to suit your requirements, you can create a template from the configuration and then share the template with other administrators. Or, you can create a template and then import the template to other NetScaler appliances that require a similar AppExpert application configuration. This simplifies and expedites the process of setting up similar applications on other appliances.

AppExpert application template files can be exported either to the template directory on the NetScaler appliance or to a folder on your local computer. You can then upload and download the templates to and from the NetScaler appliance and rename the templates that are stored in the AppExpert application templates directory on your appliance.

AppExpert application template files can be exported either to the template directory on the NetScaler appliance or to a folder on your local computer. You can then upload and download the templates to and from the NetScaler appliance and rename the templates that are stored in the AppExpert application templates directory on your appliance.

This document includes the following information:

- Exporting an AppExpert Application to a Template File
- Uploading and Download Template Files
- Understanding Netscaler Application Templates and Deployment Files
- Deleting an Application Template

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exporting an AppExpert Application to a Template File

Jan 07, 2016

When you export an AppExpert application, all application-configuration information is exported to a template file, and all deployment-specific information is exported to a deployment file. The string _deployment is automatically appended to the name of the template file to create the name of the deployment file. Both files are in XML format. If you choose to export the application template file to the NetScaler appliance, the template file is stored in the /nsconfig/nstemplates/applications directory and the deployment file is stored in the /nsconfig/nstemplates/applications/deployment_files/ directory. If you have configured a NetScaler Gateway application, you can choose to include the NetScaler Gateway policies in the template.

To export an AppExpert application to a template file

1. Navigate to **AppExpert** > **Application**, select an application entity, and then click **Edit**.
2. On the **Applications** page, click the **Export** as a Template link to export the application configuration and deployment settings as a template.
3. In the **Export Application** slider, set the following parameters:
   1. Template Filename
   2. Deployment Filename
4. Click **Continue** and **Done**.
5. Navigate to **AppExpert** > **Application** and click **Manage Templates** to show the exported configuration as files on the **Template File** and **Deployment File** tabs.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exporting a Content Switching Virtual Server Configuration to a Template File

Aug 30, 2013

You can also export a content switching configuration as an application template. You can export a content switching virtual server configuration to an application template either from the Content Switching Virtual Servers pane or from the Content Switching Visualizer. Configuration information, which includes the content switching virtual server, all associated load balancing virtual servers, services, service groups, and policies, is exported to a template file and all deployment-specific information is exported to a deployment file. The string "_deployment" is automatically appended to the name of the template file to create the name of the deployment file. Both files are in XML format. If you choose to export the application template file to the NetScaler appliance, the template file is stored in the /nsconfig/nstemplates/applications directory on the NetScaler appliance and the deployment file is stored in the /nsconfig/nstemplates/applications/deployment_files/ directory. For more information about the format of application templates and deployment files, see "Understanding NetScaler Application Templates and Deployment Files" section. The configuration information that is exported includes the content switching virtual server, all associated load balancing virtual servers, services, service groups, and policies.

However, if the content switching virtual server is already configured as the public endpoint for an AppExpert application, you cannot export the configuration to a template file . In this scenario, you must export the associated AppExpert application to a template. For more information about exporting an AppExpert application to a template file, see "Exporting an AppExpert Application to a Template File" section.

To export a content switching configuration to an application template file from the Content Switching Visualizer

1. Navigate to Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, click the name of the content switching virtual server whose configuration you want to export as a template file, and then click Visualizer.
3. In the Content Switching Visualizer, click the icon for the content switching vserver, click Related Tasks, and then click Create Template.
4. In the Export...as Template dialog box, enter a name for the template file, and then do one of the following:
   - To export the template file to the appliance, make sure that Browse (Appliance) is displayed.
   - To export the template file to your computer, click the Browse (Appliance) drop-down menu, click Local, browse to the location to which you want to save the file, and then click Save.
5. Provide the following information:
   - **Introduction Description**—Any text that introduces the AppExpert application template during import. This text is displayed on the Specify Application Name page of the AppExpert Template Wizard when the template is imported.
   - **Summary Description**—Any summary that you might want to display on the Summary page of the AppExpert Template Wizard when the template is imported.
   - **Author**—The name of the author of the template.
   - **Major**—The major version number of the template.
   - **Minor**—The minor version number of the template. This number is appended to the major version number and displayed on the Summary page of the AppExpert Template Wizard, during import, in the format Major.Minor.
6. Click OK.

To export a content switching configuration to an application template file from the Content Switching

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Virtual Servers pane

1. Navigate to Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, click the name of the content switching virtual server whose configuration you want to export as a template file, and then click Create AppExpert Template.
3. Perform steps 4 through 6 described in "To export a content switching configuration to an application template file from the Content Switching Visualizer".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating Variables in Application Templates

Aug 30, 2013

Application templates support the declaration of variables in the policy expressions and actions that are configured for an application. The ability to declare variables in policy expressions and actions enables you to replace preconfigured values in expressions (for example, configurable parameters such as the host name of a server or the target for a Rewrite action) with values that suit the environment into which you are importing the template. If variables have been configured for an AppExpert application template, the AppExpert Template Wizard, which appears when you import an AppExpert application template, includes a Specify Variable Values page on which you can specify appropriate values for the variables that are configured for the template.

As an example, consider the following policy expression that is configured to evaluate the value of the Host header in an HTTP request:

```
HTTP.REQ.HEADER("Host").CONTAINS("server1")
```

If you want the server name to be configurable at import time, you can specify the string "server1" as a variable. When importing the template, you can specify a new value for the variable on the Variables tab.

After you create a variable, you can do the following:

- Assign additional strings to an existing variable. After you create a variable for a string, you can select and assign other parts of the same or different expression to the variable. The strings you assign to a variable need not be the same. At import time, all the strings that are assigned to the variable are replaced with the value that you provide.
- View the string or strings that are assigned to the variable.
- View a list of all the entities and parameters that use the variable.

In the export application template wizard, you can define variables in certain fields (fields with an adjacent button) for the following entities:

- Cache policies
- Rewrite policies
- Rewrite actions
- Responder policies
- Responder actions

To configure a variable in a policy expression or action

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application that you want to export to a template file, and then click Export.
3. In the Export...as Template dialog box, modify the default template file name if required, specify the location where you want to save the template, and then click Configure Variables.
4. In the Configure Variables dialog box, click the tab that lists the policy expression or action for which you want to configure a variable, select the expression, and then click Configure Variables.
5. In the Variables dialog box, click the button next to the expression or value in which you want to create a variable.
6. In the Variables dialog box, do the following:
   - To create a variable, in the text box that displays the configured expression or value, select the string that you want to be configurable at import time, and then click Add. In the Add Variable dialog box, specify a name and a description for the variable, and then click Create.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- The name of the variable, its value, and the description you provided appear in the Available Variables listing in the dialog box. The name you provide will be the name of the associated field in the template import wizard, and the description will appear as alt text when the user positions the mouse pointer over the field.
- To modify a variable, in the Available Variables list, click the variable, and then click Open. In the Add Variable dialog box, modify the value and the description, and then click OK.
- To view all the strings that are assigned to a given variable, in the Available Variables listing, click the name of the variable. The strings that are assigned to the variable are highlighted.
- To view a list of all the entities and parameters in which the variable is used, in the Available Variables listing, click the variable whose references you want to view, and then click Show References.
- To assign a string to an existing variable, in the text box that displays the expression you configured, select the string you want to assign to an existing variable, right-click the selection, click Use Existing Selection, and then click the name of the variable to which you want to assign the string.

  If a variable has multiple strings assigned to it, when you specify a new value for the variable during import, all strings assigned to the variable are replaced with the new value.

7. Click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Uploading and Downloading Template Files

Jan 07, 2016

Template files can be uploaded from your local computer to the NetScaler appliance or downloaded from the appliance to your local computer. On the appliance, AppExpert application templates are always stored in the AppExpert application templates directory, which is /nsconfig/nstemplates/applications/.

To upload an AppExpert application template from your local computer to the NetScaler appliance

1. Navigate to AppExpert > Templates.
2. In the details pane, click Manage Templates.
3. In the Manage Application Templates dialog box, click Application Templates, and then click Upload.
4. In the Upload Application Template dialog box, browse to the directory in which the template file is stored, click the template file, and then click Select.
   The template file is uploaded to the AppExpert application template directory on the appliance.

To download an AppExpert application template from the NetScaler appliance to your local computer

1. Navigate to AppExpert > Templates.
2. In the details pane, click Manage Templates.
3. In the Manage Application Templates dialog box, click the AppExpert application template that you want to download, and click Download.
4. In the Download Application Template dialog box, browse to the location to which you want to save the file, and then click Save.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding NetScaler Application Templates and Deployment Files

Jan 07, 2016

When you export a NetScaler application, the following two files are automatically created:

- **NetScaler application template file**. Contains application-configuration information such as application units, rules, and configured policies.
- **Deployment file**. Contains deployment-specific information such as public endpoints, services, associated IP addresses, and configured variables.

In a template file or deployment file, each unit of application-configuration information is encapsulated in a specific XML element that is meant for that unit type. For example, each public endpoint and associated endpoint details are encapsulated within the <appendpoint> and </appendpoint> tags, and all the endpoint elements are encapsulated within the <appendpoint_list> and </appendpoint_list> tags.

Note: After you export a NetScaler application, you can add elements, remove elements, and modify existing elements before importing the application to a NetScaler appliance.

Example of a NetScaler Application Template

Following is an example of a template file that was created from a NetScaler application called "SharePoint_Team_Site":

```
<?xml version="1.0" encoding="UTF-8" ?>
<template>
<template_info>
  <application_name>SharePoint_Team_Site</application_name>
  <templateversion_major>1</templateversion_major>
  <templateversion_minor>1</templateversion_minor>
  <author>Ed</author>
  <introduction>An application for managing a SharePoint team site with images, reports, and, XML content.</introduction>
  <summary>This template includes variables</summary>
  <version_major>9</version_major>
  <version_minor>3</version_minor>
  <build_number>38</build_number>
</template_info>
<apptemplate>
  <rewrite>
    <rewriteaction_list>
      <rewriteaction>
        <name>Rw_name</name>
        <type>replace</type>
        <target>HTTP.REQ.BODY(10000).AFTER_REGEX(re/number/).BEFORE_REGEX(re/address/)</target>
        <stringbuilderexpr>"NA"</stringbuilderexpr>
        <allow_unsafe_pi1>NO</allow_unsafe_pi1>
      </rewriteaction>
      <rewriteaction>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        .
        .
        .
      </rewriteaction>
      .
      .
      .
    </rewriteaction_list>
    <rewritepolicy_list>
      <rewritepolicy>
        <name>Rw_number_NA</name>
        <rule>HTTP.REQ.BODY(100000).CONTAINS("admin")</rule>
        <action>Rw_name</action>
      </rewritepolicy>
      <rewritepolicy>
        .
        .
        .
      </rewritepolicy>
      .
      .
      .
    </rewritepolicy_list>
  </rewrite>
  <appunit_list>
    <appunit>
      <name>SharePoint_Team_Sitedefault</name>
      <rule />
      <expressiontype>PE</expressiontype>
      <servicetype>HTTP</servicetype>
      <ipv46>0.0.0.0</ipv46>
      <ipmask>*</ipmask>
      <port>0</port>
      <range>1</range>
      <persistencetype>NONE</persistencetype>
      <timeout>2</timeout>
      <persistencebackup>NONE</persistencebackup>
      <backuppersistencetimeout>2</backuppersistencetimeout>
      <lbmethod>LEASTCONNECTION</lbmethod>
      <persistmask>255.255.255.255</persistmask>
      <v6persistmasklen>128</v6persistmasklen>
      <pq>OFF</pq>
      <sc>OFF</sc>
      <m>IP</m>
      <datalength>0</datalength>
      <dataoffset>0</dataoffset>
      <sessionless>DISABLED</sessionless>
      <state>ENABLED</state>
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
      <connfailover>DISABLED</connfailover>
      <clttimeout>180</clttimeout>
      <somethod>NONE</somethod>
      <sopersistence>DISABLED</sopersistence>
      <redirectportrewrite>DISABLED</redirectportrewrite>
      <downstateflush>DISABLED</downstateflush>
      <gt2gb>DISABLED</gt2gb>
      <ipmapping>0.0.0.0</ipmapping>
      <disableprimaryondown>DISABLED</disableprimaryondown>
      <insertvserveripport>OFF</insertvserveripport>
      <authentication>OFF</authentication>
      <authn401>OFF</authn401>
      <push>DISABLED</push>
      <pushlabel>none</pushlabel>
      <l2conn>OFF</l2conn>
    </appunit>
    <appunit>
      .
      .
      .
    </appunit>
    .
    .
    .
  </appunit_list>
  </apptemplate>
  <parameters>
    <property_list>
      <property>
        <variable_definition_list>
          <variable_definition>
            <name>body_size</name>
            <defaultvalue>10000</defaultvalue>
            <description>Evaluation Scope</description>
            <startindex>14</startindex>
            <length>5</length>
          </variable_definition>
          .
          .
          .
        </variable_definition_list>
        <object_type>rewriteaction</object_type>
        <object_name>Rw_name</object_name>
        <name>target</name>
      </property>
      .
      .
      .
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    </property_list>
  </parameters>
</template>
```


Example of a Deployment File

Following is the deployment file associated with the "SharePoint_Team_Site" application in the preceding example:


```
<?xml version="1.0" encoding="UTF8" ?>
<template_deployment>
  <template_info>
    <application_name>SharePoint_Team_Site</application_name>
    <templateversion_major>1</templateversion_major>
    <templateversion_minor>1</templateversion_minor>
    <author>Ed</author>
    <introduction>An application for managing a SharePoint team site with images, reports, and, XML content.</introduction>
    <summary>This template includes variables</summary>
    <version_major>9</version_major>
    <version_minor>3</version_minor>
    <build_number>38</build_number>
  </template_info>
  <appendpoint_list>
    <appendpoint>
      <ipv46>10.111.111.1</ipv46>
      <port>80</port>
      <servicetype>HTTP</servicetype>
    </appendpoint>
  </appendpoint_list>
  <service_list>
    <service>
      <ip>10.102.29.5</ip>
      <port>80</port>
      <servicetype>HTTP</servicetype>
    </service>
    <service>
      .
      .
      .
    </service>
    .
    .
    .
  </service_list>
  <variable_list>
    <variable>
      <name>body_size</name>
      <description>Evaluation Scope</description>
      <value>10000</value>
    </variable>
    <variable>
      .
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        .
        .
      </variable>
      .
      .
      .
    </variable_list>
  </template_deployment>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Deleting a Template File

Jan 07, 2016

If you no longer need an application template and its configuration, you can delete it. When you delete a template, the template XML file that is stored in the application template directory gets deleted. When you delete a template file, you are prompted to confirm the deletion. Click **Yes** to confirm and delete the selected file from the directory.

To delete a template file from the application template directory by using the configuration utility

1. Navigate to **AppExpert** > **Applications** and then click **Manage Template**. Select a file from **Template Files** tab page or **Deployment Files** tab page and click **Delete**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler Gateway Applications

May 21, 2015

When you configure an AppExpert application to manage a web application through the Citrix® NetScaler® appliance, you also create a set of application units and configure a set of traffic optimization and security policies for each unit. The policies that you configure for each application unit (policies for features such as Compression, Caching, and Rewrite) evaluate traffic that is meant only for that unit. In addition to these policies, you might want to configure Access Gateway policies for the application as a whole to optimize the application traffic when accessed through the Access Gateway. The Access Gateway Applications feature enables you to configure Access Gateway policies (Authorization, Traffic, Clientless Access, and TCP Compression) for an AppExpert application. After you configure NetScaler Gateway policies for AppExpert applications, you can include the policy configuration in the AppExpert application templates that you create.

You can also configure NetScaler Gateway policies for intranet subnets, file shares, and other network resources.

Finally, you can create bookmarks for AppExpert applications and certain resources if you want users to be able to access them from the NetScaler Gateway home page.

You can configure the entities in the NetScaler Gateway Applications feature only by using the configuration utility.

How an NetScaler Gateway Application Works

Updated: 2013-07-17

When you create an AppExpert application in the Applications node in the configuration utility, a corresponding Access Gateway application is automatically created in the Access Gateway Applications node. Additionally, a rule that uses the AppExpert application's configured public endpoint is automatically created for the Access Gateway application entry. If multiple endpoints are configured for the AppExpert application, the rule includes all the configured public endpoints. The NetScaler appliance uses this rule to apply any configured Access Gateway policies to the traffic received at the AppExpert application's public endpoint. Traffic received at the AppExpert application's public endpoint is first evaluated against the NetScaler Gateway policies and then evaluated against the policies configured for AppExpert application's application units.

The rule that is created for the Clientless Access policies for an Access Gateway application is an advanced expression that also uses the public endpoint that is configured for the AppExpert application. Therefore, before you configure NetScaler Gateway policies for an AppExpert application, you must configure public endpoints for the AppExpert application.

When you include the NetScaler Gateway configuration in an application template, deployment-specific information, such as IP address and port information, and the rule that is created from this information are not included in the template.

How a NetScaler Configuration for a File Share Works

On the NetScaler appliance, you can configure Authorization policies for a file share that is hosted on your organization's network.

When you create a file share, you specify a name for the file share and the network path to the file share. In the network path, you can specify either the name of the server or the server IP address. A rule that uses the components of the file share path is automatically created for the file share. This rule enables the appliance to identify requests for files hosted on the file share server. Any Authorization policies that are configured for the file share are applied to incoming requests.

The NetScaler configuration for a file share cannot be saved in AppExpert application templates.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How a NetScaler Configuration for an Intranet Subnet Works

For the intranet subnets that form a part of your network, you can configure policies for Authorization, Traffic, and TCP Compression on the NetScaler appliance. When adding an intranet subnet, you specify the IP address and the netmask of the intranet subnet. A rule that uses these two parameters is automatically created for the intranet subnet. The appliance applies the configured policies to any request that has a destination IP address and netmask set to the subnet's IP address and netmask, respectively.

The NetScaler configuration for an intranet subnet cannot be saved in AppExpert application templates.

How the Other Resources Category Works

Updated: 2013-07-18

The Other Resources category enables you to configure Access Gateway policies for any network resource by using a rule of your choice. When you configure the NetScaler appliance to process requests for the network resource, you configure a classic expression to identify the requests that are associated with the network resource. You can configure Authorization, Traffic, Clientless Access, and TCP Compression policies for a network resource in Other Resources. The NetScaler appliance applies the configured NetScaler Gateway policies to any requests that match the configured rule.

The NetScaler configuration for a network resource in Other Resources cannot be saved in AppExpert application templates.

Entity Naming Conventions

The NetScaler Gateway Applications feature enforces a naming convention for some of the entities that you create in this feature. For example, the names of the profiles that you create for Traffic policies for an intranet subnet always begin with a string that consists of the name of the intranet subnet followed by an underscore (_). The name that you provide for the entity is appended to this string. If the name of a subnet is "subnet1," the name of the profile begins with "subnet1_." When such a naming convention is required (in the text box in which you type the name of an entity, for example), the user interface automatically inserts the string with which the name of the entity must begin and does not allow you to modify it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding Intranet Subnets

Mar 28, 2012

You can specify authorization and Traffic policies for traffic that is bound for the intranet subnets that are configured in your network. The rules for these policies are automatically created by using the parameters you specify for the subnet.

To configure an intranet subnet

1. In the navigation pane of the NetScaler configuration utility, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, do one of the following:
   - To add an intranet subnet, click Intranet Subnets, and then click Add.
   - To modify an intranet subnet, click an intranet subnet, and then click Open.
3. In the Create Intranet Subnet or Configure Intranet Subnet dialog box, do the following:
   1. In the Name box, type a name for the intranet subnet you are adding. This parameter cannot be changed for an existing intranet subnet.
   2. In the IP Address box, type the IP address of the intranet subnet.
   3. In the Netmask box, type the netmask that will be used for the intranet subnet.
   4. Click Create or OK, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding Other Resources

Jul 17, 2013

For a network resource that you add to Other Resources, you must configure a classic expression that identifies the subset of traffic associated with the resource. For more information about configuring a classic expression, see the .

To configure a resource in Other Resources

1. In the navigation pane of the NetScaler configuration utility, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, do one of the following:
   - To add a resource, click Other Resources, and then click Add.
   - To modify a resource, click a resource, and then click Open.
3. In the Create Resource or Configure Resource dialog box, do the following:
   1. In the Name box, type a name for the resource you are adding. This parameter cannot be changed for an existing resource.
   2. In the Rule box, type the rule that will identify the subset of traffic that is associated with the resource you are adding.
      Alternatively, click Configure, and then create the rule in the Create Expression dialog box.
   3. Click Create or OK, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Authorization Policies

Mar 28, 2012

You can configure NetScaler Gateway authorization policies for AAA users and groups to access a resource.

To configure permissions for a AAA user or group to access a resource

1. In the navigation pane of the NetScaler configuration utility, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, in the Authorization column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure authorization policies for AAA users and groups.
3. Do one of the following:
   - If the AAA user or group for which you want to configure permissions is already in the Groups/Users tree, drag the user or group from the Groups/Users tree to the Users or Groups node in the <application name> tree. Then, right-click the user or group and click Allow.
   - If the AAA user or group for which you want to configure permissions is not configured on the appliance, in the <application name> tree, right-click Users or Groups, and then click Add. In the Create AAA Group or Create AAA User dialog box, fill in the values, click Create, and then click Close.

     The user or group is created with the permission set to Allow. To change the permission setting, right-click the group or user, and then click the permission setting.
4. Click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Traffic Policies

Dec 16, 2013

The traffic policies that you configure for the resources in the NetScaler Gateway Applications node control client connections to the application. You do not have to configure a rule for the resource. The rule created automatically when you create the resource. You only need to associate a request profile with the traffic policy. In the traffic profile, you specify parameters such as the protocol, application time-out, and file type association.

To configure traffic policies for a resource

1. In the navigation pane of the NetScaler configuration utility, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, in the Traffic column, click the icon provided for the application, file share, intranet subnet, or resource for which you want to configure traffic policies.
3. In the Configure Traffic Policies dialog box, do the following:
   - To specify an existing traffic policy, click Insert Policy, and then, in the Policy Name column, click the name of the policy.
   - To configure a new policy, click Insert Policy, and then, in the Policy Name column, click New Policy. In the Create Traffic Policy dialog box, in the Name box, after the underscore (_), type a name for the policy. Then, in Request Profile, either select an existing request profile or click New to configure a new request profile. You can also select an existing profile and then click Modify to modify the profile.

     For more information about configuring a traffic policy or profile, see NetScaler Gateway , Enterprise Edition at http://edocs.citrix.com/.

   - To modify a policy that you have inserted, in the Policy Name column, click the policy name, and then click Modify Policy. To modify only the associated profile, in the Profile column, click the name of the profile, and then click Modify Profile.
   - To regenerate the priorities assigned to the policies, click Regenerate Priorities.
   - To specify a new priority value for a policy, in the Priority column, double-click the assigned priority, and then enter the value you want.
   - To unbind a policy, click the policy, and then click Unbind Policy.
4. Click Apply Changes, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Clientless Access Policies

Dec 16, 2013

Clientless access, when configured for a resource on the NetScaler appliance, allows end-users to access the resource without using the NetScaler Gateway client software. Users can use web browsers to access resources such as Outlook Web Access. You configure clientless access for a resource by configuring a clientless access policy that is associated with a clientless access profile.

To configure a clientless access policy for a resource in the NetScaler Gateway Applications node

1. In the navigation pane of the NetScaler configuration utility, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, in the Clientless Access column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure a clientless access policy.
3. In the Configure Clientless Access Policies dialog box, do the following:
   - To specify an existing clientless access policy, click Insert Policy, and then, in the Policy Name column, click the name of the policy.
   - To configure a new clientless access policy, click Insert Policy, and then, in the Policy Name column, click New Policy. In the Create Clientless Access Policy dialog box, in the Name box, after the underscore (_), type a name for the policy. Then, in Profile, either select an existing profile or click New to configure a new profile. You can also select an existing profile and then click Modify to modify the profile.

     For more information about configuring a clientless access policy or profile, see NetScaler Gateway , Enterprise Edition at http://edocs.citrix.com/.

   - To modify a policy that you have inserted, in the Policy Name column, click the policy name, and then click Modify Policy. To modify only the associated profile, in the Profile column, click the name of the profile, and then click Modify Profile.
   - To specify a new priority value for a policy, in the Priority column, double-click the assigned priority, and then enter the value you want.
   - To unbind a policy, click the policy, and then click Unbind Policy.
4. Click Apply Changes, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring TCP Compression Policies

Dec 16, 2013

You can configure TCP compression policies for an application to increase the performance of the application. TCP compression reduces network latency, reduces bandwidth requirements, and increases the speed of transmission. When configuring a TCP compression policy, you associate a compression action with the policy. The compression action specifies either Compress, GZIP, Deflate, or NoCompress as the compression type. For more information about the compression policies, and compression actions, see NetScaler Gateway , Enterprise Edition at http://edocs.citrix.com/.

To configure a TCP compression policy for a resource in the NetScaler Gateway Applications node

1. In the navigation pane of the NetScaler configuration utility, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, in the TCP Compression column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure a TCP compression policy.
3. In the Configure TCP Compression Policies dialog box, do the following:
   - To specify an existing TCP compression policy, click Insert Policy, and then, in the Policy Name column, click the name of the policy.
   - To create a new TCP compression policy, click Insert Policy, and then, in the Policy Name column, click New Policy. In the Create TCP Compression Policy dialog box, in the Policy Name box, after the underscore ("_"), type a name for the policy. Then, in Action, either select an existing action or click New and configure a new action. You can also click View to view the configured compression type.

     For more information about configuring a TCP compression policy or action, see NetScaler Gateway , Enterprise Edition at http://edocs.citrix.com/.

   - To modify a policy that you have inserted, in the Policy Name column, click the policy name, and then click Modify Policy.
   - To regenerate the priorities assigned to the policies, click Regenerate Priorities.
   - To specify a new priority value for a policy, in the Priority column, double-click the assigned priority, and then enter the value you want.
   - To unbind a policy, click the policy, and then click Unbind Policy.
4. Click Apply Changes, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Bookmarks

Oct 23, 2017

You can configure bookmarks for an application or for a resource that you configure in the Other Resources category if you want the application or resource to be accessible from the NetScaler Gateway home page.

To configure a bookmark for an NetScaler Gateway application or a resource in the Other Resources category

1. In the configuration utility, on the **Configuration** tab, in the navigation pane, expand **NetScaler Gateway** > **Resources** > and then click **Bookmarks.**
2. In the details pane, click **Add**, and then do one of the following:
3. In the **Create Bookmark** page, set the following parameters:
   1. Name. Name of the bookmark.
   2. Text to display. Text description that appears in the Access Interface.
   3. Bookmark. Type web address.
   4. Virtual Server. Name of the virtual server.
   5. Icon URL. URL of the display icon.
   6. Application Type. Type of application.
   7. SSO Type. Gateway authentication type.
   8. Use NetScaler Gateway as a reverse proxy. Select if the gateway is used as a reverse proxy.
   9. Comment. A brief description about the bookmark.
4. Click **Create** or **OK**, and then click **Close**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppQoE

Sep 30, 2015

Application level Quality of Experience (AppQoE) integrates several existing policy-based security features of the NetScaler appliance into a single integrated feature that takes advantage of a new queuing mechanism, fair queuing. Fair queuing manages requests to load-balanced web servers and applications at the virtual server level instead of at the service level, allowing it to handle queuing of all requests to a web site or application as one group before load balancing, instead of as separate streams after load balancing.

The features that are integrated into AppQoE are HTTP Denial-of-Service Protection (HDOSP), Priority Queuing (PQ), and SureConnect. Collectively these services provide protection against a number of problems:

- **Simple overload.** Any server, no matter how robust, can accept only a limited number of connections at one time. When a protected web site or application receives too many requests at once, the Surge Protection feature detects the overload and queues the excess connections til the server can accept them. The Priority Queuing feature ensures that whoever most needs access to a resource is provided access without having to wait behind other lower-priority requests. The SureConnect feature displays an alternate web page that notifies users that the resource that they requested is not available.
- **Denial-of-Service (DOS) attacks.** Any public-facing resource is vulnerable to attacks whose purpose is to bring that service down and deny legitimate users access to it. The Surge Protection, Priority Queuing, and SureConnect features help manage DOS attacks as well as other types of high load. In addition, the HTTP Denial-of-Service Protection feature targets DOS attacks against your web sites, sending challenges to suspected attackers and dropping connections if the clients do not send an appropriate response.

Until the current version of the NetScaler operating system, these features were implemented at the service level, which means that each service was assigned its own queues. While service-level queues work, they also have some disadvantages, most of which are due to the NetScaler appliance having to load balance requests before implementing any of the protection features that rely on queuing. Implementing protection features before queuing has a number of advantages, some of which are listed below:

- Absolute priority of connections as configured in the priority queuing feature can be maintained.
- Connections are not flushed if a service transitions state, as they are in a service-level queue.
- During periods of high load, such as a denial-of-service attack, HTTP DoS and SureConnect come into play before load balancing, allowing these features to detect and divert unwanted or lower-priority traffic from the load balancer before the load balancer must cope with it.

In addition to implementing fair queuing, AppQoE integrates a set of features that each provide a different set of tools to achieve a common goal: protecting your networked resources from excessive or inappropriate demand. Putting these features into a common framework enables you to configure and implement them more easily.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling AppQoE

Apr 20, 2013

To configure AppQoE, you must first enable the feature.

To enable AppQoE by using the command line

At the command prompt, type the following commands:

- enable ns feature appqoe
- show ns feature

Example

```
> enable ns feature appqoe
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging            WL               ON
 2)   Surge Protection       SP               ON
 3)   Load Balancing         LB               ON
 ...
 29)  AppQoE                 AppQoE           ON
 Done
```

To enable AppQoE by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, click Configure Advanced Features.
3. In the Configure Advanced Features dialog box, select the AppQoE check box.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppQOE Actions

Mar 18, 2016

After enabling the AppQoE feature, you must configure one or more actions for handling requests.

Important: No specific individual parameters are required to create an action, but you must include at least one parameter or you cannot create the action.

To configure an AppQoE action by using the command line

At the command prompt, type the following commands:

- add appqoe action <name> [-priority <priority>] [-respondWith (ACS | NS) [<customfile>] [-altContentSvcName <string>] [-altContentPath <string>] [-maxConn <positive_integer>] [-delay <usecs>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-dosTrigExpression <expression>] [-dosAction (**SimpleResponse** | **HICResponse**)]
- show appqoe action

Example

To configure priority queuing with policy queue depths of 10 and 1000 for medium and lowest priority queues, respectively:

```
> add appqoe action appqoe-act-basic-prhigh -priority HIGH
 Done

> add appqoe action appqoe-act-basic-prmedium -priority MEDIUM -polqDepth 10
 Done

> add appqoe action appqoe-act-basic-prlow -priority LOW -polqDepth 1000
 Done

> show appqoe action
1)    Name: appqoe-act-basic-prhigh
      ActionType: PRIORITY_QUEUING
      Priority: HIGH
      PolicyQdepth: 0
      Qdepth: 0

2)    Name: appqoe-act-basic-prmedium
      ActionType: PRIORITY_QUEUING
      Priority: MEDIUM
      PolicyQdepth: 10
      Qdepth: 0

3)    Name: appqoe-act-basic-prlow
      ActionType: PRIORITY_QUEUING
      Priority: LOW
      PolicyQdepth: 1000
      Qdepth: 0
Done
```

To modify an existing AppQoE action by using the command line

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands:

- set appqoe action <name> [-priority <priority>] [-altContentSvcName <string>] [-altContentPath <string>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-maxConn <positive_integer>] [-delay <usecs>] [-dosTrigExpression <expression>] [-dosAction (SimpleResponse | HICResponse)]
- show appqoe action

To remove an AppQoE action by using the command line

At the command prompt, type the following commands:

- rm appqoe action <name>
- show appqoe action

Parameters for configuring an AppQoE action

name

A name for the new action, or the name of the existing action that you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

priority

The priority queue to which the request is assigned. When a protected web server or application is heavily loaded and cannot accept additional requests, specifies the order in which waiting requests are to be fulfilled when resources are available. The choices are:

1. **HIGH.** Fulfills the request as soon as resources are available.
2. **MEDIUM.** Fulfills the request after it has fulfilled all requests in the HIGH priority queue.
3. **LOW.** Fulfills the request after it has fulfilled all requests in the HIGH and MEDIUM priority queues.
4. **LOWEST.** Fulfills the request only after it has fulfilled all requests in higher-priority queues.

If priority is not configured, then the NetScaler appliance assigns the request to the LOWEST priority queue by default.

respondWith

Configures the NetScaler ADC to take the specified Responder action when the specified threshold is reached. Must be used with one of the following settings:

- **ACS:** Serves content from an alternate content service. Threshold: maxConn (maximum connections) or delay.
- **NS:** Serves a built-in response from the NetScaler ADC. Threshold: maxConn (maximum connections) or delay.
- **NO ACTION:** Serves no alternative content. Assigns connections to the LOWEST priority queue if the maxConn (maximum connections) or delay threshold is reached.

altContentSvcName

If -responseWith ACS is specified, the name of the alternative content service, usually an absolute URL to the web server that hosts the alternate content.

altContentPath

If -responseWith (ACS | NS) is specified, the path to the alternative content.

polqDepth

Policy queue depth threshold value for the policy queue associated with this action. When the number of connections in the policy queue associated with this action increases to the specified number, subsequent requests are assigned to the LOWEST policy queue. Minimum value: 1 Maximum value: 4,294,967,294

priqDepth

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy queue depth threshold value for the specified priority queue. If the number of requests in the specified queue on the virtual server to which the policy associated with the current action is bound increases to the specified number, subsequent requests are assigned to the LOWEST priority queue. Minimum value: 1 Maximum value: 4,294,967,294

maxConn

The maximum number of connections that can be open for requests that match the policy rule. Minimum value: 1 Maximum value: 4,294,967,294

delay

The delay threshold, in microseconds, for requests that match the policy rule. If a matching request has been delayed for longer than the threshold, the NetScaler appliance performs the specified action. If NO ACTION is specified, then the appliance assigns requests to the LOWEST priority queue. Minimum value: 1 Maximum value: 599999,999

dosTrigExpression

Adds an optional second-level check to trigger DoS actions.

dosAction

Action to take when the ADC determines that it or a protected server is under DoS attack. Possible values: SimpleResponse, HICResponse.

These values specify HTTP challenge-response methods for validating the authenticity of incoming requests to mitigate an HTTP-DDoS attack.

In the HTTP challenge-response generation and validation process, AppQoE uses cookies to validate the client's response and verify that the client seems to be genuine. When sending a challenge, a NetScaler appliance generates two cookies:

Header cookie (_DOSQ). Contains client-specific information, so that the NetScaler appliance can verify the response.

Body cookie (_DOSH). Information used to validate the client machine. The client's browser (or the user, in the case of HIC) computes a value for this cookie. The NetScaler appliance compares that value with the expected value to verify the client.

The information that the appliance sends to the client for computing the _DOSH value is based on the DoS Action configuration.

1) SimpleResponse: In this case, a NetScaler appliance splits the value and generates a JavaScript code to combine the final value. A client machine capable of computing the original value is considered genuine.

2) HICResponse: in this case, a NetScaler appliance generates two single-digit numbers and generates images for those numbers. Then, using a backpatch framework, the appliance inserts those images as base64 strings.

Limitations:

1. This is not a trivial CAPTCHA implementation, which is why that term not used.

2. The validation number is based on a NetScaler-generated number that does not change for 120s. This number should be dynamic or client specific.

To configure an AppQoE action by using the configuration utility

1. Navigate to App-Expert > AppQoE > Actions.
2. In the details pane, do one of the following:
   - To create a new action, click Add.
   - To modify an existing action, select the action, and then click Edit.
3. In the Create AppQoE Action or the Configure AppQoE Action screen, type or select values for the parameters. The

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

contents of the dialog box correspond to the parameters described in "Parameters for configuring the AppQoE Action" as follows (asterisk indicates a required parameter):

- Name—name
- Action type—respondWith
- Priority—priority
- Policy Queue Depth—polqDepth
- Queue Depth—priqDepth
- DOS Action—dosAction

4. Click Create or OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppQoE Parameters

Oct 02, 2014

In the AppQoE parameters, you configure the session life of an AppQoE session, the file name of the file containing the customized response, and the number of client connections that can be placed in a queue.

To configure the AppQoE parameter settings by using the command line

At the command prompt, type the following commands:

- set appqoe parameter [-sessionLife <secs>] [-avgwaitingclient <positive_integer>] [-MaxAltRespBandWidth <positive_integer>] [-dosAttackThresh <positive_integer>]
- show appqoe parameter

Parameters for configuring the AppQoE parameters

sessionLife

Number of seconds to wait after displaying alternate content before the ADC displays the same content again. Default value: 300 Maximum Minimum value: 1 Maximum value: 4,294,967,294

avgwaitingclient

The average number of client requests that can be in the service waiting queue. Default value: 1000000 Maximum value: 4,294,967,294

MaxAltRespBandWidth

The maximum bandwidth to consume when sending alternate responses. If the maximum is reached, the ADC quits sending the alternate content til bandwidth consumption drops. Default value: 100 Minimum value: 1 Maximum value: 4,294,967,294

dosAtckThrsh

The denial-of-service attack threshold. The number of connections that must be waiting in queues before the ADC responds with DoS protection measures. Default value: 2000 Minimum value: 0 Maximum value: 4,294,967,294

To configure the AppQoE parameter settings by using the configuration utility

1. Navigate to AppExpert > AppQoE.
2. In the details pane, click Configure AppQoE Parameters.
3. In the Configure AppQoE params screen, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring the AppQoE Parameters" as follows (asterisk indicates a required parameter):
   - Session Life (secs)—sessionLife
   - Average waiting client—avgwaitingclient
   - Alternate Response Bandwidth Limit(Mbps) —MaxAltRespBandWidth
   - DOS Attack Threshold —dosAttackThresh
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppQoE Policies

Oct 02, 2014

To implement AppQoE, you must configure at least one policy to tell your NetScaler ADC how to distinguish the connections to be queued in a specific queue.

To configure an AppQoE policy by using the command line

At the command prompt, type the following command:
add appqoe policy <name> -rule <expression> -action <string>

Example

The following example selects requests with a User-Agent header that contains "Android", and assigns them to the medium priority queue. These requests come from smartphones and tablets that run the Google Android operating system.

```
> add appqoe action appqoe-act-primd -priority MEDIUM
Done
> add appqoe policy appqoe-pol-primd -rule "HTTP.REQ.HEADER(\"User-Agent\").CONTAINS(\"Android\")" -action appqoe-act-primd
Done
> sh appqoe policy appqoe-pol-primd
    Name: appqoe-pol-primd
    Rule: HTTP.REQ.HEADER("User-Agent").CONTAINS("Android")
    Action: appqoe-act-primd
    Hits: 0

Done
```

Parameters for configuring an AppQoE policy

name

A name for the AppQoE policy. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the type of action.

rule

A NetScaler expression that tells the appliance which connections it should handle. For complete information about policy expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at .

action

The AppQoE action to perform when a connection matches the policy.

To configure an AppQoE policy by using the configuration utility

1. Navigate to App-Expert > AppQoE > Policies.
2. In the details pane, do one of the following:
   - To create a new policy, click Add.
   - To modify an existing policy, select the policy, and then click Edit.
3. If you are creating a new policy, in the Create AppQoE Policy dialog, in the Name text box, type a name for your new policy. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the purpose and effect of this policy.

   If you are modifying an existing policy, skip this step. You cannot change the name of an existing policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. In the Action drop-down list, choose the AppQoE action to perform when the policy matches a connection. Click the plus (+) to open the Add AppQoE Action dialog and add a new action.
5. In the Rule text box, either enter the policy expression directly, or click New to create a policy expression. If you click New, perform the following steps:
   1. In the Create Expression dialog box, click Add.
   2. In the Add Expression dialog box, select a common expression from the Frequently Used Expressions drop-down list, or use the Construct Expression drop-down lists to create the expression that defines which traffic to filter.
      If you choose to create your own expression, you start by selecting the first term from the first drop-down list on the left side of the Construct Expression area. The choices in that list are:

      - HTTP: All traffic to port 80 and port 443.
      - SYS:
      - CLIENT:
      - SERVER:
      - ANALYTICS:
      - TEXT:

      The default choice is HTTP. After you make a choice in the first drop-down list (or accept the default), you can choose the next term in your expression from the drop-down list to the right of it. The terms in that list and other lists that follow change depending on your previous choices; the lists offer only terms that are valid choices. Continue to select terms until you have finished the expression.

      Use the Help and Preview Expression areas for assistance when creating the expression. For a complete description of the available choices, see the *Citrix NetScaler Policy Configuration and Reference Guide* at .

   3. When you have created the expression that you want, click OK. The expression is added in the Expression text box.
6. Click Create. The expression appears in the Rule text box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity Templates

May 21, 2015

An entity template is a collection of configuration information for an individual entity on a Citrix® NetScaler® appliance. It provides a specification and a set of defaults for a configurable NetScaler entity, such as a policy, virtual server, service, or action. By using a template that defines a set of defaults, you can quickly configure multiple entities that require a similar configuration while eliminating several configuration steps.

Entity templates are available only in the configuration utility. You use the NetScaler configuration utility to create, manage, and use any type of entity template. You can share entity templates with other administrators and manage local folders that contain the templates. You can also import entity templates from and export entity templates to your local computer.

Before creating a template, you should be familiar with the configuration of the entity.

Note: You use entity templates to configure individual entities. To configure multiple entities related to a particular Web application, you must use an application template. For more information, see "AppExpert Applications and Templates."

How Entity Templates Work

When you create a template for a NetScaler entity, you specify default values for the entity. You specify what values must be read-only, what values must not be displayed, and what values users can configure. You also configure the pages that compose the template import wizard. All the information and settings you provide are stored in the template file.

When a user imports the entity template to a NetScaler appliance, a wizard guides the user through the various pages that you configured for the template. The wizard displays the read-only parameter values and prompts the user to specify values for the configurable parameters. After the user follows the instructions in the wizard, the appliance creates the entity with the configured values.

For example, you can create an entity template for HTTP services that provides a text box for a service name and assigns preset values for the service protocol, timeouts, thresholds, and monitors. Later, when you use the template to create new HTTP services, a wizard prompts you for a service name and supplies the preset values that you would otherwise have configured manually.

The procedure for creating entity templates for load balancing virtual servers is different than the AppExpert procedure for creating other entity templates. For more information, see "Creating an Entity Template."

In addition, the procedure for using the template to create the load balancing virtual server entity is different. For more information, see "Creating an Entity from a Template."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring an Entity Template

Mar 28, 2012

You can create or modify an entity template either from the AppExpert feature node or from the associated NetScaler feature node for the entity. For example, you can create a content switching virtual server entity template in either the AppExpert feature node or the content switching feature node in the configuration utility.

If you create a template that is not based on an existing entity, you can specify the following options and settings for the template:

- The default value of a parameter.
- Whether the default values are visible to users.
- Whether the default values can be changed by users.
- The number of pages in the entity import wizard, including the page names, text, and available parameters.
- The entities that must be bound to the entity for which the template is being created.

  For example, when you are creating a cache redirection virtual server template, you can specify the policies that you want to bind to the cache redirection virtual servers that you create from the template. However, only binding information is included in the template. The bound entities are not included. If the entity template is imported to another NetScaler appliance, the bound entities must exist on the appliance at import time for the binding to succeed. If none of the bound entities exist on the target appliance, the entity (for which the template was configured) is created without any bindings. If only a subset of the bound entities exist on the target appliance, they are bound to the entity that is created from the template.

When you create a template based on an existing entity, the configuration settings of the entity appear in the template. All bound entities are selected by default, but you can modify bindings as necessary. As in the case of a template that is not based on an existing entity, only binding information is included and not the entities. You can either save the template with the existing configuration settings or use the settings as a basis for creating a new configuration for a template.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating an Entity Template

Aug 30, 2013

You can create entity templates in either the AppExpert node or the NetScaler feature node that corresponds to the type of entity. For example, you can create a content switching virtual server template from the entity templates tab of the AppExpert feature's Templates node or in the Content Switching node. You can also specify the parameters that you want the template to store, and specify whether you want the template import wizard to prompt the user for certain parameter values.

However, when creating load balancing virtual servers, you do not have the option of specifying parameter values that you want stored in the template. You create a load balancing virtual server template by selecting an existing load balancing virtual server and configuring any variables that you might want to create in existing parameters and bound policies. The variables can be assigned values when you create a load balancing virtual server from the template. The template stores load balancing parameters such as the virtual server's IP address and port number, bound policies, actions, and variable definitions. A deployment file is also created, automatically, from the load balancing configuration. The deployment file stores deployment-specific information, such as information about bound services, service groups, and the name-value pairs of variables. If the bound entities that are included in the template are already configured on the NetScaler appliance to which the template is imported, duplicates are created, with names that are generated automatically in a particular format. The duplicate entities are based on the parameter information stored in the entity template.

When you create a load balancing virtual server template from the AppExpert node, the template is always saved to the /nsconfig/nstemplates/entities/lb vserver/ folder. If you want to save the template to a different folder, create the template from the Virtual Servers pane in the Load Balancing node. The deployment file is created with the name with which you save the template file, but with the string _deployment appended to the name. The deployment file is saved to the /nsconfig/nstemplates/entities/lb vserver/deployment_files/ folder. For more information about deployment files for load balancing virtual server templates, see "Understanding Load Balancing Entity Templates and Deployment Files."

Note: You can use either of the first two procedures for creating any template, except for a load balancing virtual server template. For creating a load balancing virtual server template, use the third or fourth procedure.

To create an entity template by using the AppExpert feature node

1. Navigate to AppExpert > Templates.
2. In the details pane, on the Entity Templates tab, do one of the following:
   - To create a new template, click Add. In the Select the Template Type dialog box, select the template type, and then click OK.
   - To create a duplicate of an existing entity template, in the details pane, select the entity template, and then click Add.
3. In the Create...Template dialog box, follow the instructions to create a template.
   If you are creating a duplicate of an existing entity template, in the Create...Template dialog box, on the Specify Template Name page, you must change the name of the entity template.
4. Click Finish, and then click Exit.

To create an entity template by using its corresponding feature node

1. Navigate to Traffic Management, and select the feature (for example, Content Switching), and then select the entity (for example, Virtual Servers), for which you want to create the entity template.
2. At the top of the details pane, click Entity Templates, and then click Create Template.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Create...Template dialog box, follow the instructions to create a template.
4. Click Finish, and then click Exit.

To create a load balancing virtual server template from the AppExpert node

1. Navigate to AppExpert > Templates.
2. In the details pane, on the LB Templates tab, click Add.
3. In the Select Load Balancing Virtual Server dialog box, select the load balancing virtual server whose configuration you want to save to a template file, and then click OK.
4. In the Create Template dialog box, provide the following information:
   - Name. The name of the template.
     Note: The Folder field shows the location to which the template will be saved. You cannot modify the path that is displayed.
   - Configure Variables. Configure variables for the load balancing template. For more information, see "Configuring Variables in Load Balancing Virtual Server Templates."
   - Introduction Description. A description of the virtual server for which you are creating a template.
   - Summary Description. A summary of the configuration or additional instructions for other administrators, such as a description of any additional steps that need to be followed after the entity is successfully created.
   - Author. The creator of the template.
   - Major. An optional major version number of your choice, to be specified if you want to maintain versions of your template.
   - Minor. An optional minor version number of your choice, to be specified if you want to maintain minor versions of your template.
     You can maintain versions by incrementing one or both of the version numbers each time you maintain the template. The Entity Template Wizard concatenates and displays the major and minor version numbers during import. For example, if the major version number is 1 and the minor version is 1, the Entity Template Wizard displays a version number of 1.1.

5. Click OK.

To create a load balancing virtual server template from the Load Balancing Virtual Servers pane

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server on which to base the template,, and then click Create Template. You might have to click the scroll arrow at the bottom right of the pane to bring the Create Template button into view.
3. In the Create Template dialog box, provide the following information:
   - Name. The name of the template.
   - Folder. The location to which the template will be saved.
     Note: If you want to save the template to the appliance, you can save it only to the /nsconfig/nstemplates/entities/lbvserver/ directory (the path displayed by default in Folder. If you want to save the template file to a folder on your computer, click the down-arrow on the Browse button, click Local, and then select a folder.
   - Configure Variables. Configure variables for the load balancing template. For more information, see "Configuring Variables in Load Balancing Virtual Server Templates."
   - Introduction Description. A description of the virtual server for which you are creating a template.
   - Summary Description. A summary of the configuration or additional instructions for other administrators, such as a description of any additional steps that need to be followed after the entity is successfully created.
   - Author. The creator of the template.
   - Major. An optional major version number of your choice, to be specified if you want to maintain versions of your

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

template.

- Minor. An optional minor version number of your choice, to be specified if you want to maintain minor versions of your template.
  You can maintain versions by incrementing one or both of the version numbers each time you maintain the template. The Entity Template Wizard concatenates and displays the major and minor version numbers during import. For example, if the major version number is 1 and the minor version is 1, the Entity Template Wizard displays a version number of 1.1.

4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Variables in Load Balancing Virtual Server Templates

Aug 30, 2013

Load balancing virtual server templates support the declaration of variables in the configured load balancing parameters and in bound policies and actions. The ability to declare variables enables you to replace preconfigured values with values that suit the environment into which you are importing the template. The Entity Template Wizard, which appears when you import a template, includes a Specify Variable Values page on which you can specify appropriate values for the variables that are configured for the entity template. This wizard page appears only when you import a template that is configured with existing variables.

As an example, consider the following expression configured for a policy that is bound to a load balancing virtual server for which you are creating a template. The expression evaluates the value of the Accept-Language header in an HTTP request.

```
HTTP.REQ.HEADER("Accept-Language").CONTAINS("en-us")
```

If you want the value of the header to be configurable at import time, you can specify the string en-us as a variable. When importing the template, you can specify a new value for the variable on the Specify Variable Values page.

After you create a variable, you can do the following:

- Assign additional strings to an existing variable. After you create a variable for a string, you can select and assign other parts of the same or different expression to the variable. The strings you assign to a variable need not be the same. At import time, all the strings that are assigned to the variable are replaced with the value that you provide.
- View the string or strings that are assigned to the variable.
- View a list of all the entities and parameters that use the variable.

To configure variables in a load balancing virtual server template

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, right-click the virtual server that you want to export to a template file, and then click Create Template.
3. In the Create Template dialog box, modify the default template file name if required, specify the location where you want to save the template, and then click Configure Variables.
4. In the Configure Variables dialog box, click the tab that lists the entity for which you want to configure a variable, select the entity, and then click Configure Variables.
5. In the Variables for <Entity Type>: <Entity Name> dialog box, click the button next to the parameter value or expression in which you want to create a variable.
6. In the Variables for <Field Name> dialog box, do the following:
   - To create a variable, in the text box that displays the configured expression or value, select the string that you want to be configurable at import time, and then click Add. In the Create Variable dialog box, specify a name and a description for the variable, and then click Create.

     The name of the variable, its value, and the description you provided appear in the Available Variables listing in the dialog box. The name you provide will be the name of the associated field in the template import wizard, and the description will appear as alt text when the user positions the mouse pointer over the field.

   - To modify a variable, in the Available Variables list, click the variable, and then click Open. In the Create Variable dialog

© 1999-2017 Citrix Systems, Inc. All rights reserved.

box, modify the value and the description, and then click OK.

The new value that you specify will not replace the text selected in the text box that displays the configured expression or value. However, when you import the template, the new value will be displayed as the default value for the variable in the template import wizard.

- To view all the strings that are assigned to a given variable, in the Available Variables listing, click the name of the variable. The strings that are assigned to the variable are highlighted.
- To view a list of all the parameters, expressions, and actions in which the variable is used, in the Available Variables listing, click the variable whose references you want to view, and then click Show References.
- To assign a string to an existing variable, in the text box that displays the expression you configured, select the string you want to assign to an existing variable, right-click the selection, click Use existing Variable, and then click the name of the variable to which you want to assign the string.

  If a variable has multiple strings assigned to it, when you specify a new value for the variable during import, all strings assigned to the variable are replaced with the new value.

7. Click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Modifying an Entity Template

Aug 30, 2013

You can modify only the parameters, bindings, and pages configured for a template. The name and location of the template specified when the template was created cannot be changed. The NetScaler appliance does not provide you with the option of modifying a load balancing virtual server template.

To modify an entity template by using the AppExpert feature node

1. Navigate to AppExpert > Templates.
2. In the details pane, on the Entity Templates tab, select the template you want to change, and then click Open.
3. In the Modify...Template dialog box, follow the instructions to modify a template.
4. Click Finish, and then click Exit.

To modify an entity template by using its corresponding feature node

1. Navigate to Traffic Management, select the feature (for example, Content Switching), and then select the entity (for example, Virtual Servers) for which you want to modify the entity template.
2. At the top of the details pane, click Entity Templates, and then click Manage Template.
3. In the Manage <feature entity name> Entity Templates dialog box, select the template that you want to modify, and then click Modify.
4. In the Modify <template name> Template dialog box, follow the instructions to modify a template.
5. Click Finish, and then click Exit.
6. Click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Deleting an Entity Template

Aug 30, 2013

Deleting an entity template does not affect any objects that have been created by using the template. You can delete a load balancing virtual server template only from the AppExpert feature node.

To delete an entity template by using the AppExpert feature node

1. Navigate to AppExpert > Templates.
2. In the details pane, on the Entity Templates tab, click the template you want to delete, and then click Remove.

To delete an entity template by using its corresponding feature node

1. Navigate to Traffic Management, and select the feature (for example, Content Switching) and then select the entity (for example, Virtual Servers), for which you want to delete the entity template.
2. At the top of the details pane, click Entity Templates, and then click Manage Template.
3. In the Manage...Entity Templates dialog box, select the template that you want to delete, and then click Delete.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating an Entity from a Template

Aug 30, 2013

You can create an entity from an entity template either from the AppExpert feature node in the NetScaler configuration utility or from the NetScaler feature node that corresponds to the type of entity that you want to create. For example, you can create a content switching virtual server from a template with either the AppExpert feature node or the content switching feature node in the configuration utility.

The procedure for creating a load balancing virtual server from a template is different than the AppExpert procedure for creating other entities from templates.

After you create an instance of an entity using an entity template, you can configure it in the same way that you would any other object of that type, such as by using the configuration utility or the command line.

To create an entity from a template by using the AppExpert feature node

1. Navigate to AppExpert > Templates.
2. In the details pane, do one of the following:
   1. To create any entity other than a load balancing virtual server from a template, on the Entity Templates tab, click the template that you want to use, and then click Use Template.
   2. To create a load balancing virtual server from a template, on the LB Templates tab, click the template that you want to use, and then click Use Template.
3. In the <Entity Template Name> wizard, follow the instructions to create the entity on the NetScaler.
4. Click Finish, and then click Exit.

To create an entity from a template by using its corresponding feature node

1. Navigate to Traffic Management, and expand a feature node (for example, Content Switching), and then click an entity subnode (for example, Virtual Servers).
2. At the top of the details pane, click Entity Templates, and then click Use Template.
3. Click the name of the template that you want to use.
4. In the Use <template name> Template wizard, follow the instructions to create the entity.
   Only templates that match the current context are displayed. For example, in the details pane for content switching virtual servers, only entity templates for content switching virtual servers appear, if configured.

5. Click Finish, and then click Exit.

To create a load balancing virtual server by using a load balancing virtual server template

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Use Template.
3. In the Entity Template Wizard, follow the instructions to create a load balancing virtual server on the NetScaler.
   Only templates that match the current context are displayed. For example, when you click Browse (Appliance), only entity templates for load balancing virtual servers appear, if configured.

4. Click Finish, and then click Exit.
   Note: The Entity Template Wizard includes a Specify Variable Values page on which you can specify new values for variables. For more information about configuring variables in load balancing virtual server templates, see "Configuring Variables in Load Balancing Virtual Server Templates."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Entity Template Folders

Aug 30, 2013

You can organize only load balancing virtual server template folders.

To organize load balancing virtual server template folders

1. Navigate to AppExpert > Templates > LB Templates.
2. In the Manage LB Templates dialog box, do one of the following:
   - To change the name of a folder, select the folder and click Rename.

     You can also click the folder that you want to rename, and then press F2. You cannot rename the top-level default folder.

   - To remove the folder, select the folder and click Delete.

     You can also click the folder that you want to remove, and then press the Delete key. You cannot remove the top-level default folder.

3. Click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Uploading and Downloading Entity Templates

Aug 30, 2013

You can import the entity templates that are stored on your local computer. You can also download entity templates from the NetScaler appliance to your local computer and then import them to other NetScaler appliances.

Note: You cannot upload or download load balancing virtual server templates.

To upload an entity template to the NetScaler appliance

1. Navigate to Traffic Management, and expand a feature node (for example, Content Switching), and then click a subnode (for example, Virtual Servers) for which you want to upload an entity template.
2. At the top of the details pane, click Entity Templates, and then click Manage Template.
3. In the Manage...Entity Templates dialog box, click the top-level folder, and then click Upload.
4. In the Upload Entity Template dialog box, navigate to the template file that you want to upload, and then click Select.
5. Click Close.

To download an entity template from the NetScaler appliance

1. Navigate to Traffic Management, and expand a feature node (for example, Content Switching), and then click a subnode (for example, Virtual Servers) for which you want to upload an entity template.
2. At the top of the details pane, click Entity Templates, and then click Manage Template.
3. In the Manage...Entity Templates dialog box, click the template that you want to download, and then click Download.
4. In the Download Entity Template dialog box, navigate to the location at which you want to save the template on your local computer, enter a file name, and then click Save.
5. Click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding Load Balancing Entity Templates and Deployment Files

Mar 28, 2012

Load balancing entity templates are created in the same way that NetScaler application templates are created. When you export a load balancing virtual server to a template file, the following two files are automatically created:

- **Load balancing virtual server template file.** Contains XML elements that store the values of the parameters that are configured for the load balancing virtual server. The file also contains XML elements for storing information about bound policies.
- **Deployment file.** Contains XML elements that store deployment-specific information such as services, service groups, and configured variables.

In the template and deployment files, each unit of configuration information is encapsulated in a specific XML element that is meant for that unit type. For example, the load balancing method parameter, lbMethod, is encapsulated within the <lbmethod> and </lbmethod> tags.

Note: After you export a load balancing virtual server, you can add elements, remove elements, and modify existing elements before importing the configuration information to a NetScaler appliance.

Example of a Load Balancing Virtual Server Template

Following is an example of a template file that was created from a load balancing virtual server called "Lbvip":

```
<?xml version="1.0" encoding="UTF-8" ?>
 <template>
  <template_info>
   <entity_name>Lbvip</entity_name>
   <version_major>10</version_major>
   <version_minor>0</version_minor>
   <build_number>40.406</build_number>
  </template_info>
  <entitytemplate>
   <lbvserver_list>
    <lbvserver>
     <name>Lbvip</name>
     <servicetype>HTTP</servicetype>
     <ipv46>0.0.0.0</ipv46>
     <ipmask>*</ipmask>
     <port>0</port>
     <range>1</range>
     <persistencetype>NONE</persistencetype>
     <timeout>2</timeout>
     <persistencebackup>NONE</persistencebackup>
     <backuppersistencetimeout>2</backuppersistencetimeout>
     <lbmethod>LEASTCONNECTION</lbmethod>
     <persistmask>255.255.255.255</persistmask>
     <v6persistmasklen>128</v6persistmasklen>
     <pq>OFF</pq>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        <sc>OFF</sc>
        <m>IP</m>
        <datalength>0</datalength>
        <dataoffset>0</dataoffset>
        <sessionless>DISABLED</sessionless>
        <state>ENABLED</state>
        <connfailover>DISABLED</connfailover>
        <clttimeout>180</clttimeout>
        <somethod>NONE</somethod>
        <sopersistence>DISABLED</sopersistence>
        <sopersistencetimeout>2</sopersistencetimeout>
        <redirectportrewrite>DISABLED</redirectportrewrite>
        <downstateflush>DISABLED</downstateflush>
        <gt2gb>DISABLED</gt2gb>
        <ipmapping>0.0.0.0</ipmapping>
        <disableprimaryondown>DISABLED</disableprimaryondown>
        <insertvserveripport>OFF</insertvserveripport>
        <authentication>OFF</authentication>
        <authn401>OFF</authn401>
        <push>DISABLED</push>
        <pushlabel>none</pushlabel>
        <l2conn>OFF</l2conn>
        <appflowlog>DISABLED</appflowlog>
        <icmpvsrresponse>PASSIVE</icmpvsrresponse>
        <lbvserver_cmppolicy_binding_list>
          <lbvserver_cmppolicy_binding>
            <name>Lbvip</name>
            <policyname>NOPOLICY-COMPRESSION</policyname>
            <priority>100</priority>
            <gotopriorityexpression>END</gotopriorityexpression>
            <bindpoint>REQUEST</bindpoint>
          </lbvserver_cmppolicy_binding>
        </lbvserver_cmppolicy_binding_list>
      </lbvserver>
    </lbvserver_list>
  </entitytemplate>
 </template>
```


Example of a Deployment File

Following is the deployment file associated with the virtual server in the preceding example:


```
<?xml version="1.0" encoding="UTF-8" ?>
 <template_deployment>
  <template_info>
    <entity_name>Lbvip</entity_name>
    <version_major>10</version_major>
    <version_minor>0</version_minor>
    <build_number>40.406</build_number>
  </template_info>
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
  <service_list>
    <service>
    <ip>1.2.3.4</ip>
    <port>80</port>
    <servicetype>HTTP</servicetype>
    </service>
  </service_list>
  <servicegroup_list>
    <servicegroup>
      <name>svcgrp</name>
      <servicetype>HTTP</servicetype>
      <servicegroup_servicegroupmember_binding_list>
        <servicegroup_servicegroupmember_binding>
          <ip>1.2.3.90</ip>
          <port>80</port>
        </servicegroup_servicegroupmember_binding>
      <servicegroup_servicegroupmember_binding>
        <ip>1.2.8.0</ip>
        <port>80</port>
      </servicegroup_servicegroupmember_binding>
      <servicegroup_servicegroupmember_binding>
        <ip>1.2.8.1</ip>
        <port>80</port>
      </servicegroup_servicegroupmember_binding>
      <servicegroup_servicegroupmember_binding>
        <ip>1.2.9.0</ip>
        <port>80</port>
      </servicegroup_servicegroupmember_binding>
    </servicegroup_servicegroupmember_binding_list>
  </servicegroup>
 </servicegroup_list>
</template_deployment>
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Callouts

Jan 11, 2013

For certain types of requests, or when certain criteria are met during policy evaluation, you might want to stall policy evaluation briefly, retrieve information from a server, and then perform a specific action that depends on the information that is retrieved. At other times, when you receive certain types of requests, you might want to update a database or the content hosted on a Web server. HTTP callouts enable you to perform all these tasks.

An HTTP callout is an HTTP or HTTPS request that the NetScaler appliance generates and sends to an external application when certain criteria are met during policy evaluation. The information that is retrieved from the server can be analyzed by default syntax policy expressions, and an appropriate action can be performed. You can configure HTTP callouts for HTTP content switching, TCP content switching, rewrite, responder, and for the token-based method of load balancing.

Before you configure an HTTP callout, you must set up an application on the server to which the callout will be sent. The application, which is called the *HTTP callout agent* , must be configured to respond to the HTTP callout request with the required information. The HTTP callout agent can also be a Web server that serves the data for which the NetScaler appliance sends the callout. You must make sure that the format of the response to an HTTP callout does not change from one invocation to another.

After you set up the HTTP callout agent, you configure the HTTP callout on the NetScaler appliance. Finally, to invoke the callout, you include the callout in a default syntax policy in the appropriate NetScaler feature and then bind the policy to the bind point at which you want the policy to be evaluated.

After you have configured the HTTP callout, you must verify the configuration to make sure that the callout is working correctly.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How an HTTP Callout Works

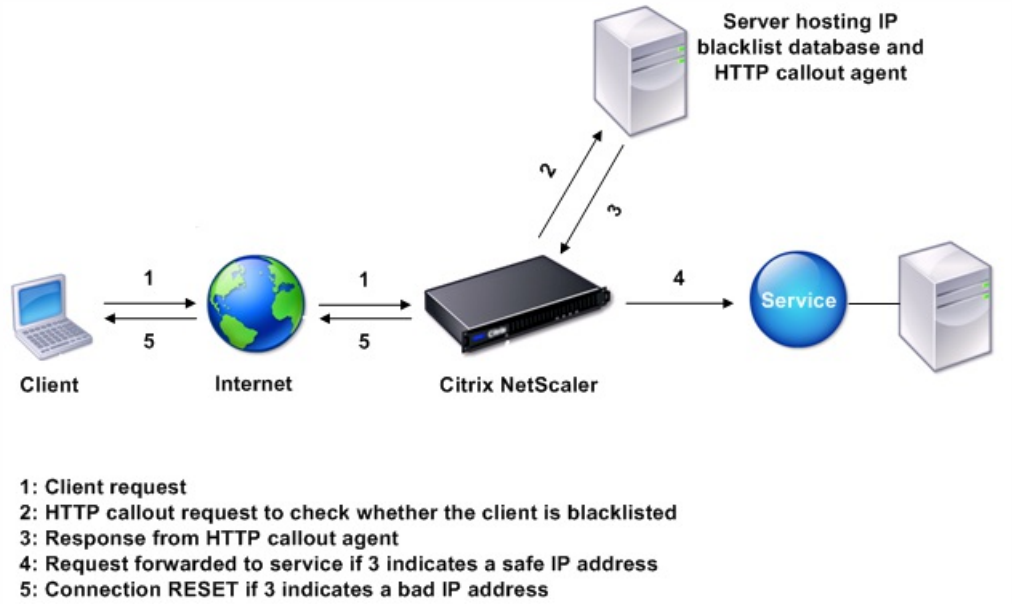
Jan 11, 2013

When the NetScaler appliance receives a client request, the appliance evaluates the request against the policies bound to various bind points. During this evaluation, if the appliance encounters the HTTP callout expression, SYS.HTTP_CALLOUT(<name>), it stalls policy evaluation briefly and sends a request to the HTTP callout agent by using the parameters configured for the specified HTTP callout. Upon receiving the response, the appliance inspects the specified portion of the response, and then either performs an action or evaluates the next policy, depending on whether the evaluation of the response from the HTTP callout agent evaluates to TRUE or FALSE, respectively. For example, if the HTTP callout is included in a responder policy, if the evaluation of the response evaluates to TRUE, the appliance performs the action associated with the responder policy.

If the HTTP callout configuration is incorrect or incomplete, or if the callout invokes itself recursively, the appliance raises an UNDEF condition, and updates the undefined hits counter.

The following figure illustrates the working of an HTTP callout that is invoked from a globally bound responder policy. The HTTP callout is configured to include the IP address of the client that is associated with an incoming request. When the NetScaler appliance receives a request from a client, the appliance generates the callout request and sends it to the callout server, which hosts a database of blacklisted IP addresses and an HTTP callout agent that checks whether the client's IP address is listed in the database. The HTTP callout agent receives the callout request, checks whether the client's IP address is listed, and sends a response that the NetScaler appliance evaluates. If the response indicates that the client's IP address is not blacklisted, the appliance forwards the response to the configured service. If the client's IP address is blacklisted, the appliance resets the client connection

Figure 1. HTTP Callout Entity Model



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Notes on the Format of HTTP Requests and Responses

May 21, 2015

The NetScaler appliance does not check for the validity of the HTTP callout request. Therefore, before you configure HTTP callouts, you must know the format of an HTTP request. You must also know the format of an HTTP response, because configuring an HTTP callout involves configuring expressions that evaluate the response from the HTTP callout agent.

This document includes the following details:

- Format of an HTTP Request
- Format of an HTTP Response

Format of an HTTP Request

An HTTP request contains a series of lines that each end with a carriage return and a line feed, represented as either <CR><LF> or \r\n.

The first line of a request (the *message line*) contains the HTTP method and target. For example, a message line for a GET request contains the keyword GET and a string that represents the object that is to be fetched, as shown in the following example:

```
GET /mysite/mydirectory/index.html HTTP/1.1\r\n
```

The rest of the request contains HTTP headers, including a required Host header and, if applicable, a message body.

The request ends with a bank line (an extra <CR><LF> or \r\n).

Following is an example of a request:

```
Get /mysite/index.html HTTP/1.1\r\n
Host: 10.101.101.10\r\n
Accept: */*\r\n
\r\n
```

Format of an HTTP Response

An HTTP response contains a status message, response HTTP headers, and the requested object or, if the requested object cannot be served, an error message.

Following is an example of a response:

```
HTTP/1.1 200 OK\r\n
Content-Length: 55\r\n
Content-Type: text/html\r\n
Last-Modified: Wed, 12 Aug 1998 15:03:50 GMT\r\n
Accept-Ranges: bytes\r\n
ETag: “04f97692cbd1:377”\r\n
Date: Thu, 19 Jun 2008 19:29:07 GMT\r\n
\r\n
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

<55-character response>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring an HTTP Callout

Dec 17, 2013

When configuring an HTTP callout, you specify the type of request (HTTP or HTTPS), destination and format of the request, the expected format of the response, and, finally, the portion of the response that you want to analyze.

For the destination, you either specify the IP address and port of the HTTP callout agent or engage a load balancing, content switching, or cache redirection virtual server to manage the HTTP callout requests. In the first case, the HTTP callout requests will be sent directly to the HTTP callout agent. In the second case, the HTTP callout requests will be sent to the virtual IP address (VIP) of the specified virtual server. The virtual server will then process the request in the same way as it processes a client request. For example, if you expect a large number of callouts to be generated, you can configure instances of the HTTP callout agent on multiple servers, bind these instances (as services) to a load balancing virtual server, and then specify the load balancing virtual server in the HTTP callout configuration. The load balancing virtual server then balances the load on those configured instances as determined by the load balancing algorithm.

For the format of the HTTP callout request, you can specify the individual attributes of the HTTP callout request (an attribute-based HTTP callout), or you can specify the entire HTTP callout request as a default syntax expression (an expression-based HTTP callout).

Note: The appliance does not check for the validity of the request. You must make sure that the request is a valid request. An incorrect or incomplete HTTP callout configuration results in a runtime UNDEF condition that is not associated with an action. The UNDEF condition merely updates the Undefined Hits counter, which enables you to troubleshoot an incorrectly configured HTTP callout. However, the appliance parses the HTTP callout request to enable you to configure certain NetScaler features for the callout. This can lead to an HTTP callout invoking itself. For information about callout recursion and how you can avoid it, see "Avoiding HTTP Callout Recursion."

Finally, regardless of whether you use HTTP request attributes or an expression to define the format of the HTTP callout request, you must specify the format of the response from the HTTP callout agent and the portion of the response that you want to evaluate. The response can be a Boolean value, a number, or text. The portion of the response that you want to evaluate is specified by an expression. For example, if you specify that the response contains text, you can use HTTP.RES.BODY(<unit>) to specify that the appliance must evaluate only the first <unit> bytes of the response from the callout agent.

At the command line, you first create an HTTP callout by using the add command. When you add a callout, all parameters are set to a default value of NONE, except the HTTP method, which is set to a default value of GET. You then configure the callout's parameters by using the set command. The set command is used to configure both types of callouts (attribute-based and expression-based). The difference lies in the parameters that are used for configuring the two types of callouts. Accordingly, the command-line instructions that follow include a set command for configuring an attribute-based callout and a set command for configuring an expression-based callout. In the configuration utility, all of these configuration tasks are performed in a single dialog box.

Note: Before you put an HTTP callout into a policy, you can modify all configured parameters except the return type. Once an HTTP callout is in a policy, you cannot completely modify an expression that is configured in the callout. For example, you cannot change HTTP.REQ.HEADER("myval") to CLIENT.IP.SRC. However, you can modify the operators and arguments that are passed to the expression. For example, you can change HTTP.REQ.HEADER("myVal1") to HTTP.REQ.HEADER("myVal2"), or HTTP.REQ.HEADER("myVal") to HTTP.REQ.HEADER("myVal").AFTER_STR(<string>). If the set command fails, create a new HTTP callout.

HTTP callout configuration involves configuring default syntax expressions. For more information about configuring default syntax expressions, see "Configuring Default Syntax Expressions: Getting Started."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure an HTTP callout by using the command line interface

At the command prompt, do the following:

1. Create a HTTP callout.
   add policy httpCallout <name>

   Example
   > add policy httpCallout mycallout
2. Configure the details of the HTTP callout.
   - To configure an attribute-based HTTP callout, type:
     set policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr|*>] [-port <port|*>] [-vServer <string>] [-returnType <returnType>] [-httpMethod (GET | POST)] [-hostExpr <string>] [-urlStemExpr <string>] [-headers <name(value)> ...] [-parameters <name(value)> ...] [-resultExpr <string>]

     Example

     > set policy httpCallout mycallout -vserver lbv1 -returnType num -httpMethod GET -hostExpr 'http.req.header("Host")' -urlStemExpr "http.req.url" -parameters Name("My Name") -headers Name("MyHeader") -resultExpr "http.res.body(10000).length"
   - To configure an expression-based HTTP callout, type:
     set policy httpCallout <name> [-vServer <string>] [-returnType <returnType>] [-httpMethod (GET | POST)] [-fullReqExpr <string>] [-resultExpr <string>]

     Example

     > set policy httpCallout mycallout1 -vserver lbv1 -returnType num -httpMethod GET -fullReqExpr q{"GET " + http.req.url + "HTTP/" + http.req.version.major + "." + http.req.version.minor.sub(1)+ "r\nHost:10.101.10.10\r\nAccept: */*\r\n\r\n"}
3. Verify the configurations of the HTTP callout.
   show policy httpCallout <name>

To configure an HTTP callout by using the configuration utility

1. Navigate to AppExpert > HTTP Callouts.
2. In the details pane, click Add.
3. In the Create HTTP Callout dialog box, configure the parameters of the HTTP callout. For a description of the parameter, hover the mouse cursor over the check box.
4. Click Create and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Verifying the Configuration

Aug 30, 2013

For an HTTP callout to work correctly, all the HTTP callout parameters and the entities associated with the callout must be configured correctly. While the NetScaler appliance does not check the validity of the HTTP callout parameters, it indicates the state of the bound entities, namely the server or virtual server to which the HTTP callout is sent. The following table lists the icons and describes the conditions under which the icons are displayed.

Table 1. Icons That Indicate the States of Entities Bound to an HTTP Callout

Icon	Indicates that
	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is UP.
	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is OUT OF SERVICE.
	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is DOWN.

For an HTTP callout to function correctly, the icon must be green at all times. If the icon is not green, check the state of the callout server or virtual server to which the HTTP callout is sent. If the HTTP callout is not working as expected even though the icon is green, check the parameters configured for the callout.

You can also verify the configuration by sending test requests that match the policy from which the HTTP callout is invoked, checking the hits counter for the policy and the HTTP callout, and verifying the responses that the NetScaler appliance sends to the client.

Note: An HTTP callout can sometimes invoke itself recursively a second time. If this happens, the hits counter is incremented by two counts for each callout that is generated by the appliance. For the hits counter to display the correct value, you must configure the HTTP callout in such a way that it does not invoke itself a second time. For more information about how you can avoid HTTP callout recursion, see "Avoiding HTTP Callout Recursion."

To view the hits counter for an HTTP callout

1. Navigate to AppExpert > HTTP Callouts.
2. In the details pane, click the HTTP callout for which you want to view the hits counter, and then view the hits in the Details area.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Invoking an HTTP Callout

Aug 30, 2013

After you configure an HTTP callout, you invoke the callout by including the SYS.HTTP_CALLOUT(<name>) expression in a default syntax policy rule. In this expression, <name> is the name of the HTTP callout that you want to invoke.

You can use default syntax expression operators with the callout expression to process the response and then perform an appropriate action. The return type of the response from the HTTP callout agent determines the set of operators that you can use on the response. If the part of the response that you want to analyze is text, you can use a text operator to analyze the response. For example, you can use the CONTAINS(<string>) operator to check whether the specified portion of the response contains a particular string, as in the following example:

SYS.HTTP_CALLOUT(mycallout).contains("Good IP address")

If you use the preceding expression in a responder policy, you can configure an appropriate responder action.

Similarly, if the part of the response that you want to evaluate is a number, you can use a numeric operator such as GT(int). If the response contains a Boolean value, you can use a Boolean operator.

Note: An HTTP callout can invoke itself recursively. HTTP callout recursion can be avoided by combining the HTTP callout expression with a default syntax expression that prevents recursion. For information about how you can avoid HTTP callout recursion, see "Avoiding HTTP Callout Recursion."

You can also cascade HTTP callouts by configuring policies that each invoke a callout after evaluating previously generated callouts. In this scenario, after one policy invokes a callout, when the NetScaler appliance is parsing the callout before sending the callout to the callout server, a second set of policies can evaluate the callout and invoke additional callouts, which can in turn be evaluated by a third set of policies, and so on. Such an implementation is described in the following example.

First, you could configure an HTTP callout called myCallout1, and then configure a responder policy, Pol1, to invoke myCallout1. Then, you could configure a second HTTP callout, myCallout2, and a responder policy, Pol2. You configure Pol2 to evaluate myCallout1 and invoke myCallout2. You bind both responder policies globally.

To avoid HTTP callout recursion, myCallout1 is configured with a unique custom HTTP header called "Request1." Pol1 is configured to avoid HTTP callout recursion by using the default syntax expression, HTTP.REQ.HEADER(\"Request1\").EQ(\"Callout Request\").NOT.

Pol2 uses the same default syntax expression, but excludes the .NOT operator so that the policy evaluates myCallout1 when the NetScaler appliance is parsing it. Note that myCallout2 identifies its own unique header called "Request2," and Pol2 includes a default syntax expression to prevent myCallout2 from invoking itself recursively.

Example

```
> add policy httpCallout myCallout1

Done

> set policy httpCallout myCallout1 -IPAddress 10.102.3.95 -port 80 -returnType TEXT -hostExpr
 "\"10.102.3.95\"" -urlStemExpr "\"/cgi-bin/check_clnt_from_database.pl\"" -headers Request1
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(100)"

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> add responder policy Pol1 "HTTP.REQ.HEADER(\"Request1\").EQ(\"Callout Request\").NOT &&
SYS.HTTP_CALLOUT(myCallout1).CONTAINS(\"IP Matched\")" RESET

Done

> bind responder global Pol1 100 END -type OVERRIDE

Done

> add policy httpCallout myCallout2

Done

> set policy httpCallout myCallout2 -IPAddress 10.102.3.96 -port 80 -returnType TEXT -hostExpr
"\"10.102.3.96\"" -urlStemExpr "\"/cgi-bin/check_clnt_location_from_database.pl\"" -headers Request2
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(200)"

 Done

> add responder policy Pol2 "HTTP.REQ.HEADER(\"Request2\").EQ(\"Callout Request\").NOT &&
 HTTP.REQ.HEADER(\"Request1\").EQ(\"Callout Request\") && SYS.HTTP_CALLOUT(myCallout2).CONTAINS
(\"APAC\")" RESET

 Done

> bind responder global Pol2 110 END -type OVERRIDE

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Avoiding HTTP Callout Recursion

Apr 03, 2013

Even though the NetScaler appliance does not check for the validity of the HTTP callout request, it parses the request once before it sends the request to the HTTP callout agent. This parsing allows the appliance to treat the callout request as any other incoming request, which in turn allows you to configure several useful NetScaler features (such as integrated caching, SureConnect, and Priority Queuing) to work on the callout request.

However, during this parsing, the HTTP callout request can hit the same policy and therefore invoke itself recursively. The appliance detects the recursive invocation and raises an undefined (UNDEF) condition. However, the recursive invocation results in the policy and HTTP callout hit counters being incremented by two counts each instead of one count each.

To prevent a callout from invoking itself, you must identify at least one unique characteristic of the HTTP callout request, and then exclude all requests with this characteristic from being processed by the policy rule that invokes the callout. You can do so by including another default syntax expression in the policy rule. The expression must precede the SYS.HTTP_CALLOUT(<name>) expression so that it is evaluated before the callout expression is evaluated. For example:
<Expression that prevents callout recursion> && SYS.HTTP_CALLOUT(<name>)
When you configure a policy rule in this way, when the appliance generates the request and parses it, the compound rule evaluates to FALSE, the callout is not generated a second time, and the hit counters are incremented correctly.

One way by which you can assign a unique characteristic to an HTTP callout request is to include a unique custom HTTP header when you configure the callout. Following is an example of an HTTP callout called "myCallout." The callout generates an HTTP request that checks whether a client's IP address is present in a database of blacklisted IP addresses. The callout includes a custom header called "Request," which is set to the value "Callout Request." A globally bound responder policy, "Pol1," invokes the HTTP callout but excludes all requests whose Request header is set to this value, thus preventing a second invocation of myCallout. The expression that prevents a second invocation is HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT.

Example

```
> add policy httpCallout myCallout
 Done

> set policy httpCallout myCallout -IPAddress 10.102.3.95 -port 80 -returnType TEXT -hostExpr "\"10.102.3.95\"" -urlStemExpr "\"/cgi-bin/check_clnt_from_database.pl\"" -headers Request("Callout Reque
Done

> add responder policy Pol1 "HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.HTTP_CALLOUT(myCallout).CONTAINS(\"IP Matched\")" RESET
Done

> bind responder global Pol1 100 END -type OVERRIDE
Done
```

Note: You can also configure an expression to check whether the URL of the request includes the URL stem expression that is configured for the HTTP callout. If you want to implement this scenario, make sure that the HTTP callout agent is dedicated to respond only to HTTP callouts and not to other client requests directed through the appliance. If the HTTP callout agent is an application or Web server that serves other client requests, such an expression will prevent the appliance from processing those client requests. Instead, use a unique custom header as described earlier.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Caching HTTP Callout Responses

Apr 03, 2013

For improved performance while using callouts, you can use the integrated caching feature to cache callout responses. The responses are stored in an integrated caching content group named calloutContentGroup for a specified time duration.

Note: To cache callout responses, make sure that the integrated caching feature is enabled.

To set the cache duration by using the command line interface

At the command prompt, type:

set policy httpCallout <name> -cacheForSecs <secs>

Example

```
> set httpcallout httpcallout1 -cacheForSecs 120
```

To set the cache duration by using the configuration utility

1. Navigate to AppExpert > HTTP Callouts.
2. In the details pane, select the HTTP callout for which you want to set the cache duration and click Open.
3. In the Configure HTTP Callout dialog box, specify the Cache Expiration Time.
4. Verify that you have entered the correct time duration, and then click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case: Filtering Clients by Using an IP Blacklist

May 21, 2015

HTTP callouts can be used to block requests from clients that are blacklisted by the administrator. The list of clients can be a publicly known blacklist, a blacklist that you maintain for you organization, or a combination of both.

The NetScaler appliance checks the IP address of the client against the pre-configured blacklist and blocks the transaction if the IP address has been blacklisted. If the IP address is not in the list, the appliance processes the transaction.

To implement this configuration, you must perform the following tasks:

1. Enable responder on the NetScaler appliance.
2. Create an HTTP callout on the NetScaler appliance and configure it with details about the external server and other required parameters.
3. Configure a responder policy to analyze the response to the HTTP callout, and then bind the policy globally.
4. Create an HTTP callout agent on the remote server.

Enabling Responder

Updated: 2013-08-30

You must enable responder before you can use it.

To enable responder by using the configuration utility

1. Make sure that you have installed the responder license.
2. In the configuration utility, expand AppExpert, and right-click Responder, and then click Enable Responder feature.

Creating an HTTP Callout on the NetScaler Appliance

Updated: 2013-08-30

Create an HTTP callout, HTTP-Callout-1, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see "Configuring an HTTP Callout."

Table 1. Parameters and Values for HTTP-Callout-1

Parameter	Value
Name	HTTP-Callout-1
Server to receive callout request	
IP Address	10.103.9.95
Port	80
Request to send to the server	
Method	GET
Host Expression	10.102.3.95
URL Stem Expression	"/cgi-bin/check_clnt_from_database.pl"
Headers	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter	Value
Name	Request
Value-expression	Callout Request
Parameters	
Name	Cip
Value-expression	CLIENT.IP.SRC
Server Response	
Return Type	TEXT
Expression to extract data from the response	HTTP.RES.BODY(100)

Configuring a Responder Policy and Binding it Globally

Updated: 2013-08-30

After you configure the HTTP callout, verify the callout configuration, and then configure a responder policy to invoke the callout. While you can create a responder policy in the Policies sub-node and then bind it globally by using the Responder Policy Manager, this demonstration uses the Responder Policy Manager to create the responder policy and bind the policy globally.

To create a responder policy and bind it globally by using the configuration utility

1. Navigate to AppExpert > Responder.
2. In the details pane, under Policy Manager, click Policy Manager.
3. In the Responder Policy Manager dialog box, click Override Global.
4. Click Insert Policy, and then, under Policy Name, click New Policy.
5. In the Create Responder Policy dialog box, do the following:
   1. In Name, type Policy-Responder-1.
   2. In Action, select RESET.
   3. In Undefined-Result Action, select Global undefined-result action.
   4. In Expression, type the following default syntax expression:
      "HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.HTTP_CALLOUT(HTTP-Callout-1).CONTAINS(\"IP Matched\")"
   5. Click Create, and then click Close.
6. Click Apply Changes, and then click Close.

Creating an HTTP Callout Agent on the Remote Server

You must now create an HTTP callout agent on the remote callout server that will receive callout requests from the NetScaler appliance and respond appropriately. The HTTP callout agent is a script that is different for each deployment and must be written with the server specifications in mind, such as the type of database and the scripting language supported.

Following is a sample callout agent that verifies whether the given IP address is part of an IP blacklist. The agent has been written in the Perl scripting language and uses a MYSQL database.

The following CGI script checks for a given IP address on the callout server.

```
#!/usr/bin/perl -w
print "Content-type: text/html\n\n";
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    use DBI();
    use CGI qw(:standard);
#Take the Client IP address from the request query
    my $ip_to_check = param('cip');
# Where a MYSQL database is running
    my $dsn = 'DBI:mysql:BAD_CLIENT:localhost';
# Database username to connect with
    my $db_user_name = ‘dbuser’;
# Database password to connect with
    my $db_password = 'dbpassword';
    my ($id, $password);
# Connecting to the database
    my $dbh = DBI->connect($dsn, $db_user_name, $db_password);
    my $sth = $dbh->prepare(qq{ select * from bad_clnt });
    $sth->execute();
    while (my ($ip_in_database) = $sth->fetchrow_array()) {
    chomp($ip_in_database);
# Check for IP match
    if ($ip_in_database eq $ip_to_check) {
        print "\n IP Matched\n";
        $sth->finish();
        exit;
    }
   }
   print "\n IP Failed\n";
   $sth->finish();
   exit;
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case: ESI Support for Fetching and Updating Content Dynamically

May 21, 2015

Edge Side Includes (ESI) is a markup language for edge-level dynamic Web content assembly. It helps in accelerating dynamic Web-based applications by defining a simple markup language to describe cacheable and non-cacheable Web page components that can be aggregated, assembled, and delivered at the network edge. By using HTTP callouts on the NetScaler appliance, you can read through the ESI constructs and aggregate or assemble content dynamically.

To implement this configuration, you must perform the following tasks:

1. Enable rewrite on the NetScaler appliance.
2. Create an HTTP callout on the appliance and configure it with details about the external server and other required parameters.
3. Configure a rewrite action to replace the ESI content with the callout response body.
4. Configure a rewrite policy to specify the conditions under which the action is performed, and then bind the rewrite policy globally.

Enabling Rewrite

Updated: 2013-08-30

Rewrite must be enabled before it is used on the NetScaler appliance. The following procedure describes the steps to enable the rewrite feature.

To enable rewrite by using the configuration utility

1. Make sure that you have installed the rewrite license.
2. In the configuration utility, expand AppExpert, and right-click Rewrite, and then click Enable Rewrite feature.

Creating an HTTP Callout on the NetScaler Appliance

Updated: 2013-08-30

Create an HTTP callout, HTTP-Callout-2, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see "Configuring an HTTP Callout."

Table 1. Parameters and Values for HTTP-Callout-2

Parameter	Value
Name	HTTP-Callout-2
Server to receive callout request	
IP Address	10.102.56.51
Port	80

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter	Value
Request to send to the server	
Method	GET
Host Expression	10.102.56.51:80
URL Stem Expression	"HTTP.RES.BODY(500).AFTER_STR(\"src=\").BEFORE_STR(\"/>\")"
Headers	
Name	Name
Value-expression	Callout
Server Response	
Return Type	TEXT
Expression to extract data from the response	HTTP.RES.BODY(100)

Configuring the Rewrite Action

Updated: 2013-08-30

Create a rewrite action, Action-Rewrite-1, to replace the ESI content with the callout response body. Use the parameter settings shown in the following table.

Table 2. Parameters and Values for Action-Rewrite-1

Parameter	Value
Name	Action-Rewrite-1
Type	Replace
Expression to choose target text reference	"HTTP.RES.BODY(500).AFTER_STR (\" <example>\").BEFORE_STR (\" </example>\")"
String expression for replacement text	"SYS.HTTP_CALLOUT(HTTP-Callout-2)"

To configure the rewrite action by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to AppExpert > Rewrite > Actions.
2. In the details pane, click Add.
3. In the Create Rewrite Action dialog box, in Name, type Action-Rewrite-1.
4. In Type, select REPLACE.
5. In Expression to choose target text reference, type the following default syntax expression:
   "HTTP.RES.BODY(500).AFTER_STR(\"<example>\").BEFORE_STR(\"</example>\")"
6. In the String expression for replacement text, type the following string expression:
   "SYS.HTTP_CALLOUT(HTTP-Callout-2)"
7. Click Create, and then click Close.

Creating the Rewrite Policy and Binding it Globally

Updated: 2013-08-30

Create a rewrite policy, Policy-Rewrite-1, with the parameter settings shown in the following table. You can create a rewrite policy in the Policies subnode and then bind it globally by using the Rewrite Policy Manager. Alternatively, you can use the Rewrite Policy Manager to perform both these tasks simultaneously. This demonstration uses the Rewrite Policy Manager to perform both tasks.

Table 3. Parameters and Values for Policy-Rewrite-1

Parameter	Value
Name	Policy-Rewrite-1
Action	Action_Rewrite-1
Undefined Result Action	-Global undefined-result action-
Expression	"HTTP.REQ.HEADER(\"Name\").CONTAINS (\"Callout\").NOT"

To configure a rewrite policy and bind it globally by using the configuration utility

1. Navigate to AppExpert > Rewrite.
2. In the details pane, under Policy Manager, click Rewrite Policy Manager.
3. In the Rewrite Policy Manager dialog box, click Override Global.
4. Click Insert Policy, and then, in the Policy Name column, click New Policy.
5. In the Create Rewrite Policy dialog box, do the following:
   1. In Name, type Policy-Rewrite-1.
   2. In Action, select Action-Rewrite-1.
   3. In Undefined-Result Action, select Global undefined-result action.
   4. In Expression, type the following default syntax expression:
      "HTTP.REQ.HEADER(\"Name\").CONTAINS(\"Callout\").NOT"
   5. Click Create, and then click Close.
6. Click Apply Changes, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case: Access Control and Authentication

May 21, 2015

In high security zones, it is mandatory to externally authenticate the user before a resource is accessed by clients. On the NetScaler appliance, you can use HTTP callouts to externally authenticate the user by evaluating the credentials supplied. In this example, the assumption is that the client is sending the user name and password through HTTP headers in the request. However, the same information could be fetched from the URL or the HTTP body.

To implement this configuration, you must perform the following tasks:

1. Enable the responder feature on the NetScaler appliance.
2. Create an HTTP callout on the appliance and configure it with details about the external server and other required parameters.
3. Configure a responder policy to analyze the response, and then bind the policy globally.
4. Create a callout agent on the remote server.

Enabling Responder

Updated: 2013-08-30

The responder feature must be enabled before it is used on the NetScaler appliance.

To enable responder by using the configuration utility

1. Make sure that the responder license is installed.
2. In the configuration utility, expand AppExpert, and right-click Responder, and then click Enable Responder feature.

Creating an HTTP Callout on the NetScaler Appliance

Updated: 2013-08-30

Create an HTTP callout, HTTP-Callout-3, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see "Configuring an HTTP Callout."

Table 1. Parameters and Values for HTTP-Callout-3

Parameter	Value
Name	HTTP-Callout-3
Server to receive callout request	
IP Address	10.103.9.95
Port	80
Request to send to the server	
Method	GET
Host Expression	10.102.3.95
URL Stem Expression	“/cgi-bin/authenticate.pl”
Headers	
Name	Request
Value-expression	Callout Request
Parameters	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter	Value
Name	Username
Value-expression	HTTP.REQ.HEADER("Username").VALUE(0)
Name	Password
Value-expression	HTTP.REQ.HEADER("Password").VALUE(0)
Server Response	
Return Type	TEXT
Expression to extract data from the response	HTTP.RES.BODY(100)

Creating a Responder Policy to Analyze the Response

Updated: 2013-08-30

Create a responder policy, Policy-Responder-3, that will check the response from the callout server and RESET the connection if the source IP address has been blacklisted. Create the policy with the parameters settings shown in the following table. While you can create a responder policy in the Policies subnode and then bind it globally by using the Responder Policy Manager, this demonstration uses the Responder Policy Manager to create the responder policy and bind the policy globally.

Table 2. Parameters and Values for Policy-Responder-3

Parameter	Value
Name	Policy-Responder-3
Action	RESET
Undefined-Result-Action	-Global undefined-result action-
Expression	"HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.HTTP_CALLOUT(HTTP-Callout-3).CONTAINS(\"Authentication Failed\")"

To create a responder policy and bind it globally by using the configuration utility

1. Navigate to AppExpert > Responder.
2. In the details pane, under Policy Manager, click Responder Policy Manager.
3. In the Responder Policy Manger dialog box, click Override Global.
4. Click Insert Policy, and then, in the Policy Name column, click New Policy.
5. In the Create Responder Policy dialog box, do the following:
   1. In Name, type Policy-Responder-3.
   2. In Action, select RESET.
   3. In Undefined-Result Action , select Global undefined-result action.
   4. In the Expression text box, type:
      "HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.HTTP_CALLOUT(HTTP-Callout-3).CONTAINS(\"Authentication Failed\")"
   5. Click Create, and then click Close.
6. Click Apply Changes, and then click Close.

Creating an HTTP Callout Agent on the Remote Server

You now need to create an HTTP callout agent on the remote callout server. The HTTP callout agent receives callout requests from the NetScaler appliance and responds appropriately. The callout agent is a script that is different for each deployment and must be written with server specifications in mind, such as the type of database and the scripting language supported.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Following is sample callout agent pseudo-code that verifies whether the supplied user name and password are valid. The agent can be implemented in any programming language of your choice. The pseudo-code is to be used only as a guideline for developing the callout agent. You can build additional functionality into the program.

To verify the supplied user name and password by using pseudo-code

1. Accept the user name and password supplied in the request and format them appropriately.
2. Connect to the database that contains all the valid user names and passwords.
3. Check the supplied credentials against your database.
4. Format the response as required by the HTTP callout.
5. Send the response to the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case: OWA-Based Spam Filtering

May 21, 2015

Spam filtering is the ability to dynamically block emails that are not from a known or trusted source or that have inappropriate content. Spam filtering requires an associated business logic that indicates that a particular kind of message is spam. When the NetScaler appliance processes Outlook Web Access (OWA) messages based on the HTTP protocol, HTTP callouts can be used to filter spam.

You can use HTTP callouts to extract any portion of the incoming message and check with an external callout server that has been configured with rules that are meant for determining whether a message is legitimate or spam. In case of spam email, for security reasons, the NetScaler appliance does not notify the sender that the email is marked as spam.

The following example conducts a very basic check for various listed keywords in the email subject. These checks can be more complex in a production environment.

To implement this configuration, you must perform the following tasks:

1. Enable the responder feature on the NetScaler appliance.
2. Create an HTTP callout on the NetScaler appliance and configure it with details about the external server and other required parameters.
3. Create a responder policy to analyze the response, and then bind the policy globally.
4. Create a callout agent on the remote server.

Enabling Responder

Updated: 2013-08-30

The responder feature must be enabled before it can be used on the NetScaler appliance.

To enable responder by using the configuration utility

1. Make sure that the responder license is installed.
2. In the configuration utility, expand AppExpert, and right-click Responder, and then click Enable Responder feature.

Creating an HTTP Callout on the NetScaler Appliance

Updated: 2013-08-30

Create an HTTP callout, HTTP-Callout-4, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see "Configuring an HTTP Callout."

Table 1. Parameters and Values for HTTP-Callout-4

Parameter	Value
Name	HTTP-Callout-4
Server to receive callout request	
IP Address	10.103.56.51
Port	80
Request to send to the server	
Method	POST
Host Expression	fffffff
URL Stem Expression	"/cgi-bin/Callout/spam_filter.pl"
Headers	
Name	Request

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter	Value
Value-expression	Callout Request
Parameters	
Name	Subject
Value-expression	("\"" + HTTP.REQ.BODY(1000).AFTER_STR("urn:schemas:httpmail:subject=").BEFORE_STR("\n").TO_LOWER + "\"")
Server Response	
Return Type	BOOL
Expression to extract data from the response	HTTP.RES.BODY(100) .CONTAINS(\"Matched\")

Creating a Responder Action

Updated: 2013-08-30

Create a responder action, Action-Responder-4. Create the action with the parameter settings shown in the following table.

Table 2. Parameters and Values for Action-Responder-4

Parameter	Value
Name	Action-Responder-4
Type	Respond with
Target	"\"HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By: ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\nCache-Control: no-cache\r\n\r\n\""

To create a responder action by using the configuration utility

1. Navigate to AppExpert > Responder > Actions.
2. In the details pane, click Add.
3. In the Create Responder Action dialog box, in Name, type Action-Responder-4.
4. In Type, click Respond with.
5. In Target, type:
   "\"HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By: ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\nCache-Control: no-cache\r\n\r\n\""
6. Click Create, and then click Close.

Creating a Responder Policy to Invoke the HTTP Callout

Updated: 2013-08-30

Create a responder policy, Policy-Responder-4, that will check the request body and, if the body contains the word "*subject*," invoke the HTTP callout to verify the email. Create the policy with the parameter settings shown in the following table. While you can create a responder policy in the Policies subnode and then bind it globally by using the Responder Policy Manager, this demonstration uses the Responder Policy Manager to create the responder policy and bind it globally.

Table 3. Parameters and Values for Policy-Responder-4

Parameter	Value
Name	Policy-Responder-4
Action	Action-Responder-4
Undefined-Result-Action	-Global undefined-result action-
Expression	"HTTP.REQ.BODY(1000).CONTAINS(\"urn:schemas:httpmail:subject\") && SYS.HTTP_CALLOUT(HTTP-Callout-4)"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter	Value

To create a responder policy by using the configuration utility

1. Navigate to AppExpert > Responder.
2. In the details pane, under Policy Manager, click Responder policy manager.
3. In the Responder Policy Manger dialog box, click Override Global.
4. Click Insert Policy, and then, in the Policy Name column, click New Policy.
5. In the Create Responder Policy dialog box, do the following:
   1. In Name, type Policy-Responder-4.
   2. In Action, click Action-Responder-4.
   3. In Undefined-Result Action, click Global undefined-result action.
   4. In the Expression text box, type:
      "HTTP.REQ.BODY(1000).CONTAINS(\"urn:schemas:httpmail:subject\") && SYS.HTTP_CALLOUT(HTTP-Callout-4)"
   5. Click Create, and then click Close.
6. Click Apply Changes, and then click Close.

Creating an HTTP Callout Agent on the Remote Server

You will now need to create an HTTP callout agent on the remote callout server. The HTTP callout agent receives callout requests from the NetScaler appliance and responds accordingly. The callout agent is a script that is different for each deployment and must be written with server specifications in mind, such as the type of database and the scripting language supported.

The following pseudo-code provides instructions for creating a callout agent that checks a list of words that are generally understood to indicate spam mails. The agent can be implemented in any programming language of your choice. The pseudo-code is to be used only as a guideline for developing the callout agent. You can build additional functionality into the program.

To identify spam email by using pseudo-code

1. Accept the email subject provided by the NetScaler appliance.
2. Connect to the database that contains all the terms against which the email subject is checked.
3. Check the words in the email subject against the spam word list.
4. Format the response as required by the HTTP callout.
5. Send the response to the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case: Dynamic Content Switching

May 21, 2015

This use case provides dynamic content switching by using an HTTP callout to get the name of the load balancing virtual server to which the request is forwarded.

1. Add a content switching virtual server.
   > add cs vserver cs_vserver1 HTTP 10.102.29.196 80
2. Create an HTTP callout.
   > add policy httpCallout http_callout1
3. Configure the HTTP callout to respond with the name of the load balancing virtual server from a request that contains the client IP address in the HTTP header "X-CLIENT-IP".
   > set policy httpCallout http_callout1 -IPAddress 10.217.14.23 -port 80 -returnType TEXT -hostExpr "\"www.get-lbvip.com\"" -urlStemExpr "\"/index.html\"" -headers X-CLIENT-IP(CLIENT.IP.SRC) -resu
4. Configure the content switching action to retrieve the callout response.
   > add cs action cs_action1 -targetVserverExpr 'SYS.HTTP_CALLOUT(http_callout1)'
   Note: You must bind a load balancing virtual server to the content switching virtual server to account for:
   - The non-availability of the load balancing virtual server that the callout resolves to.
   - A UNDEF condition that results from the execution of the callout.

   > bind cs vserver cs_vserver1 -lbvserver default_lbvip
5. Configure the content switching policy.
   > add cs policy cs_policy1 -rule true -action cs_action1
6. Binding the content switching policy to the content switching virtual server.
   > bind cs vserver cs_vserver1 -policyName cs_policy1 -priority 10

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Pattern Sets and Data Sets

Jun 20, 2013

Policy expressions for string matching operations on a large set of string patterns tend to become long and complex. Resources consumed by the evaluation of such complex expressions are significant in terms of processing cycles, memory, and configuration size. You can create simpler, less resource-intensive expressions by using pattern matching.

Depending on the type of patterns that you want to match, you can use one of the following features to implement pattern matching:

- A pattern set is an array of indexed patterns used for string matching during default syntax policy evaluation. Example of a pattern set: imagetypes {svg, bmp, png, gif, tiff, jpg}.
- A data set is a specialized form of pattern set. It is an array of patterns of types number (integer), IPv4 address, or IPv6 address.

In many cases, you can use either pattern sets or data sets. However, in cases where you want specific matches for numerical data or IPv4 and IPv6 addresses, you must use data sets.

Note: Pattern sets and data sets can be used only in default syntax policies.

To use pattern sets or data sets, first create the pattern set or data set and bind patterns to it. Then, when you configure a policy for comparing a string in a packet, use an appropriate operator and pass the name of the pattern set or data set as an argument.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How String Matching works with Pattern Sets and Data Sets

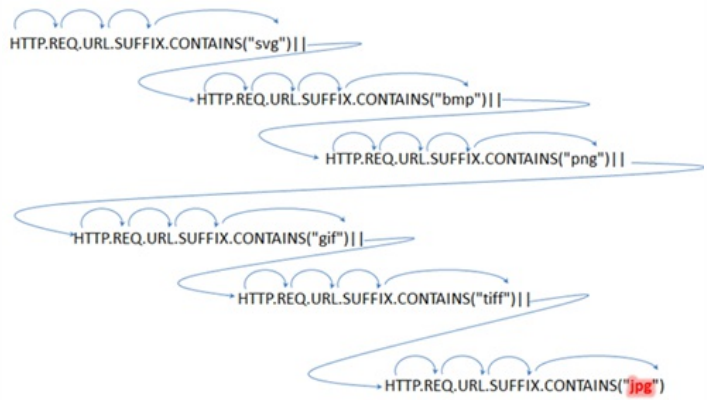
Jul 11, 2013

A pattern set or data set contains a set of patterns, and each pattern is assigned a unique index. When a policy is applied to a packet, an expression identifies a string to be evaluated, and the operator compares the string to the patterns defined in the pattern set or data set until a match is found or all patterns have been compared. Then, depending on its function, the operator returns either a boolean value that indicates whether or not a matching pattern was found or the index of the pattern that matches the string.

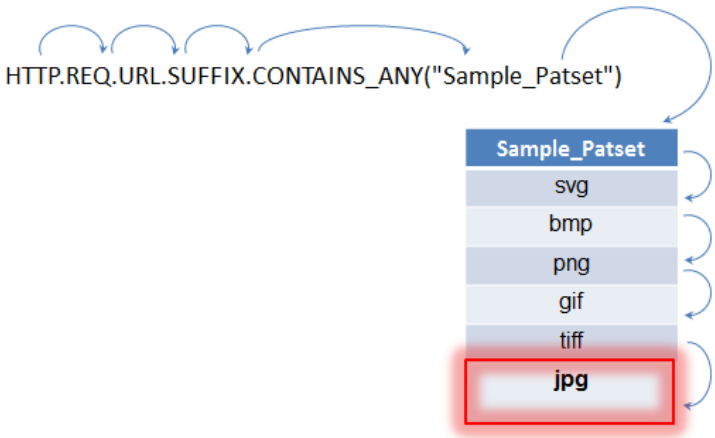
Note: This topic explains the working of a pattern set. Data sets work the same way. The only difference between pattern sets and data sets is the type of patterns defined in the set.
Consider the following use case to understand how patterns can be used for string matching.

You want to determine whether the URL suffix (target text) contains any of the image file extensions. Without using pattern sets, you would have to define a complex expression, as follows:

HTTP.REQ.URL.SUFFIX.CONTAINS("svg") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("bmp") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("png") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("gif") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("tiff") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("jpg")
If the URL has a suffix of "jpg," with the above compound expression, the NetScaler appliance has to iterate through the entire compound expression sequentially, from one sub expression to the next, to determine that the request refers to a jpg image. The following figure shows the steps in the process.



When a compound expression includes hundreds of sub expressions, the above process is resource intensive. A better alternative is an expression that invokes a pattern set, as shown in the following figure.



During policy evaluation as shown above, the operator (CONTAINS_ANY) compares the string identified in the request with the patterns defined in the pattern set until a match is found. With the Sample_Patset expression, the multiple iterations through six sub expressions are reduced to just one.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

By eliminating the need to configure compound expressions that perform string matching with multiple OR operations, pattern sets or data sets simplify configuration and accelerate processing of requests and responses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Pattern Set

Oct 29, 2013

To configure a pattern set, you must specify the strings that are to serve as patterns. You can manually assign a unique index value to each of these patterns, or you can allow the index values to be assigned automatically.

Note: Pattern sets are case sensitive (unless you specify the expression to ignore case). Therefore, the string pattern "product1," for example, is not the same as the string pattern "Product1."

Points to remember about index values

- You cannot bind the same index value to more than one pattern.
- An automatically assigned index value is one number larger than the highest index value of the existing patterns within the pattern set. For example, if the highest index value of existing patterns in a pattern set is 104, the next automatically assigned index value will be 105.
- If you do not specify an index for the first pattern, index value 1 is automatically assigned to that pattern.
- Index values are not regenerated automatically if one or more patterns are deleted or modified. For example, if the set contains five patterns, with indexes from 1 through 5, and if the pattern with an index of 3 is deleted, the other index values in the pattern set are not automatically regenerated to produce values from 1 through 4.
- The maximum index value that can be assigned to a pattern is 4294967290. If that value is already assigned to a pattern in the set, you must manually assign index values to any newly added patterns. An unused index value that is lower than a currently used value cannot be assigned automatically.

To configure a pattern set by using the command line interface

At the command prompt, do the following:

1. Create a pattern set.
   add policy patset <name>

   Example:
   > add policy patset samplepatset
2. Bind patterns to the pattern set.
   bind policy patset <name> <string> [-index <positive_integer>]

   Example:
   > bind policy patset samplepatset product1 -index 1
   Note: Repeat this step for all the patterns you want to bind to the pattern set.
3. Verify the configuration.
   show policy patset <name>

To configure a pattern set by using the configuration utility

1. Navigate to AppExpert > Pattern Sets.
2. In the details pane, click Add to open the Create Pattern Set dialog box.
3. Specify a name for the pattern set in the Name text box.
4. Under Specify Pattern, type the first pattern and, optionally, specify values for the following parameters:
   - Treat back slash as escape character—Select this check box to specify that any backslash characters that you might include in the pattern are to be treated as escape characters.
   - Index—A user assigned index value, from 1 through 4294967290.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Verify that you have entered the correct characters, and then click Add.
6. Repeat steps 4 and 5 to add additional patterns, and then click Create.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Data Set

Oct 11, 2017

To configure a data set, you must specify the strings that are to serve as patterns, and assign a type (number, IPv4 address, or IPv6 address) to each pattern. You can manually assign a unique index value to each of these patterns, or you can allow the index values to be assigned automatically. Dataset is not related to HTTP or any 7 layer protocol. It works only on text/string. There are different types of dataset such as num, ulong, ipv4, and ipv6. To use a dataset on IP, we have to convert the input into text/string.

Note: Data sets are case sensitive (unless you specify the expression to ignore case). Therefore, the string pattern "product1," for example, is not the same as the string pattern "Product1."
The rules applied for index values of data sets are the same as those applied for pattern sets. For information about index values, see "Configuring a Pattern Set."

To configure a data set by using the command line interface

At the command prompt, do the following:

1. Create a data set.
   add policy dataset <name> <type>

   Example:
   > add policy dataset sampledataset ipv4
2. Bind patterns to the data set.
   bind policy dataset <name> <value> [-index <positive_integer>]

   Example:
   > bind policy dataset sampledataset 10.102.29.1 -index 1
   Note: Repeat this step for all the patterns you want to bind to the data set.
3. add policy expression sampleexpression client.ip.src.typecast_text_t.equals_any("sampledataset")
4. Verify the configuration.
   show policy dataset <name>

To configure a data set by using the configuration utility

Navigate to AppExpert > Data Sets, click Add and specify the relevant details.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using Pattern Sets and Data Sets

Aug 30, 2013

Default syntax policy expressions that take pattern sets or data sets as an argument can be used to perform string matching operations.

The usage is as follows:
<text>.<operator>("<name>")
where,

- <text> is the expression that identifies a string in a packet. Example: HTTP.REQ.HEADER("Host").
- <operator> is one of the operators described in the following table.

Table 1. Operators for pattern sets and data sets

Operator	Description
<text>.CONTAINS_ANY(<name>)	Returns true if the target text contains one or more of the patterns defined in the specified pattern set or data set.
<text>.SUBSTR_ANY(<name>)	Returns the first string that matches any pattern defined in the specified pattern set or data set.
<text>.BEFORE_STR_ANY(<name>)	Returns the text that is present before the first occurrence of any of the patterns defined in the specified pattern set or data set.
<text>.AFTER_STR_ANY(<name>)	Returns the text that is present after the first occurrence of any of the patterns defined in the specified pattern set or data set.
<text>.EQUALS_ANY (<name>)	Returns true if the target text exactly matches any of the patterns defined in the specified pattern set or data set.
<text>.ENDSWITH_ANY(<name>)	Returns true if the target text ends with any of the patterns that are defined in the specified pattern set or data set.
<text>.STARTSWITH_ANY(<name>)	Returns true if the target text starts with any of the patterns that are defined in the specified pattern set or data set.
<text>.STARTSWITH_INDEX(<name>)	Evaluates whether the target text starts with any of the patterns that are defined in the specified pattern set or data set. If a match is found, the index of the matching pattern is returned. Otherwise, 0 is returned.
<text>.ENDSWITH_INDEX(<name>)	Evaluates whether the target text ends with any of the patterns that are defined in the specified pattern set or data set. If a match is found, the index of the matching pattern is returned. Otherwise, 0 is returned.
<text>.CONTAINS_INDEX(<name>)	Evaluates whether the target text contains any of the patterns that are defined in the specified pattern set or data set. If a match is found, the index of the matching pattern is returned. Otherwise, 0 is returned.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operator	Description
<text>.EQUALS_INDEX(<name>)	Evaluates whether the target text exactly matches any of the patterns that are defined in the specified pattern set or data set. If an exact match is found, the index of the pattern is returned. Otherwise, 0 is returned.

- <name> is the name of the pattern set or data set

For sample usage, see "Sample Usage."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Sample Usage

Jun 13, 2013

To understand the usage of pattern sets in expressions, consider the example of a pattern set named "imagetypes."

Table 1. Pattern set "imagetypes"

Patterns	Index value
svg	1
bmp	2
png	3
gif	4
tiff	5
jpg	6

Example 1: Determine whether the suffix of an HTTP request is one of the file extensions defined in the "imagetypes" pattern set.

- **Expression.** HTTP.REQ.URL.SUFFIX.EQUALS_ANY("imagetypes")
- **Sample URL.** http://www.example.com/homepageicon.jpg
- **Result.** TRUE

Example 2: Determine whether the suffix of an HTTP request is one of the file extensions defined in the "imagetypes" pattern set, and return the index of that pattern.

- **Expression.** HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes")
- **Sample URL.** http://www.example.com/mylogo.gif
- **Result.** 4 (The index value of the pattern "gif".)

Example 3: Use the index value of a pattern to determine whether the URL suffix is within a specified index-value range.

- **Expression.** HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").GE(3) && HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").LE(5)
- **Sample URL.** http://www.example.com/mylogo.gif
- **Result.** TRUE (The index value of gif file types is 4.)

Example 4: Implement one set of policies for file extensions bmp, jpg, and png, and a different set of policies for gif, tiff, and svg files.

An expression that returns the index of a matched pattern can be used to define traffic subsets for a web application. The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

following two expressions could be used in content switching policies for a content switching virtual server:

- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").LE(3)
- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").GE(4)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy Extensions

Jun 18, 2015

The policy extension feature enables you to write extension functions for built-in policy types. The extensions can be used in policy expressions, just like built-in functions. They are executed when the corresponding policy expressions are evaluated. This feature is useful for:

- Adding customized functions to existing Policies.
- Implementing logical constructs for complex customer requirements.

The policy extension feature addresses these limitations by enabling users to write extension functions for built in Policy types. The extensions can then be used in the policy expressions, just like built-in functions. They are executed when the corresponding policy expressions are evaluated.

The following table lists the policy types that can be used when writing an extension, and their associated mappings.

Policy Type	Mapped Policy Type	Output
TEXT_T	NSTEXT	String
BOOL_AT	NSBOOL	Boolean
NUM_AT	NSNUM	Number (double-precision floating point)
DOUBLE_AT	NSDOUBLE	Number (double-precision floating point)

Prerequisites for Using Policy Extensions

The imported functions must conform to the existing policy standards. Therefore:

- The function name must start with a letter and may contain numbers or underscores.
- The function name is treated as case insensitive by NetScaler policies.
- The function must return a single value even if the extension language returns multiple values.
- Functions with a variable number of arguments are not supported.

How do Policy Extensions Work?

The existing policies on a NetScaler appliance use an interpreter to evaluate the functions, which are imported in a policy extension file. When a user imports a new function in a policy extension file:

1. The extension file is validated for syntax and other conditions.
2. If the validation fails, the error is reported to the user.
3. If the validation succeeds, the extension file is imported to the NetScaler appliance and its contents can be used in policy expressions, just like any built-in policy function
   1. If the policy expression evaluation returns an error during runtime, it is reported as an undef event and the associated error counter is incremented.
      Note: If a policy undef event occurs and the policy rule contains one or more policy extension functions, the **show ns extension <name>** command displays the undef hits when applied to those policy extensions. If the extension function is aborted, the abort counter value is incremented.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. If the policy expression evaluation is successful, expression evaluation resumes until the entire expression is evaluated, or until it is aborted because of an error.

If the extension function takes too long to run, it is aborted, and the error counter pertaining to that extension function is incremented. The extension function is sandboxed, which prevents:

- Excessive CPU usage on the NetScaler appliance.
- Excessive memory usage on the NetScaler appliance.
- Usage of harmful built-in libraries or third-party libraries or binaries.
- Long-running scripts that could potentially cause the NetScaler appliance to reboot.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Policy Extensions

Jun 18, 2015

When your policy extension file is ready, import it to the NetScaler appliance. The import process copies the extension file into a directory on the NetScaler appliance and checks for syntax errors.

After the import, you have to make the extension file available for use in the policy expressions.

Note: The import command is used to download the file content from an external source <src>, or an internal source, onto the NetScaler file system. To load this file content into the packet engine(s) for the first time, use the add command. If there is an update to the file content, the updated content can be downloaded to the NetScaler file system by issuing the import command with -overwrite argument. This updates the contents in the file system. To load the updated content to the packet engine(s), use the update command.

To configure policy extensions by using the command line interface

1. Import the policy extension file to the NetScaler Appliance, from either a web server (using HTTP) or your local workstation.
   1. HTTP Import
      If you have a web server available, you can store the extension file in the webserver directory and import it to the NetScaler appliance.

      import ns extension <src> <name> [-comment<string>] [-overwrite]

      Example:
      import ns extension http://myhost/path/to/extension myextension -comment “Custom crc calculation
   2. Local Import
      You can use SSH client to copy the extension file from your workstation to the /var/tmp directory of the NetScaler appliance

      scp extension-file-name <ns-userid@ns-ip-addr>:/var/tmp

      where
      - extension-file-name is the name of the extension file on your client machine.
      - ns-userid is the NetScaler appliance user with permission to write to /var/tmp.
      - ns-ip-addr is the NetScaler IP address.

      After copying the file to the NetScaler appliance, execute the import command on the NetScaler appliance.

      import ns extension local:<extension-file-name extension-name>

      Note: The CLI should be used to import a local extension file, by running the **import** command.
2. Add the policy extension to the packet engine for evaluation.
   add ns extension <name> [-comment <string>]

   Example:
   add ns extension myextension

After an extension file is imported, you can update it, if you included the -overwrite parameter in the import command, or remove it. You can also display the details of an imported extension file.

To update an extension file on the NetScaler appliance from the source

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

update ns extension <name>

Note: You can update the extension file only after importing the specified extension file to the NetScaler appliance with the -overwrite parameter.
Example:

update ns extension myextension

To remove an extension file from the NetScaler appliance

At the command prompt type:

rm ns extension <name>

Example

rm ns extension myextension

To display the details of the specified extension function on the NetScaler appliance

At the command prompt, type:

show ns extension <name>

Example:

show ns extension myextension

To configure policy extensions by using the configuration utility

1. Import the policy extension file to the NetScaler Appliance, from either a web server (using HTTP) or your local workstation.
   1. Navigate to **AppExpert** > **Policy Extensions**, click **Policy Extension**, from the **Import From** drop-down list, select the URL for the location of the extension file that you want to import.
   2. Navigate to **AppExpert** > **Policy Extensions**, click **Policy Extension** and import the extension file by selecting File in the **Import** From drop-down.
2. Add the policy extension to the packet engine for evaluation.
   Navigate to **AppExpert** > **Policy Extensions** and, on the**Policy Extensions** tab, add the extension file.

To update an extension file on the NetScaler appliance from the source

Navigate to **AppExpert** > **Policy Extensions** and, on the **Policy Extensions** tab, update the extension file.

To remove an extension file from the NetScaler appliance

Navigate to **AppExpert** > **Policy Extensions** and, on the **Policy Extensions** tab, remove the extension file.

To display the details of the specified extension function on the NetScaler appliance

Navigate to **AppExpert** > **Policy Extensions** and, on the **Policy Extensions Functions** tab, click the click drop-down arrow of the extension function that you want to see the details.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy Extensions - Language Overview

Dec 18, 2015

The extension language is based on the Lua 5.2 programming language. Lua provides a compact execution engine with good performance that is designed for embedding in C programs, like NetScaler software.

The extension language is dynamically typed, which means each object carries its own type information. Any variable can hold any type at any time during execution, so variable types are not declared.

The language is also free form, where white space between tokens is ignored. Statements may be separated by semicolons, but that is not required and usually not done. Blocks of statements are typically terminated by end. There are no brackets around blocks like the { and } in C or Java.

Identifiers are sequences of letters (a through z and A through Z), digits (0 through 9), and underscores (_), not starting in a digit. Identifiers are case-sensitive, so var, VAR, and Var are all different identifiers.

Comments are started by --. Everything after -- is ignored to the end of the line. Example:

```
-- This is a comment.
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Simple Types

Dec 18, 2015

The language allows values of the following simple types:

- Numbers
- Strings
- Boolean
- Nil
- Other Types

Numbers

All numbers (even integers) are represented by IEEE 754 floating point values. Integers up to 2^54 have exact representations. Numeric values can be represented by:

- Signed and unsigned decimal integers (examples: 10, -5)
- Real numbers with decimal points (10.5, 3.14159)
- Real numbers with exponents (1.0e+10)
- Hexadecimals (0xffff0000)

NetScaler policy expressions have three numeric types:

- 32-bit integers (num_at)
- 64-bit integers (unsigned_long_at)
- 64-bit floating point (double_at)

All of these are converted into the number type when passed into an extension function, and numbers are converted to the expected policy numeric type when returned.

Strings

Strings are byte sequences of any length. They correspond to the policy **text_at** type. Strings can contain null (0x00) bytes. Arbitrary binary data can be held in strings, including any character code representation (e.g. UTF-8 and full Unicode). However, string functions **likestring.upper()** assume 8-bit ASCII.

Strings are automatically allocated when used. There is no need (or even way) to explicitly allocate buffers for strings. Strings are also automatically deallocated by garbage collection when no longer in use. There is no need (or even way) to explicitly free strings. This automatic allocation and deallocation avoids some common problems in languages like C, such as memory leaks and dangling pointers.

String literals are character strings enclosed in double or single quotes. There is no difference between the two types of quotes: "a string literal" is the same as 'a string literal'. The usual backslash escaping is available: \s (bell), \b (backspace), \f (form feed), \n (newline/line feed), \t (horizontal tab), \\ (backslash), \"(double quote), and \' (single quote). Decimal byte values can be entered by a backslash and one to three digits (\d, \dd, \ddd). Hexadecimal byte values can be entered by a backslash, an x, and two hex digits (\xhh)

A special syntax call the long bracket notation can be used for long, multi-line string literals. This notation encloses the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

string in double square brackets with zero or more equal signs between the brackets -- the idea is to come up with a combination of brackets and equals that is not in the string. No escape sequences are honored in the string. Some examples:

```
[[This is a multi-line string
using long bracket notation.]]

[=[This is a multi-line string
using long notation
with [[ and ]] and
and an unescaped \ in it.]=]
```

Long bracket notation can be used to make a multi-line comment. Example:

```
--[[
This is a multi-line
comment.
--]]
```

Boolean

The usual true and false boolean values are provided. Note that boolean values are different than number values, in contrast to C where zero is assumed to be false and any non-zero value is true.

Nil

nil is a special value that means "no value". It is its own type and is not equivalent to any other value, in contrast to C where NULL is defined to be zero.

Other Types

There are two other types, userdata and threads. These are advanced topics and are not covered here.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Variables

Dec 18, 2015

Variables hold values that may change during extension execution. Because of dynamic typing, any variable may hold values of any type. There are no type declarations for variables. Instead, a variable's type is determined at run time. In fact, the type of a variable's value may change during execution, although this is not a recommended practice. A variable initially has the value nil.

Variable names are identifiers, so are strings of letters, digits, and underscores not beginning in a digit.
Examples: headers, combined_headers.

- Global Variables
- Local Variables

Global Variables

In Lua, variables that are not otherwise declared are global within the program. However, global variables are not allowed in policy extension functions, because there are multiple Packet Engines in which a function can be executed, and each Packet Engine has its own memory.

If you use a global variable in your extension, you will get a runtime error: *attempt to update or create a global* reported in **/var/log/ns.log**.

Typos in variable names are a potential problem, because the variable with the typo will be interpreted as another, global variable, and will not cause a syntax error as in language like C or Java. As noted above, you will get a runtime error instead.

Local Variables

A variable may be declared to be local to a block of statements, such as a function. This is done by *local variable-name*. The variable will be scoped to the block, that is, it will only exist within the block. The local declaration may optionally assign a value to the variable.

Examples:

```
local headers = {}
local combined_headers = {}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions

Dec 18, 2015

Expressions compute values from variable and literal values.

- Arithmetic Operations
- Relational Operations
- Logical Operations
- Concatenation
- Length
- Precedence

Arithmetic Operations

Arithmetic operations are performed on number values. If a string value is used in an arithmetic operation, it is converted to a number -- if that fails, an error is returned.

a + b	add a and b
a - b	subtract b from a
a * b	multiply a and b
a / b	divide a by b
a % b	modulo = a - math.floor(a/b)*b
a ^ b	raise a to the b power; b can be any number
-a	negate a

Relational Operations

Relational operations compare two values and return true if the relation is satisfied and false if it is not. Relational operations can be performed between values of any type. If the values are not the same type, false is returned. Numbers are compared in the usual way. Strings are compared using the collating sequence for the current locale.

a == b	a is equal to b
a ~= b	a is not equal to b

© 1999-2017 Citrix Systems, Inc. All rights reserved.

a < b	a is less than b
a > b	a is greater than b
a <= b	a is less than or equal to b
a >= b	a is greater than or equal to b

Logical Operations

Logical operations are traditionally performed on Boolean values, but in this language they can be performed on any two values. nil and false is considered false and any other value is considered true. Logical operations use short-cut evaluation, where if the first value determines the result of the operation, the second value is not evaluated.

a and b	if a is false or nil then return a else return b
a or b	if a is not false and not nil then return a else return b
not a	if a is not false or nil return false else return true

The and and or operations can be used for conditional evaluation within an expression:

a or b	can be used to provide a default value b if a is uninitialized (nil). This is useful for optional parameters in functions.
a and b or c	can be used to chose non-nil b or c based on the condition a. If a is true, then a and b returns b, and b or c returns b. If a is false, then a and b returns false and false or c returns c. This is equivalent to a ? b : c in the C programming language.

Concatenation

String concatenation is s1 .. s2. This creates a new string large enough to hold the contents of s1 and s2 and copies the contents to the new string. An error results if s1 or s2 are not strings. Note that repeated concatenation may have considerable copying overhead. If you build a string of n bytes by concatenating one byte at a time, this will copy n*(n+1)/2 bytes. For better performance, you can put pieces of a string to be concatenated into a table (discussed later) and then use the table.concat()function. An example of this is shown in the COMBINE_HEADERS() example.

Length

The length of a string s is returned by #s. The # operator is also used with array tables, as discussed later.

Precedence

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operator precedence determines the order in which operations are performed in an expression, with higher precedence operations done before those with lower precedence. Precedence order can as usual be overridden by parentheses. For example, in a + b * c, * has higher precedence than +, so the expression is evaluated as a + (b * c).

highest	^
	not # - (unary)
	* / %
	+ -
	..
	= ~= < > <= >=
	and
lowest	or

Operations with the same precedence are performed left to right (left associative), except ^ and .. that are performed right to left (right associative). So a^b^c is evaluated as a^(b^c).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Assignment

Dec 18, 2015

The assignment statement evaluates an expression and assigns the resulting value to a variable.

variable = expression

As noted earlier, values of any type can be assigned to any variable, so the following is allowed:

```
local v1 = "a string literal"
v1 = 10
```

An assignment statement can actually set multiple variables, using the form

variable1, variable2, ... = expression1, expression2, ...

If there are more variables than expressions, the extra variables are assigned nil. If there are more expressions than variables, the extra expression values are discarded. The expressions are all evaluated before the assignments, so this can be used to succinctly exchange the values of two variables:

v1, v2 = v2, v1

is equivalent to

```
tmp = v1
v2 = v1
v1 = tmp
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tables

Dec 18, 2015

Tables are collections of entries with keys and values. They are the only aggregate data structure provided. All other data structures (arrays, lists, sets, and so on) are built from tables. Table keys and values can be any type, including other tables. Keys and values within the same table can mix types.

- Table Constructors
- Table Usage
- Tables as Arrays
- Tables as Records

Table Constructors

Table constructors allow you to specify a table with keys and associated values. The syntax is:

{[key1] = value1, [key2] = value2, ...}

where the keys and values are expressions. If the keys are strings that are not reserved words, the brackets and quotes around the keys can be omitted. Example:

{key1 = "value1", key2 = "value2", key3 = "value3"}

An empty table is specified simply by {}.

A table constructor may be used in an assignment to set a variable to refer to a table. Examples:

```
local t1 = {} -- set t1 to an empty table
local t2 = {key1 = "value1", key2 = "value2', key3 = "value3"}
```

Note that tables themselves are anonymous. More than one variable may refer to the same table. Continuing the above example:

```
local t3 = t2 -- both t2 and t3 refer to the same table
```

Table Usage

As you would expect, you can use keys to find values in a table. The syntax is table[key], where table is a table reference (typically a variable assigned a table), and key is an expression providing the key. If this is used in an expression and the key exists in the table, this returns the value associated with the key. If the key is not in the table, this returns nil. If this is used as the variable in an assignment, and the key does not exist in the table, it creates a new entry for the key and value. If the key already exists in the table, it replaces the key's value with the new value. Examples:

```
local t = {} -- sets t to an empty table
t["k1"] = "v1" -- creates an entry for key "k1" and value "v1"
v1 = t["k1"] -- sets v1 to the value for key "k1" = "v1"
t["k1"] = "new_v1" -- sets the value for key "k1" to "new_v1"
```

Tables as Arrays

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The traditional array can be implemented using a table with integer keys as indices. An array can have any indices, including negative ones, but the convention is to start arrays at index 1 (not 0 as is the case with languages like C and Java). There is a special purpose table constructor for such arrays:

{value1, value2, value3, ... }

Array references are then array[index].

The length operator # returns the number of elements in an array with consecutive indices starting at 1. Example:

```
local a = {"value1", "value2", "value3"}
local length = #a -- sets length to the length of array a = 3
```

Arrays can be sparse, where only the defined elements are allocated. But # cannot be used on a sparse array with non-consecutive indices. Example:

```
local sparse_array = {} -- set up an empty array
sparse_array[1] = "value1" -- add an element at index 1
sparse_array[99] = "value99" -- add an element at index 99
```

Multidimensional arrays can be set up as tables of tables. For example, a 3x3 matrix could be set up by:

```
local m = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}
local v22 = m[2][2] -- sets v22 to 5
```

Tables as Records

Records with fields can be implemented as tables with field name keys. The reference form table.field can be used for table["field"]. Examples:

```
local person = {name = "John Smith", phone = "777-777-7777"}
local name = person.name -- sets name to "John Smith"
```

An array of tables can be used for a sequence of records. Example:

```
local people = {
{name = "John Smith", phone = "777-777-7777"},
{name = "Jane Doe", phone = "888-888-8888"}
...
}

name = people[2].name -- sets name to "Jane Doe"
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Control Structures

Dec 18, 2015

The extension function language provides the usual statements to control program execution.

- If Then Else
- While Do and Repeat Until
- Numeric For
- Break
- Goto

If Then Else

If statements select blocks of statements to execute based on one or more conditions. There are three forms:

If then Form

```
if expression then
   statements to execute if expression is not false or nil
end
```

If then else Form

```
if expression then
   statements to execute if expression is not false or nil
else
   statements to execute if expression is false or nil
end
```

If then elseif else Form

```
if expression1 then
   statements to execute if expression1 is not false or nil
   elseif expression2 then
      statements to execute if expression2 is not false or nil
      . . .
else
   statements to execute if all expressions are false or nil
end
```

Example:

```
if headers[name] then
   local next_value_index = #(headers[name]) + 1
   headers[name][next_value_index] = value
else
   headers[name] = {name .. ":" .. value}
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
end
```

Note:

- The expression is not enclosed in parentheses as is the case in C and Java.
- There is no equivalent to the C/Java switch statement. You have to use a series of if elseif statements to do the equivalent.

While Do and Repeat Until

The **while** and **repeat** statements provide loops controlled by an expression.

```
while expression do
   statements to execute while expression is not false or nil
end

repeat
   statements to execute until expression is not false or nil
until expression
```

Example for while:

```
local a = {1, 2, 3, 4}
local sum, i = 0, 1 -- multiple assignment initializing sum and i
while i <= #a do -- check if at the end of the array
   sum = sum + a[i] -- add array element with index i to sum
   i = i + 1 -- move to the next element
end
```

Example for repeat:

```
sum, i = 0, 1 -- multiple assignment initializing sum and i
repeat
   sum = sum + a[i] -- add array element with index i to sum
   i = i + 1 -- move to the next element
until i > #a -- check if past the end of the array
```

Of course it is possible to write a loop that does not terminate, for example, if you omit the i = i + 1 statement in either of these examples. When such a function is executed, NetScaler will detect that the function did not complete in a reasonable time and will kill it with a runtime error:

Cpu limit reached. Terminating extension execution in [[string "function extension function..."]]: line line-number.

will be reported in /var/log/ns.log.

Numeric For

There are two types of for loops. The first is the numeric for, which is similar to the usual use of the for statement in C and Java. The numeric for statement initializes a variable, tests if the variable has passed a final value, and if not executes a block of statements, increments the variable, and repeats. The syntax for the numerical for loop is:

```
for variable = initial, final, increment do
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    statements in the loop body
end
```

where initial, final, and increment are all expressions that yield (or can be converted to) numbers. variable is considered to be local to the for loop statement block; it cannot be used outside of the loop. increment can be omitted; the default is 1. The expressions are evaluated once at the beginning of the loop. The terminating condition is variable > final if the increment is positive and variable < final if the increment is negative. The loop terminates immediately if the increment is 0.

Example (equivalent to the while and repeat loops in the preceding section):

```
sum = 0
for i = 1, #a do -- increment defaults to 1
    sum = sum + a[i]
end
```

The second type of for loop is the generic for, which can be used for more flexible types of loops. It involves the use of functions, so will be discussed later after functions have been introduced.

Break

The break statement is used inside a while, repeat, or for loop. It will terminate the loop and resume execution at the first statement after the loop. Example (also equivalent to the preceding while, repeat, and for loops):

```
sum, i = 0, 1
while true do
    if i > #a then
        break
    end
    sum = sum + a[i]
    i = i + 1
end
```

Goto

The goto statement can be used to jump forward or backward to a label. The label is an identifier, and its syntax is ::label::. The goto statement is goto label. Example (once again equivalent to the preceding loops):

```
sum, i = 0, 1
::start_loop::
    if i > #a then
        goto end_loop -- forward jump
    end
    sum = sum + a[i]
    i = i + 1
    goto start_loop -- backwards jump
::end_loop::
...
```

There has been a long running controversy over using gotos in programming. In general, you should try to use the other control structures to make your functions more readable and reliable. But occasional judicious use of gotos may lead to

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

better programs. In particular, gotos may be useful in handling errors.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Functions

Dec 18, 2015

Functions are a basic building block of programming -- they are a convenient and powerful way to group statements that perform a task. They are also the interface between NetScaler policy expressions and extension functions -- you define extension functions that are called by policy expressions. Functions consist of function definitions that specify what values are passed into and out of the function and what statements are executed for the function, and function calls, which execute functions with specific input data and get results from the function.

- NetScaler Policy Extension Function Definition
- Local Function Definition
- Function Body and Return
- Function Calls
- Iterator Functions and Generic For Loops

NetScaler Policy Extension Function Definition

Since the NetScsaler policy expression language is strongly typed, the definition of an extension function must specify the types of its inputs and its return value. The Lua function definition has been extended to include these types:

```
function self-type:function-name(parameter1: parameter1-type, etc.): return-type
   statements
end
```

where,

> the types are NSTEXT, NSNUM, NSBOOL, or NSDOUBLE.
>
> self-type is the type of the implicit self parameter that is passed into the function. When the extension function is used in an NetScaler policy expression, this is the value generated by the expression to the left of the function. Another way to view this is that the function extends that type in the NetScaler policy language.
>
> The parameter-types are the types of each parameter specified in the extension function call in the policy expression. An extension function can have zero or more parameters.
>
> return-type is the type of the value returned by the extension function call. It will be the input to the part of the policy expression, if any, to the right of the function, or else is the value of the expression result.

Example:

function NSTEXT:COMBINE_HEADERS() : NSTEXT

Use of the extension function in a policy expresssion:

HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n").COMBINE_HEADERS()

Here the self parameter is the result of HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n"), which is a text value. The result of the COMBINE_HEADERS() call is text, and since there is nothing to the right of this call, the result of the entire

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

expression is text.

Local Function Definition

Besides extension functions, no global functions can be defined in an extension file. But local functions can be defined within extension functions using the normal Lua function statement. This declares the name of the function and the names of its parameters (also known as arguments), and like all declarations in Lua, does not specify any types. The syntax for this is:

```
local function function-name(parameter1-name, parameter2-name, etc.)
    statements
end
```

The function and parameter names are all identifiers. (The function name is actually a variable and the function statement is shorthand for local function-name = function(parameter1, etc.), but you don't have to understand this subtlety to use functions.)

Note that etc. is used here for continuation of the pattern of parameter names instead of the usual ... . This is because ... itself actually means a variable parameter list, which will not be discussed here.

Function Body and Return

The block of statements between the function and end statements is the function body. In the function body, the function parameters act like local variables, with values supplied by the function calls, as described previously.

The return statement supplies values to be returned to the caller of the function. It must appear at the end of a block (in a function, if then, for loop, and so on; It can be in its own block do return ... end). It may specify no, one, or more than one return values:

```
return -- returns nil
return expression -- one return value
return expression1, expression2, ... -- multiple return values
```

Examples:

```
local function fsum(a)
    local sum = 0
    for i = 1, #a do
        sum = sum + a[i]
    end
    return sum
end

local function fsum_and_average(a)
    local sum = 0
    for i = 1, #a do
        sum = sum + a[i]
    end
    return sum, sum/#a
end
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function Calls

A function call executes the body of a function, supplying values for its parameters, and receiving results. The syntax for a function call is function-name(expression1, expression2, etc.), where the function parameters are set to the corresponding expressions. The number of expressions and parameters need not be the same. If there are fewer expressions than parameters, the remaining parameters are set to nil. So you can make one or more parameters at the end of the call optional, and your function can check if they are specified by checking if they are not nil. A common way to do this is with the or operation:

```
function f(p1, p2) -- p2 is optional
    p2 = p2 or 0 -- if p2 is nil, set to a default of 0
    . . .
end
```

If there are more expressions than parameters, the remaining expression values are ignored.

As noted previously, functions can return multiple values. These returns can be used in a multiple assignment statement. Example:

```
local my_array = {1, 2, 3, 4}
local my_sum, my_ave = sum_and_average(my_array)
```

Iterator Functions and Generic For Loops

Now that we have introduced functions, we can talk about generic for loops. The syntax for the generic for loop (with one variable) is:

```
for variable in iterator(parameter1, parameter2, etc.) do
    statements in the for loop body
end
```

where iterator() is a function with zero or more parameters that provides a value for variable on each iteration of the loop body. The iterator function keeps track of where it is in the iteration using a technique called closure, which you don't have to worry about here. It signals the end of the iteration by returning nil. Iterator functions can return more than one value, for use in a multiple assignment.

Writing an iterator function is beyond the scope of this paper, but there are a number of useful built-in iterators that illustrate the concept. One is the pairs() iterator, which iterates through the entries in a table and returns two values, the key and the value of the next entry.

Example:

```
local t = {k1 = "v1", k2 = "v2", k3 = "v3"}
local a = {} -- array to accumulate key-value pairs
local n = 0 -- number of key-value pairs
for key, value in pairs(t) do
    n = n + 1
    a[n] = key .. " = " .. value -- add key-value pair to the array
end
local s = table.concat(a, "; ") -- concatenate all key-value pairs into one string
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Another useful iterator is the string.gmatch() function, which will be used in the following COMBINE_HEADERS() example.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy Extensions - Use Cases

Dec 18, 2015

Certain customer applications have requirements that cannot be addressed with existing policies and expressions. The policy extension feature enables customers to add customized functions to their applications to meet their requirement.

The following use cases illustrate the addition of new functions using the policy extension feature on the NetScaler appliance.

- Case 1: Custom Hash
- Case 2: Collapse double slashes in URLs
- Case 3: Combine Headers

Case 1: Custom Hash

The CUSTOM_HASH function provides a mechanism to insert any type of hash value in the responses sent to the client. In this use case, the hash function is used to compute the hash of the query string for a rewrite HTTP request and insert an HTTP header named CUSTOM_HASH with the computed value. The CUSTOM_HASH function implements the DJB2 hash algorithm.

Sample Usage of CUSTOM_HASH

COPY

```
> add rewrite action test_custom_hash insert_http_header "CUSTOM_HASH" "HTTP.REQ.URL.QUERY.CUSTOM_HASH"
```

Sample Definition of CUSTOM_HASH()

COPY

```
-- Extension function to compute custom hash on the text

-- Uses the djb2 string hash algorithm

function NSTEXT:CUSTOM_HASH() : NSTEXT

   local hash = 5381

   local len = string.len(self)

   for i = 1, len do

      hash = bit32.bxor((hash * 33), string.byte(self, i))

   end

   return tostring(hash)

end
```

Line-by-line description of the above sample

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Case 2: Collapse Double Slashes in URLs

Collapsing double slashes in URLs improves the website rendering time, because browsers parse the single slash URLs more efficiently. The single slash URLs also to maintain compatibility with applications that do not accept double slashes. The policy extension feature allows customers to add a function that replaces the double slashes with single slashes in the URLs. The following example illustrates the addition of a policy extension function that that collapses double slashes in URLs.

Sample Definition of COLLAPSE_DOUBLE_SLASHES() COPY

```
-- Collapse double slashes in URL to a single slash and return the result

function NSTEXT:COLLAPSE_DOUBLE_SLASHES() : NSTEXT

    local result = string.gsub(self, "//", "/")

    return result

end
```

Line-by-line description of the above sample

Case 3: Combine Headers

Certain customer applications cannot handle multiple headers in a request. Also, parsing of duplicate headers with same header values, or multiple headers with same name but different values in a request, consumes time and network resources. The policy extension feature allows customers to add a function to combine these headers into single headers with a value combining the original values. For example, combining the values of the headers H1 and H2.

Original Request:

```
GET /combine_headers HTTP/1.1
User-Agent: amigo unit test
Host: myhost
H2: h2val1
H1: abcd
Accept: */*
H2: h2val2
Content-Length: 0
H2: h2val3
H1: 1234
```

Modified Request:

```
GET /combine_headers HTTP/1.1
User-Agent: amigo unit test
Host: myhost
H2: h2val1, h2val2, h2val3
H1: abcd, 1234
Accept: */*
Content-Length: 0
```

In general, this type of request modification is done using the Rewrite feature, using policy expressions to delineate the part of the request to be modified (the target) and the modification to be performed (the string builder expression). However, policy expressions do not have

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the ability to iterate over an arbitrary number of headers.

The solution to this problem requires an extension to the policy facility. To do this, we will define an extension function, called COMBINE_HEADERS. With this function, we can set up the following rewrite action:

> **add rewrite action** combine_headers_act replace 'HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n")' 'HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n").COMBINE_HEADERS'

Here, the rewrite target is HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n"). The AFTER_STR("HTTP/1.1\r\n") is required because FULL_HEADER includes the first line of the HTTP request (e.g. GET /combine_headers HTTP/1.1).

The string builder expression is HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n").COMBINE_HEADERS, where the headers (minus the first line) are fed into the COMBINE_HEADERS extension function, which combines and returns the values for headers.

Sample Definition of COMBINE_HEADERS() COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
-- Extension function to combine multiple headers of the same name into one header.


function NSTEXT:COMBINE_HEADERS(): NSTEXT

    local headers = {} -- headers

    local combined_headers = {} -- headers with final combined values

    -- Iterate over each header (format "name:value\r\n")

    -- and build a list of values for each unique header name.

    for name, value in string.gmatch(self, "([^:]+):([^\r\n]*)\r\n") do

        if headers[name] then

            local next_value_index = #(headers[name]) + 1

            headers[name][next_value_index] = value

        else

            headers[name] = {name .. ":" .. value}

        end

    end


    -- iterate over the headers and concat the values with separator ","

    for name, values in pairs(headers) do

        local next_header_index = #combined_headers + 1

        combined_headers[next_header_index] = table.concat(values, ",")

    end


    -- Construct the result headers using table.concat()

    local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"

    return result_str

end
```

Line-by-line description of the above sample

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy Extensions - Library Reference

Dec 18, 2015

The list of libraries supported in policy extensions.

- Basic Library
- String Library
- Regular Expression Patterns - Character Classes
- Regular Expression Patterns - Pattern Items
- Table Library
- Math Library
- Bitwise Library
- Operating System Library
- NetScaler Library

Basic Library

assert(v[,message])	Issues an error, with an optional message, when v is false.
error(message)	Terminates a function and reports the error message.
ipairs(a)	Iterator for an array a. Returns an index and value for each iteration.
pairs(t)	Iterator for a table t. Returns a key and value for each iteration.
tonumber(e[,base])	Converts e to a number, with an optional base.
tostring(v)	Converts v to a string
type(v)	Returns the type of v: number, string, boolean, table, etc.
getmetatable (object)	Returns nil if the object does not have a metatable. Otherwise, if the object's metatable has a "__metatable" field, returns the associated value. Otherwise, returns the metatable of the given object.
setmetatable (table, metatable)	Sets the metatable for the given table. (You cannot change the metatable of other types from Lua, only from C.) If metatable is nil, removes the metatable of the given table. If the original metatable has a "__metatable" field, raises an error.
select (index, ···)	Returns all arguments after argument number index. If index is string "#", then it returns the total number of extra arguments it received.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pcall (f [, arg1, ···])	Calls function f with the given arguments in protected mode. It returns status code as first result which tells whether call succeeded or not. If call succeeded, then along with status code it also returns all results from the call, otherwise returns error message.
xpcall (f, msgh [, arg1, ···])	This function is similar to pcall, except that it also takes an argument for error handling.
_VERSION	Returns the current interpreter version.

String Library

string.byte(s[,i[,j]])	Returns the byte values for s[i] to s[j]. Default i = 1 and j = i
string.char(...)	Returns a string constructed of the integer parameters.
string.find(s,pattern[,init[,plain])	Looks for the first match of a regular expression pattern in s. Return the first and last indices of match or nil. init is index to start, default 1. plain = true means pattern is not a regex.
string.format(form,...)	Returns a formatted version of the parameters.
string.gmatch(s,pattern)	Iterator for searching s with the regex pattern. Returns matching values.
string.gsub(s,pattern,repl[,n])	Returns a copy of s in which all (or n) occurences of the pattern have been replaced by repl.
string.len(s)	Returns the string length.
string.lower(s)	Returns a copy of the string converted to lowercase.
string.match(s,pattern[,init])	Looks for the first match of the regex pattern in s and returns the captures or the whole pattern. init is the index to start, default 1.
string.rep(s,n[,sep])	Returns a string that is n copies of s, with separator sep, default no separator
string.reverse(s)	Returns a string that is s reversed.
string.sub(s,i[,j])	Returns the substring of s from s[i] to s[j], default j is the end of the string.
string.upper(s)	Returns a copy of the string converted to uppercase.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

string.dump (function)	Returns a string containing a binary representation of the given function.

Regular Expression Patterns - Character Classes

x	the character x, except for magic characters ^$()%.[]*+-?)
.	any character
%a	any letter
%c	any control character
%d	any digit
%g	any printable character except space
%l	any lowercase letter
%p	any punctuation character
%s	any white space character
%u	any uppercase letter
%w	any alphanumber letter
%x	an escaped magic character x (for example %%)
[set]	a set of characters: sequence of individual characters, ranges x-y, and % classes
[^set]	characters not in the set.

Regular Expression Patterns - Pattern Items

X	a character class
X*	0 or more longest repetitions of characters in X

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X+	1 or more repetitions of characgers in X
X-	0 or more shortest repetitions of characters in X
X?	0 or 1 character in X
%n	n=1 to 9; matches nth captured string
%bxy	matches substring between two balanced characters x and y. Example %b() matches substring between two balanced parentheses.
%f[set]	matches an empty string at at any position such that the next character belongs to set and the previous character does not belong to set.

A pattern is a sequence of pattern items. ^pattern matches the beginning of a string and pattern$ matches the end of the string.

Matched substrings can be captured using (pattern). Parentheses with no pattern () capture the current string position (a number).

Table Library

table.concat(list[,sep[,i[,j]]])	Returns a string list[i] .. sep .. list[i+1] .. sep . . . list[j]. Default sep is the empty string. Default i is 1, j is #list.
table.insert(list,[pos,]value)	Inserts value into list at index pos. Default for pos is #list (end of the list).
table.pack(...)	Returns an array containing the parameters starting at index 1, and a key n with the total number of parameters.
table.remove(list[,pos])	Removes from list the element at position pos, shifting elements to fill the position. Returns the removed element. Default for posis #list (end of the list.)
table.sort(list[,comp])	Sort the elements of list in place. comp is the comparison function to use. Default for comp is <.
table.unpack(list[,i[,j]])	Returns list[i] through list[j]. Default for i is 1 and j is #list<c/ode>.

Math Library

Various trigonometric and logarithmic functions not shown.

math.abs(x)	Returns the absolute value of x.
math.ceil(x)	Returns the smallest integer >= x.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

math.floor(x)	Returns the largest integer <= x.
math.fmod(x,y)	Returns the remainder of x/y rounding the quotient towards zero.
math.huge	A value >= any other number.
math.max(x,...)	Returns the maximum argument.
math.min(x,...)	Returns the minimum argument.
math.modf(x)	Returns the integral and fractional parts of x.
math.random()	Returns a pseudo-random number between 0 and 1.
math.random(m)	Returns a pseudo-random integer between 1 and m.
math.random(m, n)	Returns a pseudo-random integer betwen m and n.
math.randomseed(x)	Sets the pseudo-random number generator set to x.
math.sqrt(x)	Returns the square root of x (x^0.5)
math.acos(x)	Returns the arc cosine of x (in radians).
math.asin(x)	Returns the arc sine of x (in radians).
math.atan(x)	Returns the arc tangent of x (in radians).
math.atan2(y, x)	Returns the arc tangent of y/x (in radians).
math.cos(x)	Returns the cosine of x.
math.cosh(x)	Returns the hyperbolic cosine of x.
math.sin(x)	Returns the sine of x.
math.sinh(x)	Returns the hyperbolic sine of x.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

math.tan(x)	Returns the tangent of x.
math.tanh(x)	Returns the hyperbolic tangent of x.
math.deg(x)	Returns the angle x (given in radians) in degrees.
math.exp(x)	Returns the value e^x.
math.frexp (x)	Returns m and e such that x = m2e, e is an integer and the absolute value of m is in the range [0.5, 1).
math.ldexp (m, e)	Returns m2e (e should be an integer).
math.log (x [, base])	Returns the logarithm of x in the given base. The default for base is e.
math.pow (x, y)	Returns x^y.
math.rad (x)	Returns the angle x (given in degrees) in radians.
math.pi	The value of π.

Bitwise Library

Unless otherwise stated:

- All functions accept numeric arguments in the range (-2^51,+2^51).
- Each argument is normalized to the remainder of its division by 2^32 and truncated to an integer (in some unspecified way), so that its final value falls in the range [0,2^32 - 1].
- All results are in the range [0,2^32 - 1].

bit32.arshift(x,disp)	Returns x arithmetically shifted disp bits to the right (+disp) or left (-disp).
bit32.band(...)	Returns the bitwise and of the arguments.
bit32.bnot(x)	Returns the bitwise negation of x.
bit32.bor(...)	Returns the bitwise or of the arguments.
bit32.btest(...)	Returns true if the bitwise and of the arguments is not zero.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

bit32.bxor(...)	Returns the bitwise exclusive or of the arguments.
bit32.extract(n,field[,width])	Returns the bits in n from field to field + width - 1 (bits number from most to least significant). Default for width is 1.
bit32.replace(n,v,field[,width])	Returns a copy of n with bits from field to field + width -1 replaced by v. Default width is 1.
bit32.lrotate(x,disp)	Returns x rotated disp bits to the left (+disp) or right (-disp).
bit32.lshift(x,disp)	Returns x shifted disp bits to the left (+disp) or right (-disp).
bit32.rrotate(x,disp)	Returns x rotated disp bits to the right (+disp) or left (-disp).
bit32.rshift(x,disp)	Returns x shifted disp bits to the right (+disp) or left (-disp).

Operating System Library

os.clock ()	Returns an approximation of the amount in seconds of CPU time.
os.date ([format [, time]])	Returns a string or a table containing date and time, formatted according to the given string format.
os.time ([table])	Returns the current time when called without arguments, or a time representing the date and time specified by the given table.
os.difftime (t2, t1)	Returns the number of seconds from time t1 to time t2.

NetScaler Library

ns.logger:level(message)	To log messages where level is emergency, alert, critical, error, warning, notice, info, or debug. The parameters are the same as the C printf() function: a format string, and a variable number of arguments to supply values for the % specifiers in the format string.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting Policy Extensions

Dec 18, 2015

If your extension function is not behaving as expected, you can use extension tracing functionality to verify the behavior of your extension function. You can also add logging to your extension function by using the custom logging functionality, where you can define the log level to be captured on the NetScaler appliance.

This topic provides information on:

- Extension Tracing
- Custom Logging

Extension Tracing

To show what your extension function is doing, extension tracing functionality logs the execution of the function to the NetScaler system log (/var/log/ns.log). The trace logging uses the DEBUG log level, which normally is not enabled. Therefore, you have to enable ALL log levels. Then you can enable tracing by setting the -trace option of the set ns extension command. The available settings are:

- off turn off tracing (equivalent to unset ns extension -trace).
- calls trace function calls with arguments and function returns with the first return value.
- lines trace the above plus line numbers for executed lines.
- all trace the above plus local variables changed by executed lines.

Example

set audit syslogParams -loglevel ALL

set ns extension combine_headers -trace all

Each trace message has the format

log-header : default NSEXTENSION Message message-number 0 : "TRACE function-name CALL call-number: event"

Where,

- log-header supplies timestamps, the NetScaler IP address, and the Packet Engine ID.
- message-number is a sequential number identifying the log message.
- function-name is the extension function name.
- call-number is a sequential number for each extension function call. It can be used to group all the trace messages for an extension function call.
- event is one of the following:
  - CALL function-name ; parameter-values indicates that the function has been called with the specified parameters.
  - RETURN FROM function-name ; return = value indicates that a function has returned the specified (first) value. (Additional return values are not reported.)
  - LINE line-number ; variable-values indicates that a line has been executed and lists any variables with changed values.

Where,

- value or values is

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- a number, with or without a decimal point,
- a string, enclosed in double quotes and with escaped characters as described earlier,
- a boolean true or false,
- nil,
- a table constructor, of the format {[key1]=value1,[key2]=value2, ...}.

- parameter-values is parameter1 = value1 ; parameter2 = value2 , ...
- variable-values is variable1 = value1 ; variable2 = value2 , ...

An example of abbreviated log messages:

>shell tail -f /var/log/ns.log | grep TRACE | more

... NSEXTENSION Message 3035 0 : "TRACE combine_headers CALL 30 : CALL COMBINE_HEADERS; self = "User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nHost: 10.217.24.7\r\nAccept: */*\r\nH2: h2val1\r\nH1: abcd\r\nH2: h2val2\r\nH2: h2val3\r\n\r\n""

... NSEXTENSION Message 3036 0 : "TRACE combine_headers CALL 30 : LINE 4; headers = {}"

... NSEXTENSION Message 3037 0 : "TRACE combine_headers CALL 30 : LINE 5; combined_headers = {}"

... NSEXTENSION Message 3038 0 : "TRACE combine_headers CALL 30 : CALL gmatch"

... NSEXTENSION Message 3039 0 : "TRACE combine_headers CALL 30 : RETURN FROM gmatch; return = function 0x2bee5a80"

... NSEXTENSION Message 3040 0 : "TRACE combine_headers CALL 30 : CALL for iterator"

... NSEXTENSION Message 3041 0 : "TRACE combine_headers CALL 30 : RETURN FROM for iterator; return = " curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3""

... NSEXTENSION Message 3042 0 : "TRACE combine_headers CALL 30 : LINE 9; name = "User-Agent"; value = " curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3""

... NSEXTENSION Message 3043 0 : "TRACE combine_headers CALL 30 : LINE 10"

... NSEXTENSION Message 3044 0 : "TRACE combine_headers CALL 30 : LINE 14; headers = {["User-Agent"]={[1]="User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3"}}"

. . .

... NSEXTENSION Message 3117 0 : "TRACE combine_headers CALL 30 : CALL for iterator"

... NSEXTENSION Message 3118 0 : "TRACE combine_headers CALL 30 : RETURN FROM for iterator; return = nil"

... NSEXTENSION Message 3119 0 : "TRACE combine_headers CALL 30 : LINE 19"

... NSEXTENSION Message 3120 0 : "TRACE combine_headers CALL 30 : CALL concat"

... NSEXTENSION Message 3121 0 : "TRACE combine_headers CALL 30 : RETURN FROM concat; return = "User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\nAccept: */*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3"" ... NSEXTENSION Message 3122 0 : "TRACE combine_headers CALL 30 : LINE 25; result_str = "User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\nAccept: */*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3\r\n\r\n""

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

... NSEXTENSION Message 3123 0 : "TRACE combine_headers CALL 30 : RETURN FROM COMBINE_HEADERS; return = "User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\nAccept: */*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3\r\n\r\n""

Custom Logging

You can also add your own logging to your extension function. To do so, use the built-in ns.logger:*level*() function, where *level* is emergency , alert , critical , error , warning , notice , info , or debug . The parameters are the same as the C printf() function: a format string, and a variable number of arguments to supply values for the % specified in the format string. For example, you might add the following to the COMBINE_HEADERS function to log the result of a call:

```
local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"

ns.logger:info("Result: %s", result_str)

return result_str
```

The above function would log the following message to/var/log/ns.log for the sample input shown in the abbreviated log messages examples in the Extension Tracing section above.

... : default NSEXTENSION Message 143 0 : "Result: Host: 10.217.24.7:2000^M H1: abcd, 1234^M User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Accept: */*^M H2: h2val1, h2val2, h2val3^M ^M"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Variables

Jun 02, 2015

Variables are named objects that store information in the form of tokens. These tokens are used within and across different transactions on the NetScaler Appliance for internal computation and policy processing.

The NetScaler appliance supports creation of variables of the following types:

- **Singleton variables.** Can have a single value of one of the following types: ulong and text (max-size). The ulong type is an unsigned 64-bit integer, the text type is a sequence of bytes, and max-size is the maximum number of bytes in the sequence.
- **Map variables.** Maps hold values associated with keys: each key-value pair is called a map entry. The key for each entry is unique within the map. Maps are specified as follows:

  map (key_type, value_type, max-values).

  where,

  - *key_type* is the data type of the key. It is of type text (max-size).
  - *value_type* is the data type of the values of the map. It can be of type ulong or text (max-size).
  - *max-values* is the maximum number of entries that the map can contain. It is of type ulong.

Values for these variables are set using assignments which must be invoked on policy actions.

Note: Variables are not yet supported in a high-availability setup or in a cluster.

Variables Scope

A map variable or a singleton variable can have a global scope. Alternatively, the scope of a singleton variable can be limited to a single transaction.

- **Global Scope Variable** - A variable with global scope (the default) has only one instance, and that instance has the same value(s) across all cores of a NetScaler appliance and across all nodes of a cluster or HA configuration. Global variable values exist until they are explicitly deleted, until they expire, or until a standalone appliance is restarted or all nodes of a cluster or HA configuration are restarted.
- **Transaction Scope Variable** - A variable with transaction scope has a separate instance, with its own value, for each transaction processed by the NetScaler appliance. When the transaction processing is complete, the transaction variable value is deleted.

Note: Transaction scope variables are available in NetScaler release 10.5.e or later.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring and Using Variables

Mar 20, 2014

You must first create a variable and then assign a value or specify the operation that must be performed on the variable. After performing these operations, you can use the assignment as a policy action.

Note: Once configured, a variable's settings cannot be modified or reset. If the variable needs to be changed, the variable and all references to the variable (expressions and assignments) must be deleted. The variable can then be re-added with new settings, and the references (expressions and assignments) can be re-added.

To configure variables by using the command line interface

1. Create a variable.
   add ns variable <name> -type <string> [-scope global] [-ifFull (undef | lru)] [-ifValueTooBig (undef | truncate)] [-ifNoValue (undef | init)] [-init <string>] [-expires <positive_integer>] [-comment <string>]

   Note: Refer to the man page "man add ns variable" for description of the command parameters.
   Example 1: Create a ulong variable named "my_counter" and initialize it to 1.
   add ns variable my_counter –type ulong -init 1
   Example 2: Create a map named "user_privilege_map". The map will contain keys of maximum length 15 characters and text values of maximum length 10 characters, with a maximum of 10000 entries.
   add ns variable user_privilege_map -type map(text(15),text(10),10000)
   Note: If the map contains 10000 unexpired entries, assignments for new keys reuse one of the least recently used entries. By default, an expression trying to get a value for a non-existent key will initialize an empty text value.
2. Assign the value or specify the operation to be performed on the variable. This is done by creating an assignment.
   add ns assignment <name> -variable <expression> [-set <expression> | -add <expression> | -sub <expression> | -append <expression> | -clear] [-comment <string>]

   Note: A variable is referenced by using the variable selector ($). Therefore, **$variable1** is used to refer to text or ulong variables. Similarly, **$variable2[key-expression]** is used to refer to map variables.
   Example 1: Define an assignment named "inc_my_counter" that automatically adds 1 to the "my_counter" variable.
   add ns assignment inc_my_counter -variable $my_counter -add 1
   Example 2: Define an assignment named "set_user_privilege" that adds to the "user_privilege_map" variable an entry for the client's IP address with the value returned by the "get_user_privilege" HTTP callout.
   add ns assignment set_user_privilege -variable $user_privilege_map[client.ip.src.typecast_text_t] -set sys.http.callout(get_user_privilege)
   Note: If an entry for that key already exists, the value will be replaced. Otherwise a new entry for the key and value will be added. Based on the previous declaration for user_privilege_map, if the map already has 10000 entries, one of the least recently used entries will be reused for the new key and value.
3. Invoke the variable assignment in a policy.
   There are two functions that can operate on map variables.

   - **$name.valueExists(key-expression).** Returns true if there is a value in the map selected by the key-expression. Otherwise returns false. This function will update the expiration and LRU information if the map entry exists, but will not create a new map entry if the value does not exist.

   - **$name.valueCount.** Returns the number of values currently held by the variable. This is the number of entries in a map. For a singleton variable, this is 0 if the variable is uninitialized or 1 otherwise.

   Example: Invoke the assignment named "set_user_privilege" with a compression policy.
   > add cmp policy set_user_privilege_pol -rule $user_privilege_map.valueExists(client.ip.src.typecast_text_t).not -resAction set_user_privilege

To configure variables by using the configuration utility

1. Navigate to AppExpert > NS Variables, to create a variable.
2. Navigate to AppExpert > NS Assignments, to assign value(s) to the variable.
3. Navigate to the appropriate feature area where you want to configure the assignment as an action.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case: Caching User Privileges

Oct 29, 2017

In this use case, user privileges ("GOLD", "SILVER", and so on) must be retrieved from an external web service.

To achieve this use case, perform the following operations:

1. Create an HTTP callout to fetch the user privileges from the external web service.
   add policy httpcallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port <port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod (GET | POST)] [-hostExpr <string>] [-urlStemExpr <string>] [-h
   add policy httpcallout get_user_privilege -ipaddress 10.217.193.84 -port 80 -returnType text -httpMethod GET -hostExpr '"/get_user_privilege"' -resultExpr 'http.res.body(5)'
2. Store the privileges in a variable.
   add ns variable <name> -type <string> [-scope (global | transaction)][-ifFull (undef | lru)] [-ifValueTooBig (undef | truncate)][-ifNoValue (undef | init)] [-init <string>] [-expires <positive_integer>] [-comr
   add ns variable user_privilege_map -type map(text(15),text(10),10000) -expires 1200
   add ns assignment set_user_privilege -variable $user_privilege_map[client.ip.src] -set sys.http_callout(get_user_privilege)
3. Create a policy to check if there is already a cached entry for the client's IP address; if not, it calls the HTTP callout to set a map entry for the client.
   add cmp policy <name> -rule <expression> -resAction <string>
   add cmp policy set_user_privilege_pol -rule $user_privilege_map.valueExists(client.ip.src).not -resAction set_user_privilege>
4. Create a policy that compresses if the cached privilege entry for the client is "GOLD".
   add cmp policy <name> -rule <expression> -resAction <string>
   add cmp policy compress_if_gold_privilege_pol -rule '$user_privilege_map[client.ip.src].eq("GOLD")' -resAction compress
5. Bind the compression policies globally.
   bind cmp global <policyName> [-priority <positive_integer>] [-state (ENABLED | DISABLED)] [-gotoPriorityExpression <expression>] [-type <type>] [-invoke (<labelType> <labelName>)]
   bind cmp global set_user_privilege_pol -priority 10 NEXT
   bind cmp global compress_if_gold_privilege_pol -priority 20 END

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case: Limiting the Number of Sessions

Jun 25, 2015

In this use case, the requirement is to limit the number of active backend sessions. In the deployment, each session login has login in the URL and each session logout has logout in the URL. On successful login, the backend sets a sessionid cookie with a unique 10 character value.

To achieve this use case, perform the following operations:

1. Create a map variable that can store each active session. The key of the map is the sessionid. The expiry time for the variable is set to 600 seconds (10 minutes).
   > add ns variable session_map -type map(text(10),ulong,100) -expires 600
2. Create the following assignments for the map variable:
   - Create an entry for the sessionid and set that value to 1 (this value is not actually used).
     > add ns assignment add_session -variable '$session_map[http.req.cookie.value("sessionid")]' -set 1
   - Deallocate the entry for a session ID, which implicitly decrements the value count for session_map.
     > add ns assignment delete_session -variable '$session_map[http.req.cookie.value("sessionid")]' -clear
3. Create responder policies for the following:
   - To check if a map entry exists for that sessionid in the HTTP request. The add_session assignment is executed if the map entry does not exist.
     > add responder policy add_session_pol 'http.req.url.contains("twbkwbis.P_SabanciLogin") || $session_map.valueExists(http.req.cookie.value("netsuis"))' add_session
     Note: The valueExists() function in the add_session_pol policy counts as a reference to the session's map entry, so each request resets the expiration timeout for its session. If no requests for a session are received after 10 minutes, the session's entry will be deallocated.
   - To check when the session is logged out. The delete_session assignment is executed.
     > add responder policy delete_session_pol "http.req.url.contains(\"Logout\")" delete_session
   - To check for login requests and if the number of active sessions exceed 100. If these conditions are satisfied, in order to limit the number of sessions, the user is redirected to a page that indicates that the server is busy.
     > add responder action redirect_too_busy redirect "/too_busy.html"
     > add responder policy check_login_pol "http.req.url.contains(\"twbkwbis.P_SabanciLogin\") && $session_map.valueCount > 100" redirect_too_busy
4. Bind the responder policies globally.
   > bind responder global add_session_pol 30 next
   > bind responder global delete_session_pol 10
   > bind responder global check_login_pol 20

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policies and Expressions

May 26, 2015

The following topics provide the conceptual and reference information that you require for configuring advanced policies on the Citrix® NetScaler® appliance.

You can also download a list of all the expressions supported on the NetScaler appliance and the hierarchical order in which they can be invoked. The reference is in a zip file which you can download from:

- For NetScaler 10.5: http://support.citrix.com/article/CTX141344
- For NetScaler 10.1: http://support.citrix.com/article/CTX137705

Introduction to Policies and Expressions	Describes the purpose of expressions, policies, and actions, and how different NetScaler applications make use of them.
Configuring Advanced Policies	Describes the structure of advanced policies and how to configure them individually and as policy banks.
Configuring Advanced Expressions: Getting Started	Describes expression syntax and semantics, and briefly introduces how to configure expressions and policies.
Advanced Expressions: Evaluating Text	Describes expressions that you configure when you want to operate on text (for example, the body of an HTTP POST request or the contents of a user certificate).
Advanced Expressions: Working with Dates, Times, and Numbers	Describes expressions that you configure when you want to operate on any type of numeric data (for example, the length of a URL, a client's IP address, or the date and time that an HTTP request was sent).
Advanced Expressions: Parsing HTTP, TCP, and UDP Data	Describes expressions for parsing IP and IPv6 addresses, MAC addresses, and data that is specific to HTTP and TCP traffic.
Advanced Expressions: Parsing SSL Certificates	Describes how to configure expressions for SSL traffic and client certificates, for example, how to retrieve the expiration date of a certificate or the certificate issuer.
Advanced Expressions: IP and MAC Addresses, Throughput, VLAN IDs	Describes expressions that you can use to work with any other client- or server-related data not discussed in other chapters.
Typecasting Data	Describes expressions for transforming data of one type to another.
Regular Expressions	Describes how to pass regular expressions as arguments to operators in advanced

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	expressions.
Configuring Classic Policies and Expressions	Provides details on how to configure the simpler policies and expressions known as classic policies and classic expressions.
Expressions Reference	A reference for classic and advanced expression arguments.
Summary Examples of Advanced Expressions and Policies	Examples of classic and advanced expressions and policies, in both quick reference and tutorial format, that you can customize for your own use.
Tutorial Examples of Advanced Policies for Rewrite	Examples of advanced policies for use in the Rewrite feature.
Tutorial Examples of Classic Policies	Examples of classic policies for NetScaler features such as application firewall and SSL.
Migration of Apache mod_rewrite Rules to Advanced Policies	Examples of functions that were written using the Apache HTTP Server mod_rewrite engine, with examples of these functions after translation into Rewrite and Responder policies on the NetScaler.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Introduction to Policies and Expressions

Jun 04, 2015

For many NetScaler features, policies control how a feature evaluates data, which ultimately determines what the feature does with the data. A policy uses a logical expression, also called a rule, to evaluate requests, responses, or other data, and applies one or more actions determined by the outcome of the evaluation. Alternatively, a policy can apply a profile, which defines a complex action.

Some NetScaler features use default syntax policies, which provide greater capabilities than do the older, classic, policies. If you migrated to a newer release of the NetScaler software and have configured classic policies for features that now use default syntax policies, you might have to manually migrate policies to the default syntax.

This document contains the following details:

- Classic and Default Syntax Policies
- Classic and Default Syntax Expressions
- Converting Classic Expressions to the Newer Default Expression Syntax
- Before You Proceed

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Classic and Default Syntax Policies

May 25, 2015

Classic policies evaluate basic characteristics of traffic and other data. For example, classic policies can identify whether an HTTP request or response contains a particular type of header or URL.

Default syntax policies can perform the same type of evaluations as classic policies. In addition, default syntax policies enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example, transforming data in the body of a request into an HTTP header).

In addition to assigning a policy an action or profile, you bind the policy to a particular point in the processing associated with the NetScaler features. The bind point is one factor that determines when the policy will be evaluated.

This document includes the following details:

- Benefits of Using Default Syntax Policies
- Basic Components of a Classic or Default Syntax Policy
- How Different NetScaler Features Use Policies
- About Actions and Profiles
- About Policy Bindings
- About Evaluation Order of Policies
- Order of Evaluation Based on Traffic Flow

Benefits of Using Default Syntax Policies

Default syntax policies use a powerful expression language that is built on a class-object model, and they offer several options that enhance your ability to configure the behavior of various NetScaler features. With default syntax policies, you can do the following:

- Perform fine-grained analyses of network traffic from layers 2 through 7.
- Evaluate any part of the header or body of an HTTP or HTTPS request or response.
- Bind policies to the multiple bind points that the default syntax policy infrastructure supports at the default, override, and virtual server levels.
- Use goto expressions to transfer control to other policies and bind points, as determined by the result of expression evaluation.
- Use special tools such as pattern sets, policy labels, rate limit identifiers, and HTTP callouts, which enable you to configure policies effectively for complex use cases.

Additionally, the configuration utility extends robust graphical user interface support for default syntax policies and expressions and enables users who have limited knowledge of networking protocols to configure policies quickly and easily. The configuration utility also includes a policy evaluation feature for default syntax policies. You can use this feature to evaluate a default syntax policy and test its behavior before you commit it, thus reducing the risk of configuration errors.

Basic Components of a Classic or Default Syntax Policy

Updated: 2013-09-02

Following are a few characteristics of both classic and default syntax policies:

Name.

Each policy has a unique name.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Rule.

The rule is a logical expression that enables the NetScaler feature to evaluate a piece of traffic or another object. For example, a rule can enable the NetScaler to determine whether an HTTP request originated from a particular IP address, or whether a Cache-Control header in an HTTP request has the value “No-Cache.”

Default syntax policies can use all of the expressions that are available in a classic policy, with the exception of classic expressions for the SSL VPN client. In addition, default syntax policies enable you to configure more complex expressions.

Bindings.

To ensure that the NetScaler can invoke a policy when it is needed, you associate the policy, or bind it, to one or more bind points.
You can bind a policy globally or to a virtual server. For more information, see "About Policy Bindings."

An associated action.

An action is a separate entity from a policy. Policy evaluation ultimately results in the NetScaler performing an action. For example, a policy in the integrated cache can identify HTTP requests for .gif or .jpeg files. An action that you associate with this policy determines that the responses to these types of requests are served from the cache.

For some features, you configure actions as part of a more complex set of instructions known as a profile.

How Different NetScaler Features Use Policies

Updated: 2013-09-30

The NetScaler supports a variety of features that rely on policies for operation. The following table summarizes how the NetScaler features use policies.

Table 1. NetScaler Feature, Policy Type, and Policy Usage

Feature Name	Policy Type	How You Use Policies in the Feature
System	Classic	For the Authentication function, policies contain authentication schemes for different authentication methods. For example, you can configure LDAP and certificate-based authentication schemes. You also configure policies in the Auditing function.
DNS	Default	To determine how to perform DNS resolution for requests.
SSL	Classic and Default	To determine when to apply an encryption function and add certificate information to clear text. To provide end-to-end security, after a message is decrypted, the SSL feature re-encrypts clear text and uses SSL to communicate with Web servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature Name	Policy Type	How You Use Policies in the Feature
Compression	Classic and Default	To determine what type of traffic is compressed.
Integrated Caching	Default	To determine whether HTTP responses are cacheable.
Responder	Default	To configure the behavior of the Responder function.
Protection Features	Classic	To configure the behavior of the Filter, SureConnect, and Priority Queuing functions.
Content Switching	Classic and Default	To determine what server or group of servers is responsible for serving responses, based on characteristics of an incoming request. Request characteristics include device type, language, cookies, HTTP method, content type, and associated cache server.
AAA - Traffic Management	Classic Exceptions: ● Traffic policies support only default syntax policies ● Authorization policies support both classic and default syntax policies.	To check for client-side security before users log in and establish a session. Traffic policies, which determine whether single sign-on (SSO) is required, use only the default syntax. Authorization policies authorize users and groups that access intranet resources through the appliance.
Cache Redirection	Classic	To determine whether responses are served from a cache or from an origin server.
Rewrite	Default	To identify HTTP data that you want to modify before serving. The policies provide rules for modifying the data. For example, you can modify HTTP data to redirect a request to a new home page, or a new server, or a selected server based on the address of the incoming request, or you can modify the data to mask server information in a response for security purposes. The URL Transformer function identifies URLs in HTTP transactions and text files for the purpose of evaluating whether a URL should be transformed.
Application	Classic and Default	To identify characteristics of traffic and data that should or should not be

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature Name	Policy Type	How You Use Policies in the Feature
Firewall		admitted through the firewall
NetScaler Gateway, Clientless Access function	Default	To define rewrite rules for general Web access using the NetScaler Gateway.
NetScaler Gateway	Classic	To determine how the NetScaler Gateway performs authentication, authorization, auditing, and other functions.

About Actions and Profiles

Updated: 2013-09-30

Policies do not themselves take action on data. Policies provide read-only logic for evaluating traffic. To enable a feature to perform an operation based on a policy evaluation, you configure actions or profiles and associate them with policies.

Note: Actions and profiles are specific to particular features. For information about assigning actions and profiles to features, see the documentation for the individual features.

About Actions

Actions are steps that the NetScaler takes, depending on the evaluation of the expression in the policy. For example, if an expression in a policy matches a particular source IP address in a request, the action that is associated with this policy determines whether the connection is permitted.

The types of actions that the NetScaler can take are feature specific. For example, in Rewrite, actions can replace text in a request, change the destination URL for a request, and so on. In Integrated Caching, actions determine whether HTTP responses are served from the cache or an origin server.

In some NetScaler features actions are predefined, and in others they are configurable. In some cases, (for example, Rewrite), you configure the actions using the same types of expressions that you use to configure the associated policy rule.

About Profiles

Some NetScaler features enable you to associate profiles, or both actions and profiles, with a policy. A profile is a collection of settings that enable the feature to perform a complex function. For example, in the application firewall, a profile for XML data can perform multiple screening operations, such as examining the data for illegal XML syntax or evidence of SQL injection.

Use of Actions and Profiles in Particular Features

The following table summarizes the use of actions and profiles in different NetScaler features. The table is not exhaustive. For more information about specific uses of actions and profiles for a feature, see the documentation for the feature.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 2. Use of Actions and Profiles in Different NetScaler Features

Feature	Use of an Action	Use of a Profile
Application firewall	Synonymous with a profile	All application firewall features use profiles to define complex behaviors, including pattern-based learning. You add these profiles to policies.
NetScaler Gateway	The following features of the NetScaler Gateway use actions: • **Pre-Authentication.** Uses Allow and Deny actions. You add these actions to a profile. • **Authorization.** Uses Allow and Deny actions. You add these actions to a policy. • **TCP Compression.** Uses various actions. You add these actions to a policy.	The following features use a profile: • Pre-Authentication • Session • Traffic • Clientless Access After configuring the profiles, you add them to policies.
Rewrite	You configure URL rewrite actions and add them to a policy.	Not used.
Integrated Caching	You configure caching and invalidation actions within a policy	Not used.
AAA - Traffic Management	You select an authentication type, set an authorization action of ALLOW or DENY, or set auditing to SYSLOG or NSLOG.	You can configure session profiles with a default timeout and authorization action.
Protection Features	You configure actions within policies for the following functions: • Filter • Compression • Responder • SureConnect	Not used.
SSL	You configure actions within SSL policies	Not used.
System	The action is implied. For the Authentication function, it is either Allow or Deny. For Auditing, it is Auditing On or Auditing Off.	Not used.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature	Use of an Action	Use of a Profile
DNS	The action is implied. It is either Drop Packets or the location of a DNS server.	Not used.
SSL Offload	The action is implied. It is based on a policy that you associate with an SSL virtual server or a service.	Not used.
Compression	Determine the type of compression to apply to the data	Not used.
Content Switching	The action is implied. If a request matches the policy, the request is directed to the virtual server associated with the policy.	Not used.
Cache Redirection	The action is implied. If a request matches the policy, the request is directed to the origin server.	Not used.

About Policy Bindings

Updated: 2013-09-30

A policy is associated with, or bound to, an entity that enables the policy to be invoked. For example, you can bind a policy to request-time evaluation that applies to all virtual servers. A collection of policies that are bound to a particular bind point constitutes a policy bank.

Following is an overview of different types of bind points for a policy:

Request time global.

A policy can be available to all components in a feature at request time.

Response time global.

A policy can be available to all components in a feature at response time.

Request time, virtual server-specific.

A policy can be bound to request-time processing for a particular virtual server. For example, you can bind a request-time policy to a cache redirection virtual server to ensure that particular requests are forwarded to a load balancing virtual server for the cache, and other requests are sent to a load balancing virtual server for the origin.

Response time, virtual server-specific.

A policy can also be bound to response-time processing for a particular virtual server.

User-defined policy label.

For default syntax policies, you can configure custom groupings of policies (policy banks) by defining a policy label and collecting a set of related policies under the policy label.

Other bind points.

The availability of additional bind points depends on type of policy (classic or default syntax), and specifics of the relevant NetScaler feature. For example, classic policies that you configure for the NetScaler Gateway have user and group bind points.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For additional information about default syntax policy bindings, see "Binding Policies That Use the Default Syntax" and Configuring a Policy Bank for a Virtual Server. For additional information about classic policy bindings, see "Configuring a Classic Policy."

About Evaluation Order of Policies

For classic policies, policy groups and policies within a group are evaluated in a particular order, depending on the following:

- The bind point for the policy, for example, whether the policy is bound to request-time processing for a virtual server or global response-time processing. For example, at request time, the NetScaler evaluates all request-time classic policies before evaluating any virtual server-specific policies.
- The priority level for the policy. For each point in the evaluation process, a priority level that is assigned to a policy determines the order of evaluation relative to other policies that share the same bind point. For example, when the NetScaler evaluates a bank of request-time, virtual server-specific policies, it starts with the policy that is assigned to the lowest priority value. In classic policies, priority levels must be unique across all bind points.

For default syntax policies, as with classic policies, the NetScaler selects a grouping, or bank, of policies at a particular point in overall processing. Following is the order of evaluation of the basic groupings, or banks, of default syntax policies:

1. Request-time global override
2. Request-time, virtual server-specific (one bind point per virtual server)
3. Request-time global default
4. Response-time global override
5. Response-time virtual server-specific
6. Response-time global default

However, within any of the preceding banks of policies, the order of evaluation is more flexible than in classic policies. Within a policy bank, you can point to the next policy to be evaluated regardless of the priority level, and you can invoke policy banks that belong to other bind points and user-defined policy banks.

Order of Evaluation Based on Traffic Flow

As traffic flows through the NetScaler and is processed by various features, each feature performs policy evaluation. Whenever a policy matches the traffic, the NetScaler stores the action and continues processing until the data is about to leave the NetScaler. At that point, the NetScaler typically applies all matching actions. Integrated Caching, which only applies a final Cache or NoCache action, is an exception.

Some policies affect the outcome of other policies. Following are examples:

- If a response is served from the integrated cache, some other NetScaler features do not process the response or the request that initiated it.
- If the Content Filtering feature prevents a response from being served, no subsequent features evaluate the response.

If the application firewall rejects an incoming request, no other features can process it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Classic and Default Syntax Expressions

May 25, 2015

One of the most fundamental components of a policy is its rule. A policy rule is a logical expression that enables the policy to analyze traffic. Most of the policy's functionality is derived from its expression.

An expression matches characteristics of traffic or other data with one or more parameters and values. For example, an expression can enable the NetScaler to accomplish the following:

- Determine whether a request contains a certificate.
- Determine the IP address of a client that sent a TCP request.
- Identify the data that an HTTP request contains (for example, a popular spreadsheet or word processing application).
- Calculate the length of an HTTP request.

This document includes the following details:
- About Classic Expressions
- About Default Syntax Expressions

About Classic Expressions

Classic expressions enable you to evaluate basic characteristics of data. They have a structured syntax that performs string matching and other operations.

Following are a few simple examples of classic expressions:

- An HTTP response contains a particular type of Cache Control header.

res.http.header Cache-Control contains public

- An HTTP response contains image data.

res.http.header Content-Type contains image/

- An SSL request contains a certificate.

req.ssl.client.cert exists

About Default Syntax Expressions

Updated: 2013-09-02

Any feature that uses default syntax policies also uses default syntax expressions. For information about which features use default syntax policies, see the table "NetScaler Feature, Policy Type, and Policy Usage."

Default syntax expressions have a few other uses. In addition to configuring default syntax expressions in policy rules, you configure default syntax expressions in the following situations:

Integrated Caching:

You use default syntax expressions to configure a selector for a content group in the integrated cache.

Load Balancing:

You use default syntax expressions to configure token extraction for a load balancing virtual server that uses the TOKEN

© 1999-2017 Citrix Systems, Inc. All rights reserved.

method for load balancing.

Rewrite:

You use default syntax expressions to configure rewrite actions.

Rate-based policies:

You use default syntax expressions to configure limit selectors when configuring a policy to control the rate of traffic to various servers.

Following are a few simple examples of default syntax expressions:

- An HTTP request URL contains no more than 500 characters.

http.req.url.length <= 500

- An HTTP request contains a cookie that has fewer than 500 characters.

http.req.cookie.length < 500

- An HTTP request URL contains a particular text string.

http.req.url.contains(".html")

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Converting Classic Expressions to the Newer Default Expression Syntax

Sep 15, 2016

You can convert a classic expression to the default expression syntax by using the nspepi conversion tool. You can also use the tool to convert all the classic expressions in the NetScaler configuration to the default syntax (with the exception of NetScaler entities that currently support only classic expressions).

The conversion tool does not convert policies configured for the following features, because the features currently support only classic policies:

- Authentication, Pre-authentication
- SSL
- Cache redirection
- VPN (session, traffic, and tunnel traffic)
- Content filtering (The responder feature not only provides you with functionality that is equivalent to that provided by the content filtering feature but also surpasses the content filtering feature in the use cases that it supports. Additionally, responder supports the more powerful default syntax for policy expressions.)

The following NetScaler features support both classic and default syntax expressions and, therefore, support the conversion of classic expressions to default syntax expressions:

- Application firewall policies
- Authorization policies
- Named expressions
- Compression policies
- Content switching policies
- User-defined, rule-based tokens/persistency (the –rule parameter value that is specified for a load balancing virtual server)

This document includes the following details:

- About the Conversion Process
- Converting Expressions
- Converting a NetScaler Configuration File
- Conversion Warnings

About the Conversion Process

Updated: 2013-09-02

When parsing a NetScaler configuration file, the conversion tool performs the following actions:

1. In commands that create classic named expressions, the conversion tool replaces the names of the classic expressions with default syntax expressions.
2. In commands that support only the classic syntax, if classic named expressions are used, the conversion tool replaces the names of the classic expressions with the actual classic expressions they represent. This action ensures that the names of expressions in classic-only features do not reference the default syntax expressions created from Step 1.
3. In commands associated with entities that support both the classic syntax and the default syntax, the conversion tool replaces all classic expressions in commands with default syntax expressions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

Consider the following sample configuration commands:
add policy expression ne_c1 "METHOD == GET"
add policy expression ne_c2 "ne_c1 || URL == /*.htm "
add filter policy pol1 -rule "ne_c2" -reqAction YES
add cmp policy pol2 -rule "REQ.HTTP.HEADER Accept CONTAINS \'text/html\'" -resAction COMPRESS
add cmp policy pol3 -rule "ne_c1 || ne_c2" -resAction GZIP
In the commands that create the classic named expressions ne_c1 and ne_c2, the tool replaces the names of the expressions with actual default syntax expressions. This action, which corresponds to Step 1 described earlier, results in the following commands:

add policy expression ne_c2 "HTTP.REQ.METHOD.EQ(\"GET\")"

add policy expression ne_c1 "HTTP.REQ.URL.SUFFIX.EQ(\"htm\")"

The filter policy command supports only the classic syntax. Therefore, the conversion tool replaces the classic named expression ne_c1 with the actual classic expression it represents. Note that the tool replaces ne_c1 in the expression for ne_c2, and then replaces ne_c2 in the filter policy with the classic expression. This action, which corresponds to Step 2 described earlier, results in the following command:

add filter policy pol1 -rule "METHOD == GET || URL == /*.htm" -reqAction YES

The compression feature supports both classic and default syntax expressions. Therefore, in the command that creates the compression policy pol2, the conversion tool replaces the expression with a default syntax expression. This action, which corresponds to Step 3 described earlier, results in the following command:

add cmp policy pol2 -rule "HTTP.REQ.HEADER(\"Accept\").AFTER_STR(\"text/html\").LENGTH.GT(0)" -resAction COMPRESS

The command that creates the compression policy pol3 is unaffected by the conversion process because, after the conversion process is complete, ne_c1 and ne_c2 reference the default syntax expressions that result from Step 1.

Client security messages are not supported in the newer default policy format and, therefore, are lost. The SYS.EVAL_CLASSIC_EXPR function is replaced with a default policy expression. The following entities support the SYS.EVAL_CLASSIC_EXPR function:

- DNS policies
- Rate limit selectors
- Cache selectors
- Cache policies
- Content switching policies
- Rewrite policies
- URL transformation policies
- Responder policies
- Application firewall policies
- Authorization policies
- Compression policies
- CVPN access policies

After performing the conversion, the tool saves the changes in a new configuration file. The new configuration file is

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

created in the directory in which the input file exists. The name of the new configuration file is the same as the name of the input configuration file except for the string new_ used as a prefix. Conversion warnings are reported in a warning line at the end of the screen output. Additionally, a warning file is created in the directory in which the input configuration file resides. For more information about the warning file and the types of warnings that are reported, see "Conversion Warnings."

Converting Expressions

Updated: 2013-09-02

You can use the nspepi tool to convert a single classic expression to the default syntax. The nspepi tool must be run from the shell prompt on the NetScaler appliance.

To convert a classic expression to the default syntax by using the command line interface

At the shell prompt, type:

nspepi -e "<classic expression>"
Example

```
root@NS# nspepi -e "REQ.HTTP.URL == /*.htm"
 "HTTP.REQ.URL.REGEX_MATCH(re#/(.*)\.htm#)"
```

Converting a NetScaler Configuration File

Updated: 2013-09-02

You can use the nspepi tool to convert all the classic expressions in a NetScaler configuration file to the default syntax (except for those commands that do not support the default syntax). The nspepi tool must be run from the shell prompt on the NetScaler appliance.

To convert all the classic expressions in a NetScaler configuration file to the default syntax by using the command line interface

At the shell prompt, type:

nspepi -f "<ns config file>" -v
Example

```
root@NS# nspepi -f ns.conf
OUTPUT: New configuration file created: new_ns.conf
OUTPUT: New warning file created: warn_ns.conf
WARNINGS: Total number of warnings due to bind commands: 18
WARNINGS: Line numbers which has bind command issues: 305, 306, 706, 707, 708, 709, 710, 711, 712, 713,
714, 715, 767, 768, 774, 775, 776, 777
root@NS#
```

Conversion Warnings

When classic expressions that are included in CLI commands are upgraded to the default syntax, the number of characters in the expression might exceed the 1499-character limit. The commands that include expressions longer than 1499 characters fail when the configuration is being applied. You must manually update these commands.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

In addition, multiple classic policies can be bound to a given bind point with priority 0 or with equal priority, but the default syntax policy infrastructure does not support a priority value of 0 or policies with the same priority at a given bind point. These commands fail when the configuration is being applied. The commands must be updated manually with the correct priority values.

The line numbers of lines that threw a warning during conversion are listed at the end of the output in a warning line. In addition, a warning file is created in the same directory as the one in which the old and new configuration files reside. The name of the warning file is the same as the name of the input configuration file except that the string warn_ is added as a prefix.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Before You Proceed

May 25, 2015

Before configuring expressions and policies, be sure you understand the relevant NetScaler feature and the structure of your data, as follows:

- Read the documentation on the relevant feature.
- Look at the data stream for the type of data that you want to configure.

You may want to run a trace on the type of traffic or content that you want to configure. This will give you an idea of the parameters and values, and operations on these parameters and values, that you need to specify in an expression.

Note: The NetScaler supports either classic or default syntax policies within a feature. You cannot have both types in the same feature. Over the past few releases, some NetScaler features have migrated from using classic policies and expressions to default syntax policies and expressions. If a feature of interest to you has changed to the default syntax format, you may have to manually migrate the older information. Following are guidelines for deciding if you need to migrate your policies:

- If you configured classic policies in a version of the Integrated Caching feature prior to release 9.0 and then upgrade to version 9.0 or later, there is no impact. All legacy policies are migrated to the default syntax policy format.
- For other features, you need to manually migrate classic policies and expressions to the default syntax if the feature has migrated to the default syntax.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Default Syntax Policies

Sep 30, 2013

You can create default syntax policies for various NetScaler features, including DNS, Rewrite, Responder, and Integrated Caching, and the clientless access function in the NetScaler Gateway. Policies control the behavior of these features.

When you create a policy, you assign it a name, a rule (an expression), feature-specific attributes, and an action that is taken when data matches the policy. After creating the policy, you determine when it is invoked by binding it globally or to either request-time or response-time processing for a virtual server.

Policies that share the same bind point are known as a *policy bank*. For example, all policies that are bound to a virtual server constitute the policy bank for the virtual server. When binding the policy, you assign it a priority level to specify when it is invoked relative to other policies in the bank. In addition to assigning a priority level, you can configure an arbitrary evaluation order for policies in a bank by specifying Goto expressions.

In addition to policy banks that are associated with a built-in bind point or a virtual server, you can configure *policy labels*. A policy label is a policy bank that is identified by an arbitrary name. You invoke a policy label, and the policies in it, from a global or virtual-server-specific policy bank. A policy label or a virtual-server policy bank can be invoked from multiple policy banks.

For some features, you can use the policy manager to configure and bind policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rules for Names in Identifiers Used in Policies

Mar 20, 2012

The names of identifiers in the named expression, HTTP callout, pattern set, and rate limiting features must begin with an ASCII alphabet or an underscore (_). The remaining characters can be ASCII alphanumeric characters or underscores (_).

The names of these identifiers must not begin with the following reserved words:

- The words ALT, TRUE, or FALSE or the Q or S one-character identifier.

- The special-syntax indicator RE (for regular expressions) or XP (for XPath expressions).
- Expression prefixes, which currently are the following:
  - CLIENT
  - EXTEND
  - HTTP
  - SERVER
  - SYS
  - TARGET
  - TEXT
  - URL
  - MYSQL
  - MSSQL

Additionally, the names of these identifiers cannot be the same as the names of enumeration constants used in the policy infrastructure. For example, the name of an identifier cannot be IGNORECASE, YEAR, or LATIN2_CZECH_CS (a MySQL character set).

Note: The NetScaler appliance performs a case-insensitive comparison of identifiers with these words and enumeration constants. For example, names of the identifiers cannot begin with TRUE, True, or true.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating or Modifying a Policy

Nov 14, 2013

All policies have some common elements. Creating a policy consists, at minimum, of naming the policy and configuring a rule. The policy configuration tools for the various features have areas of overlap, but also differences. For the details of configuring a policy for a particular feature, including associating an action with the policy, see the documentation for the feature.

To create a policy, begin by determining the purpose of the policy. For example, you may want to define a policy that identifies HTTP requests for image files, or client requests that contain an SSL certificate. In addition to knowing the type of information that you want the policy to work with, you need to know the format of the data that the policy is analyzing.

Next, determine whether the policy is globally applicable, or if it pertains to a particular virtual server. Also consider the effect that the order in which your policies are evaluated (which will be determined by how you bind the policies) will have on the policy that you are about to configure.

To create a policy by using the command line interface

At the command prompt, type the following commands to create a policy and verify the configuration:

- add responder|dns|cs|rewrite|cache policy <policyName> -rule <expression> [<feature-specific information>]
- show rewrite policy <name>

Example 1:

```
add rewrite policy "pol_remove-ae" true "act_remove-ae"
 Done
> show rewrite policy pol_remove-ae
      Name: pol_remove-ae
      Rule: true
      RewriteAction: act_remove-ae
      UndefAction: Use Global
      Hits: 0
      Undef Hits: 0
      Bound to: GLOBAL RES_OVERRIDE
      Priority: 90
      GotoPriorityExpression: END
 Done
>
```

Example 2:

```
add cache policy BranchReportsCachePolicy -rule q{http.req.url.query.value("actionoverride").contains("branchReport s")} -action cache
Done
 show cache policy BranchReportsCachePolicy
      Name: BranchReportsCachePolicy
      Rule: http.req.url.query.value("actionoverride").contains("branchReports")
      CacheAction: CACHE
      Stored in group: DEFAULT
      UndefAction: Use Global
      Hits: 0
      Undef Hits: 0
 Done
```

Note: At the command line, quote marks within a policy rule (the expression) must be escaped or delimited with the q delimiter. For more information, see "Configuring Default Syntax Expressions in a Policy."

To create or modify a policy by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. In the navigation pane, expand the name of the feature for which you want to configure a policy, and then click Policies. For example, you can select Content Switching, Integrated Caching, DNS, Rewrite, or Responder.
2. In the details pane, click Add, or select an existing policy and click Open. A policy configuration dialog box appears.
3. Specify values for the following parameters. (An asterisk indicates a required parameter. For a term in parentheses, see the corresponding parameter in "Parameters for creating or modifying a policy.")
4. Click Create, and then click Close.
5. Click Save. A policy is added.
   Note: After you create a policy, you can view the policy's details by clicking the policy entry in the configuration pane. Details that are highlighted and underlined are links to the corresponding entity (for example, a named expression).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy Configuration Examples

Sep 02, 2013

These examples show how policies and their associated actions are entered at the command line interface. In the configuration utility, the expressions would appear in the Expression window of the feature-configuration dialog box for the integrated caching or rewrite feature.

Following is an example of creating a caching policy. Note that actions for caching policies are built in, so you do not need to configure them separately from the policy.

```
add cache policy BranchReportsCachePolicy -rule q{http.req.url.query.value("actionoverride").contains("branchReports")} -action cache
```

Following is an example of a Rewrite policy and action:

```
add rewrite action myAction1 INSERT_HTTP_HEADER "myHeader" "valueForMyHeader"
add rewrite policy myPolicy1 "http.req.url.contains(\"myURLstring\")" myAction1
```

Note: At the command line, quote marks within a policy rule (the expression) must be escaped or delimited with the q delimiter. For more information, see "Configuring Default Syntax Expressions in a Policy."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding Policies That Use the Default Syntax

Sep 30, 2013

After defining a policy, you indicate when the policy is to be invoked by binding the policy to a bind point and specifying a priority level. You can bind a policy to only one bind point. A bind point can be global, that is, it can apply to all virtual servers that you have configured. Or, a bind point can be specific to a particular virtual server, which can be either a load balancing or a content switching virtual server. Not all bind points are available for all features.

The order in which policies are evaluated determines the order in which they are applied, and features typically evaluate the various policy banks in a particular order. Sometimes, however, other features can affect the order of evaluation. Within a policy bank, the order of evaluation depends on the values of parameters configured in the policies. Most features apply all of the actions associated with policies whose evaluation results in a match with the data that is being processed. The integrated caching feature is an exception.

Feature-Specific Differences in Policy Bindings

You can bind policies to built-in, global bind points (or banks), to virtual servers, or to policy labels.

However, the NetScaler features differ in terms of the types of bindings that are available. The following table summarizes how you use policy bindings in various NetScaler features that use policies.

Table 1. Feature-Specific Bindings for Policies

Feature Name	Virtual Servers Configured in the Feature	Policies Configured in the Feature	Bind Points Configured for the Policies	Use of Policies in the Feature
DNS	none	DNS policies	Global	To determine how to perform DNS resolution for requests.
Content Switching Note: This feature can support either or classic policies or policies that use the default syntax, but not both.	Content Switching (CS)	Content Switching policies	• Content switching or cache redirection virtual server • Policy label	To determine what server or group of servers is responsible for serving responses, based on characteristics of an incoming request. Request characteristics include device type, language, cookies, HTTP method, content type, and associated cache server.
Integrated Caching	none	Caching policies	• Global override • Global default	To determine whether HTTP responses can be stored in, and served from, the NetScaler appliance's integrated cache.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature Name	Virtual Servers Configured in the Feature	Policies Configured in the Feature	Bind Points Configured for the Policies	Use of Policies in the Feature
			• Policy label • Load balancing, content switching, or SSL offload virtual server	
Responder	none	Responder policies	• Global override • Global default • Policy label • Load balancing, content switching, or SSL offload virtual server	To configure the behavior of the Responder function.
Rewrite	none	Rewrite policies	• Global override • Global default • Policy label • Load balancing, content switching, or SSL offload virtual server	To identify HTTP data that you want to modify before serving. The policies provide rules for modifying the data. For example, you can modify HTTP data to redirect a request to a selected server based on the address of the incoming request, or to mask server information in a response for security purposes.
URL Transform function in the Rewrite feature	none	Transformation policies	• Global override • Global default	To identify URLs in HTTP transactions and text files for the purpose of evaluating whether a URL should be altered.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature Name	Virtual Servers Configured in the Feature	Policies Configured in the Feature	Bind Points Configured for the Policies	Use of Policies in the Feature
NetScaler Gateway (clientless VPN functions only)	VPN server	Clientless Access policies	• Policy label • VPN Global • VPN server	To determine how the NetScaler Gateway performs authentication, authorization, auditing, and other functions, and to define rewrite rules for general Web access using the NetScaler Gateway.

Bind Points and Order of Evaluation

For a policy to take effect, you must ensure that the policy is invoked at some point during processing. To do so, you associate the policy with a bind point. The collection of policies that is bound to a bind point is known as a policy bank.

Following are the bind points that the NetScaler evaluates, listed in the typical order of evaluation within a policy bank

1. **Request-time override.** When a request flows through a feature, the NetScaler first evaluates request-time override policies for the feature.
2. **Request-time Load Balancing virtual server.** If policy evaluation cannot be completed after all the request-time override policies have been evaluated, the NetScaler processes request-time policies for load balancing virtual servers.
3. **Request-time Content Switching virtual server.** If policy evaluation cannot be completed after all the request-time policies for load balancing virtual servers have been evaluated, the NetScaler processes request-time policies for content switching virtual servers.
4. **Request-time default.** If policy evaluation cannot be completed after all request-time, virtual server-specific policies have been evaluated, the NetScaler processes request-time default policies.
5. **Response-time override.** At response time, the NetScaler starts with policies that are bound to the response-time override bind point.
6. **Response-time Load Balancing virtual server.** If policy evaluation cannot be completed after all response-time override policies have been evaluated, the NetScaler process the response-time policies for load balancing virtual servers.
7. **Response-time Content Switching virtual server.** If policy evaluation cannot be completed after all policies have been evaluated for load balancing virtual servers, the NetScaler process the response-time policies for content switching virtual servers.
8. **Response-time default.** If policy evaluation cannot be completed after all response-time, virtual-server-specific policies have been evaluated, the NetScaler processes response-time default policies.

Policy Evaluation across Features

In addition to attending to evaluation of policies within a feature, if you have bound policies to a content switching virtual server, note that these policies are evaluated before other policies. Binding a policy to a content switching vserver produces a different result in NetScaler versions 9.0.x and later than in 8.x versions. In NetScaler 9.0 and later versions, evaluation occurs as follows:

- Content switching policies are evaluated before other policies. If a content switching policy evaluates to TRUE, the target load balancing vserver is selected.
- If all content switching policies evaluate to FALSE, the default load balancing vserver under the content switching VIP is selected.

After a target load balancing vserver is selected by the content switching process, policies are evaluated in the following

© 1999-2017 Citrix Systems, Inc. All rights reserved.

order:

1. Policies that are bound to the global override bind point.
2. Policies that are bound to the default load balancing vserver.
3. Policies that are bound to the target content switching vserver.
4. Policies that are bound to the global default bind point.

To be sure that the policies are evaluated in the intended order, follow these guidelines:

- Make sure that the default load balancing vserver is not directly reachable from the outside; for example, the vserver IP address can be 0.0.0.0.
- To prevent exposing internal data on the load balancing default vserver, configure a policy to respond with a "503 Service Unavailable" status and bind it to the default load balancing vserver.

Entries in a Policy Bank

Each entry in a policy bank has, at minimum, a policy and a priority level. You can also configure entries that change the priority-based evaluation order, and you can configure entries that invoke external policy banks.

The following table summarizes each entry in a policy bank.

Table 2. Format of Each Entry in a Policy Bank

Policy Name	Priority	Goto Expression	Invocation Type	Policy Bank to Be Invoked
The policy name, or a "dummy" policy named NOPOLICY. The NOPOLICY entry controls evaluation flow without processing a rule.	An integer.	Optional. Identifies the next policy in the bank to evaluate, or ends any further evaluation	Optional. Indicates that an external policy bank will be invoked. This field restricts the choices to a global policy label or a virtual server.	Optional. Used with Invocation Type. This is the label for a policy bank or a virtual server name. The NetScaler returns to the current bank after processing the external bank.

If the policy evaluates to TRUE, the NetScaler stores the action that is associated with the policy. If the policy evaluates to FALSE, the NetScaler evaluates the next policy. If the policy is neither TRUE nor FALSE, the NetScaler uses the associated Undef (undefined) action.

Evaluation Order within a Policy Bank

Within a policy bank, the evaluation order depends on the following items:

A priority.

The most minimal amount of information about evaluation order is a numeric priority level. The lower the number, the higher the priority.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

A Goto expression.

If supplied, the Goto expression indicates the next policy to be evaluated, typically within the same policy bank.. Goto expressions can only proceed forward in a bank. To prevent looping, a policy bank configuration is not valid if a Goto statement points backwards in the bank.

Invocation of other policy banks.

Any entry can invoke an external policy bank. The NetScaler provides a built-in entity named NOPOLICY that does not have a rule. You can add a NOPOLICY entry in a policy bank when you want to invoke another policy bank, but do not want to process any other rules prior to the invocation. You can have multiple NOPOLICY entries in multiple policy banks.

Values for a Goto expression are as follows:

NEXT.

This keyword selects the policy with the next higher priority level in the current policy bank.

An integer.

If you supply an integer, it must match the priority level of another policy in the current policy bank.

END.

This keyword stops evaluation after processing the current policy, and no additional policies in this bank are processed.

Blank.

If the Goto expression is empty, it is the same as specifying END.

A numeric expression.

This is a default syntax expression that resolves to a priority number for another policy in the current bank.

USE_INVOCATION_RESULT.

This phrase can be used only if you are invoking an external policy bank. Entering this phrase causes the NetScaler to perform one of the following actions:

- If the final Goto in the invoked policy bank has a value of END or is empty, the invocation result is END, and evaluation stops.
- If the final Goto expression in the invoked policy bank is anything other than END, the NetScaler performs a NEXT.

The following table illustrates a policy bank that uses Goto statements and policy bank invocations.

Table 3. Example of a Policy Bank That Uses Gotos and External Bank Invocations

Policy Name	Priority	Goto	Invocation	Policy Bank to Be Invoked
ClientCertificatePolicy (rule: does the request contain a client certificate?)	100	300	None	None
SubnetPolicy (rule: is the client from a private subnet?)	200	NEXT	None	None
NOPOLICY	300	USE INVOCATION RESULT	Request vserver	My_Request_VServer

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy Name	Priority	Goto	Invocation	Policy Bank to Be Invoked
NOPOLICY	350	USE INVOCATION RESULT	Policy Label	My_Policy_Label
WorkingHoursPolicy (rule: is it working hours?)	400	END	None	None

How Policy Evaluation Ends

Evaluation of a policy bank ends when one of the following takes place:

- A policy evaluates to TRUE and its Goto statement value is END.
  No further policies or policy banks in this feature are evaluated.
- An external policy bank is invoked, its evaluation returns an END, and the Goto statement uses a value of USE_INVOCATION_RESULT or END.
  Evaluation continues with the next policy bank for this feature. For example, if the current bank is the request-time override bank, the NetScaler next evaluates request-time policy banks for the virtual servers.
- The NetScaler has walked through all the policy banks in this feature, but has not encountered an END.
  If this is the last entry to be evaluated in this policy bank, the NetScaler proceeds to the next feature.

How Features Use Actions after Policy Evaluation

After evaluating all relevant policies for a particular data point (for example, an HTTP request), the NetScaler stores all the actions that are associated with any policy that matched the data.

For most features, all the actions from matching policies are applied to a traffic packet as it leaves the NetScaler. The Integrated Caching feature only applies one action: CACHE or NOCACHE. This action is associated with the policy with the lowest priority value in the “highest priority” policy bank (for example, request-time override policies are applied before virtual server-specific policies).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding a Policy Globally

Nov 14, 2013

The following binding procedures are typical. However, refer to the documentation for the feature of interest to you for complete instructions.

To bind an Integrated Caching policy globally by using the command line interface

At the command prompt, type the following commands to bind an Integrated Caching policy and verify the configuration:

- bind cache global <policy> -priority <positiveInteger> [-type REQ_OVERRIDE | REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT]
- show cache global

  Example

  ```
  bind cache global _nonPostReq -priority 100 -type req_default
   Done
  > show cache global
  1)    Global bindpoint: REQ_DEFAULT
        Number of bound policies: 2

  2)    Global bindpoint: RES_DEFAULT
        Number of bound policies: 1
   Done
  ```

  The type argument is optional to maintain backward compatibility. If you omit the type, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression.

To bind a Rewrite policy globally by using the command line interface

At the command prompt, type the following commands to bind a Rewrite policy and verify the configuration:

- bind rewrite global <policyName> <priority> [-type REQ_OVERRIDE | REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT]
- show rewrite global

  Example

  ```
  bind rewrite global pol_remove-pdf 100
   Done
  > show rewrite global
  1)    Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

  2)    Global bindpoint: REQ_OVERRIDE
        Number of bound policies: 1

   Done
  ```

  The type argument is optional for globally bound policies, to maintain backward compatibility. If you omit the type, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression.

To bind a compression policy globally by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to bind a compression policy and verify the configuration:

- bind cmp global <policyName> -priority <positiveInteger> [-type REQ_OVERRIDE | REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT]
- show cmp global

Example

```
> bind cmp global cmp_pol_1 -priority 100
 Done
> show cmp policy cmp_pol_1
      Name: cmp_pol_1
      Rule: HTTP.REQ.URL.SUFFIX.EQ("BMP")
      Response Action: COMPRESS
      Hits: 0

      Policy is bound to following entities
      1) GLOBAL REQ_DEFAULT
      Priority: 100
      GotoPriorityExpression: END
 Done
>
```

To bind a Responder policy globally by using the command line interface

At the command prompt, type the following commands to bind a Responder policy and verify the configuration:

- bind responder global <policyName> <priority> [-type OVERRIDE | DEFAULT]
- show responder global

Example

```
bind responder global pol404Error1 200
Done
> show responder global
1)    Global bindpoint: REQ_DEFAULT
      Number of bound policies: 1

Done
```

To bind a DNS policy globally by using the command line interface

At the command prompt, type the following commands to bind a DNS policy and verify the configuration:

- bind dns global <policyName> <priority>
- show dns global

Example

```
> bind dns global pol_ddos_drop1 150
 Done
> show dns global
Policy name : pol_ddos_drop
      Priority : 100
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        Goto expression : END
    Policy name : pol_ddos_drop1
        Priority : 150
     Done
    >
```

To bind an Integrated Caching, Responder, Rewrite, or Compression policy globally by using the configuration utility

1. In the navigation pane, click the name of the feature for which you want to bind the policy.
2. In the details pane, click <Feature Name> policy manager.
3. In the Policy Manager dialog box, select the bind point to which you want to bind the policy (for example, for Integrated Caching, Rewrite, or Compression, you could select Request and Default Global). The Responder does not differentiate between request-time and response-time policies.
4. Click Insert Policy and, from the Policy Name pop-up menu, select the policy name. A priority is assigned automatically to the policy, but you can click the cell in the Priority column and drag it anywhere within the dialog box if you want the policy to be evaluated after other policies in this bank. The priority is automatically reset. Note that priority values within a policy bank must be unique.
5. Click Apply Changes.
6. Click Close. A message in the status bar indicates that the policy is bound successfully.

To bind a DNS policy globally by using the configuration utility

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings.
3. In the global bindings dialog box, click Insert Policy, and select the policy that you want to bind globally.
4. Click in the Priority field and enter the priority level.
5. Click OK. A message in the status bar indicates that the policy is bound successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding a Policy to a Virtual Server

Nov 14, 2013

A globally bound policy applies to all load balancing and content switching virtual servers.

Note that when binding a policy to a virtual server, you must identify it as a request-time or a response-time policy.

To bind a policy to a load balancing or content switching virtual server by using the command line interface

At the command prompt, type the following commands to bind a policy to a load balancing or content switching virtual server and verify the configuration:

- bind lb | cs vserver <virtualServerName> -policyName <policyName> -priority <positiveInteger> -type REQUEST | RESPONSE
- show lb vserver <name>

  Example

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 50
 Done
> show lb vserver lbvip
      lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
      State: DOWN
      Last state change was at Wed Jul 15 05:54:24 2009 (+226 ms)
      Time since last state change: 28 days, 01:57:26.350
      Effective State: DOWN
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
      No. of Bound Services :  0 (Total)       0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Vserver IP and Port insertion: OFF
      Push: DISABLED  Push VServer:
      Push Multi Clients: NO
      Push Label Rule: none

 1)     Policy : ns_cmp_msapp Priority:50
 2)     Policy : cf-pol Priority:1      Inherited
 Done
```

To bind a policy to an SSL offload virtual server by using the command line interface

At the command prompt, type the following commands to bind a policy to an SSL offload virtual server and verify the configuration:

bind ssl vserver <vServerName>@ -policyName <policyName> -priority <positiveInteger>

To bind a policy to a virtual server by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. In the navigation pane, expand Traffic Management > Load Balancing, Traffic Management > Content Switching, Traffic Management > SSL Offload, Security > AAA- Application Traffic, or NetScaler Gateway, and then click Virtual Servers.
2. In the details pane, double-click the virtual server to which you want to bind the policy, and then click Open.
3. On the Policies tab, click the icon for the type of policy that you want to bind (the choices are feature-specific), and then click the name of the policy. Note that for some features, you can bind both classic policies and policies that use the default syntax to the virtual server.
4. If you are binding a policy to a Content Switching virtual server, in the Target field select a load balancing virtual server to which traffic that matches the policy is sent.
5. Click OK. A message in the status bar indicates that the policy is bound successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Displaying Policy Bindings

Oct 29, 2013

You can display policy bindings to verify that they are correct.

To display policy bindings by using the command line interface

At the command prompt, type the following commands to display policy bindings and verify the configuration:
show rewrite policy <name>

Example

```
> show rewrite policy pol_remove-pdf
      Name: pol_remove-pdf
      Rule: http.req.url.contains(".pdf")
      RewriteAction: act_remove-ae
      UndefAction: Use Global
      Hits: 0
      Undef Hits: 0
      Bound to: GLOBAL REQ_DEFAULT
      Priority: 100
      GotoPriorityExpression: END
 Done
>
```

To display global policy bindings for Integrated Caching, Rewrite, or Responder by using the configuration utility

1. In the navigation pane, expand the feature that contains the policy that you want to view, and then click Policies.
2. In the details pane, click the policy. Bound policies have a check mark next to them.
3. At the bottom of the page, under Details, next to Bound to, view the entity to which the policy is bound.

To display global policy bindings for DNS or Clientless Access in the NetScaler Gateway by using the configuration utility

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings.

To display global policy bindings for Content Switching by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Policies.
2. In detailed pane, select policy.
3. In the details pane, click Show Bindings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Unbinding a Policy

Nov 14, 2013

If you want to re-assign a policy or delete it, you must first remove its binding.

To unbind an integrated caching, rewrite, or compression default syntax policy globally by using the command line interface

At the command prompt, type the following commands to unbind an integrated caching, rewrite, or compression default syntax policy globally and verify the configuration:

- unbind cache | rewrite | cmp global <policyName> [-type req_override | req_default | res_override | res_default] [-priority <positiveInteger>]
- show cache | rewrite | cmp global

  Example

  ```
  > unbind cache global_nonPostReq
   Done
  > show cache global
  1)    Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

  2)    Global bindpoint: RES_DEFAULT
        Number of bound policies: 1

   Done
  ```

  The priority is required only for the “dummy” policy named NOPOLICY.

To unbind a responder policy globally by using the command line interface

At the command prompt, type the following commands to unbind a responder policy globally and verify the configuration:

- unbind responder global <policyName> [-type override | default] [-priority <positiveInteger>]
- show responder global

  Example

  ```
  > unbind responder global pol404Error
   Done
  > show responder global
  1)    Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

   Done
  ```

  The priority is required only for the “dummy” policy named NOPOLICY.

To unbind a DNS policy globally by using the command line interface

At the command prompt, type the following commands to unbind a DNS policy globally and verify the configuration:

- unbind responder global <policyName>
- unbind responder global

  Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
unbind dns global dfgdfg
Done
show dns global
Policy name : dfgdfggfhg
    Priority : 100
    Goto expression : END
Done
```

To unbind a default syntax policy from a virtual server by using the command line interface

At the command prompt, type the following commands to unbind a default syntax policy from a virtual server and verify the configuration:

- unbind cs vserver <name> -policyName <policyName> [-priority <positiveInteger>] [-type REQUEST | RESPONSE]
- show lb vserver <name>

Example

```
unbind cs vserver vs-cont-switch  -policyName pol1
Done
> show cs vserver vs-cont-switch
    vs-cont-switch (10.102.29.10:80) - HTTP Type: CONTENT
    State: UP
    Last state change was at Wed Aug 19 08:56:55 2009 (+18 ms)
    Time since last state change: 0 days, 02:47:55.750
    Client Idle Timeout: 180 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    Port Rewrite : DISABLED
    State Update: DISABLED
    Default:       Content Precedence: RULE
    Vserver IP and Port insertion: OFF
    Case Sensitivity: ON
    Push: DISABLED  Push VServer:
    Push Label Rule: none
Done
```

The priority is required only for the “dummy” policy named NOPOLICY.

To unbind an integrated caching, responder, rewrite, or compression default syntax policy globally by using the configuration utility

1. In the navigation pane, click the feature with the policy that you want to unbind (for example, Integrated Caching).
2. In the details pane, click <Feature Name> policy manager.
3. In the Policy Manager dialog box, select the bind point with the policy that you want to unbind, for example, Default Global.
4. Click the policy name that you want to unbind, and then click Unbind Policy.
5. Click Apply Changes.
6. Click Close. A message in the status bar indicates that the policy is unbound successfully.

To unbind a DNS policy globally by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings.
3. In the Global Bindings dialog box, select policy and click unbind policy.
4. Click OK. A message in the status bar indicates that the policy is unbinded successfully.

To unbind a default syntax policy from a load balancing or content switching virtual server by using the configuration utility

1. Navigate to Traffic Management, and expand Load Balancing or Content Switching, and then click Virtual Servers.
2. In the details pane, double-click the virtual server from which you want to unbind the policy.
3. On the Policies tab, in the Active column, clear the check box next to the policy that you want to unbind.
4. Click OK. A message in the status bar indicates that the policy is unbinded successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating Policy Labels

May 25, 2015

In addition to the built-in bind points where you set up policy banks, you can also configure user-defined policy labels and associate policies with them.

Within a policy label, you bind policies and specify the order of evaluation of each policy relative to others in the bank of policies for the policy label. The NetScaler also permits you to define an arbitrary evaluation order as follows:

- You can use “goto” expressions to point to the next entry in the bank to be evaluated after the current one.
- You can use an entry in a policy bank to invoke another bank.

This document includes the following details:

- Creating Policy Labels
- Binding a Policy to a Policy Label

Creating Policy Labels

Updated: 2013-11-14

Each feature determines the type of policy that you can bind to a policy label, the type of load balancing virtual server that you can bind the label to, and the type of content switching virtual server from which the label can be invoked. For example, a TCP policy label can only be bound to a TCP load balancing virtual server. You cannot bind HTTP policies to a policy label of this type. And you can invoke a TCP policy label only from a TCP content switching virtual server.

After configuring a new policy label, you can invoke it from one or more banks for the built-in bind points.

To create a caching policy label by using the command line interface

At the command prompt, type the following commands to create a Caching policy label and verify the configuration:

- add cache policylabel <labelName> -evaluates req|res
- show cache policylabel<labelName>

  Example

```
> add cache policylabel lbl-cache-pol -evaluates req
 Done

> show cache policylabel lbl-cache-pol
      Label Name: lbl-cache-pol
      Evaluates: REQ
      Number of bound policies: 0
      Number of times invoked: 0
 Done
>
```

To create a Content Switching policy label by using the command line interface

At the command prompt, type the following commands to create a Content Switching policy label and verify the configuration:

- add cs policylabel <labelName> http|tcp|rtsp|ssl

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- show cs policylabel <labelName>

  Example

```
> add cs policylabel lbl-cs-pol http
 Done
> show cs policylabel lbl-cs-pol
      Label Name: lbl-cs-pol
      Label Type: HTTP
      Number of bound policies: 0
      Number of times invoked: 0
 Done
```

To create a Rewrite policy label by using the command line interface

At the command prompt, type the following commands to create a Rewrite policy label and verify the configuration:

- add rewrite policylabel <labelName> http_req|http_res|url|text|clientless_vpn_req|clientless_vpn_res
- show rewrite policylabel <labelName>

  Example

```
> add rewrite policylabel lbl-rewrt-pol http_req
 Done

> show rewrite policylabel lbl-rewrt-pol
      Label Name: lbl-rewrt-pol
      Transform Name: http_req
      Number of bound policies: 0
      Number of times invoked: 0
 Done
```

To create a Responder policy label by using the command line interface

At the command prompt, type the following commands to create a Responder policy label and verify the configuration:

- add responder policylabel <labelName>
- show responder policylabel <labelName>

  Example

```
> add responder policylabel lbl-respndr-pol
 Done

> show responder policylabel lbl-respndr-pol
      Label Name: lbl-respndr-pol
      Number of bound policies: 0
      Number of times invoked: 0
 Done
```

Note: Invoke this policy label from a policy bank. For more information, see "Binding a Policy to a Policy Label."

To create a policy label by using the configuration utility

1. In the navigation pane, expand the feature for which you want to create a policy label, and then click Policy Labels. The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

choices are Integrated Caching, Rewrite, Content Switching, or Responder.
2. In the details pane, click Add.
3. In the Name box, enter a unique name for this policy label.
4. Enter feature-specific information for the policy label. For example, for Integrated Caching, in the Evaluates drop-down menu, you would select REQ if you want this policy label to contain request-time policies, or select RES if you want this policy label to contain response-time policies. For Rewrite, you would select a Transform name.
5. Click Create.
6. Configure one of the built-in policy banks to invoke this policy label. For more information, see "Binding a Policy to a Policy Label."
   A message in the status bar indicates that the policy label is created successfully.

Binding a Policy to a Policy Label

As with policy banks that are bound to the built-in bind points, each entry in a policy label is a policy that is bound to the policy label. As with policies that are bound globally or to a vserver, each policy that is bound to the policy label can also invoke a policy bank or a policy label that is evaluated after the current entry has been processed. The following table summarizes the entries in a policy label.

Name

The name of a policy, or, to invoke another policy bank without evaluating a policy, the “dummy” policy name NOPOLICY. You can specify NOPOLICY more than once in a policy bank, but you can specify a named policy only once.

Priority

An integer. This setting can work with the Goto expression.

Goto Expression

Determines the next policy to evaluate in this bank. You can provide one of the following values:

NEXT:

Go to the policy with the next higher priority.

END:

Stop evaluation.

USE_INVOCATION_RESULT:

Applicable if this entry invokes another policy bank. If the final Goto in the invoked bank has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy bank performs a NEXT.

Positive number:

The priority number of the next policy to be evaluated.

Numeric expression:

An expression that produces the priority number of the next policy to be evaluated.

The Goto can only proceed forward in a policy bank.

If you omit the Goto expression, it is the same as specifying END.

Invocation Type

Designates a policy bank type. The value can be one of the following:

Request Vserver:

Invokes request-time policies that are associated with a virtual server.

Response Vserver:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Invokes response-time policies that are associated with a virtual server.

Policy label:

Invokes another policy bank, as identified by the policy label for the bank.

Invocation Name

The name of a virtual server or a policy label, depending on the value that you specified for the Invocation Type.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Policy Label or Virtual Server Policy Bank

May 25, 2015

After you have created policies, and created policy banks by binding the policies, you can perform additional configuration of policies within a label or policy bank. For example, before you configure invocation of an external policy bank, you might want to wait until you have configured that policy bank.

This document includes the following details:

- Configuring a Policy Label
- Configuring a Policy Bank for a Virtual Server

Configuring a Policy Label

Updated: 2013-11-14

A policy label consists of a set of policies and invocations of other policy labels and virtual server-specific policy banks. An Invoke parameter enables you to invoke a policy label or a virtual server-specific policy bank from any other policy bank. A special-purpose NoPolicy entry enables you to invoke an external bank without processing an expression (a rule). The NoPolicy entry is a “dummy” policy that does not contain a rule.

For configuring policy labels from the NetScaler command line, note the following elaborations of the command syntax:

- gotoPriorityExpression is configured as described in "Entries in a Policy Bank."
- The type argument is required. This is unlike binding a conventional policy, where this argument is optional.
- You can invoke the bank of policies that are bound to a virtual server by using the same method as you use for invoking a policy label.

To configure a policy label by using the command line interface

At the command prompt, type the following commands to configure a policy label and verify the configuration:

- bind cache | rewrite | responder policylabel <policylabelName> -policyName <policyName> -priority <priority> [-gotoPriorityExpression <gotopriorityExpression>] [-invoke reqvserver | resvserver | policylabel <policyLabelName> | <vserverName>]
- show cache | rewrite | responder policylabel <policylabelName>

  Example

```
bind cache policylabel _reqBuiltinDefaults -policyName  _nonGetReq -priority 100
Done
show cache policylabel _reqBuiltinDefaults
    Label Name: _reqBuiltinDefaults
    Evaluates: REQ
    Number of bound policies: 3
    Number of times invoked: 0
1)   Policy Name: _nonGetReq
     Priority: 100
     GotoPriorityExpression: END
2)   Policy Name: _advancedConditionalReq
     Priority: 200
     GotoPriorityExpression: END

3)   Policy Name: _personalizedReq
     Priority: 300
     GotoPriorityExpression: END
Done
```

To invoke a policy label from a Rewrite policy bank with a NOPOLICY entry by using the command line interface

At the command prompt, type the following commands to invoke a policy label from a Rewrite policy bank with a NOPOLICY entry and verify the configuration:

- bind rewrite global <policyName> <priority> <gotoPriorityExpression> -type REQ_OVERRIDE | REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT -invoke reqvserver | resvserver | policylabel <policyLabelName> | <vserverName>
- show rewrite global

  Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> bind rewrite global NOPOLICY 100 -type REQ_DEFAULT -invoke policylabel lbl-rewrt-pol
 Done
> show rewrite global
1)     Global bindpoint: REQ_DEFAULT
       Number of bound policies: 1

2)     Global bindpoint: REQ_OVERRIDE
       Number of bound policies: 1
 Done
```

To invoke a policy label from an Integrated Caching policy bank by using the command line interface

At the command prompt, type the following commands to invoke a policy label from an Integrated Caching policy bank and verify the configuration:

- bind cache global NOPOLICY -priority <priority> -gotoPriorityExpression <gotopriorityExpression> -type REQ_OVERRIDE | REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT -invoke reqvserver | resvserver | policylabel <policyLabelName> | <vserverName>
- show cache global

Example

```
 bind cache global NOPOLICY -priority 100 -gotoPriorityExpression END -type REQ_DEFAULT -invoke policylabel lbl-cache-pol
 Done
> show cache global
1)     Global bindpoint: REQ_DEFAULT
       Number of bound policies: 2

2)     Global bindpoint: RES_DEFAULT
       Number of bound policies: 1

 Done
```

To invoke a policy label from a Responder policy bank by using the command line interface

At the command prompt, type the following commands to invoke a policy label from a Responder policy bank and verify the configuration:

- bind responder global NOPOLICY <priority> <gotopriorityExpression> -type OVERRIDE | DEFAULT -invoke vserver | policylabel <policyLabelName> | <vserverName>
- show responder global

Example

```
> bind responder global NOPOLICY 100 NEXT -type DEFAULT -invoke policylabel lbl-respndr-pol
 Done
> show responder global
1)     Global bindpoint: REQ_DEFAULT
       Number of bound policies: 2

 Done
```

To configure a policy label by using the configuration utility

1. In the navigation pane, expand the feature for which you want to configure a policy label, and then click Policy Labels. The choices are Integrated Caching, Rewrite, or Responder.
2. In the details pane, double-click the label that you want to configure.
3. If you are adding a new policy to this policy label, click Insert Policy, and in the Policy Name field, select New Policy. For more information about adding a policy, see "Creating or Modifying a Policy." Note that if you are invoking a policy bank, and do not want a rule to be evaluated prior to the invocation, click Insert Policy, and in the Policy Name field select NOPOLICY.
4. For each entry in this policy label, configure the following:

   Policy Name:

   This is already determined by the Policy Name, new policy, or NOPOLICY entry that you inserted in this bank.

   Priority:

   A numeric value that determines either an absolute order of evaluation within the bank, or is used in conjunction with a Goto expression.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression:

The policy rule. Policy expressions are described in detail in the following chapters. For an introduction, see "Configuring Default Syntax Expressions: Getting Started."

Action:

The action to be taken if this policy evaluates to TRUE.

Goto Expression:

Optional. Used to augment the Priority level to determine the next policy or policy bank to evaluate. For more information on possible values for a Goto expression, see the table "Entries in a Policy Bank."

Invoke:

Optional. Invokes another policy bank.

5. Click Ok. A message in the status bar indicates that the policy label is configured successfully.

Configuring a Policy Bank for a Virtual Server

Updated: 2013-09-02

You can configure a bank of policies for a virtual server. The policy bank can contain individual policies, and each entry in the policy bank can optionally invoke a policy label or a bank of policies that you configured for another virtual server. If you invoke a policy label or policy bank, you can do so without triggering an expression (a rule) by selecting a NOPOLICY “dummy” entry instead of a policy name.

To add policies to a virtual server policy bank by using the command line interface

At the command prompt, type the following commands to add policies to a virtual server policy bank and verify the configuration:

- bind lb|cs vserver <virtualServerName> <serviceType> [-policyName <policyName>] [-priority <positiveInteger>] [-gotoPriorityExpression <expression>] [-type REQUEST|RESPONSE]
- show lb|cs vserver <virtualServerName>

Example

```
add lb vserver vs-cont-sw TCP
Done
show lb vserver vs-cont-sw
    vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
    State: DOWN
    Last state change was at Wed Aug 19 10:04:02 2009 (+279 ms)
    Time since last state change: 0 days, 00:02:14.420
    Effective State: DOWN
    Client Idle Timeout: 9000 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    No. of Bound Services :  0 (Total)      0 (Active)
    Configured Method: LEASTCONNECTION
    Mode: IP
    Persistence: NONE
    Connection Failover: DISABLED
Done
```

To invoke a policy label from a virtual server policy bank with a NOPOLICY entry by using the command line interface

At the command prompt, type the following commands to invoke a policy label from a virtual server policy bank with a NOPOLICY entry and verify the configuration:

- bind lb|cs vserver <virtualServerName> -policyName NOPOLICY_REWRITE|NOPOLICY_CACHE|NOPOLICY_RESPONDER -priority <integer> -type REQUEST|RESPONSE -gotoPriorityExpression <gotopriorityExpression> -invoke reqVserver|resVserver|policyLabel <vserverName>|<labelName>
- show lb vserver

Example

```
> bind lb vserver vs-cont-sw -policyname NOPOLICY-REWRITE -priority 200 -type REQUEST -gotoPriorityExpression NEXT -invoke policyLabel lbl-rewrt-pol
Done
```

To configure a virtual server policy bank by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. In the left navigation pane, expand Traffic Management > Load Balancing, Traffic Management > Content Switching, Traffic Management > SSL Offload, Security > AAA - Application Traffic, or NetScaler Gateway, as appropriate, and then click Virtual Servers.
2. In the details pane, select the virtual server that you want to configure, and then click Open.
3. In the Configure Virtual Server dialog box click the Policies tab.
4. To create a new policy in this bank, click the icon for the type of policy or policy label that you want to add to the virtual server's bank of policies, click Insert Policy. Note that if you want to invoke a policy label without evaluating a policy rule, select the NOPOLICY "dummy" policy.
5. To configure an existing entry in this policy bank, enter the following:

   Priority:

   A numeric value that determines either an absolute order of evaluation within the bank or is used in conjunction with a Goto expression.

   Expression:

   The policy rule. Policy expressions are described in detail in the following chapters. For an introduction, see "Configuring Default Syntax Expressions: Getting Started."

   Action: on:

   The action to be taken if this policy evaluates to TRUE.

   Goto Expression:

   Optional. Determines the next policy or policy bank to evaluate. For more information on possible values for a Goto expression, see "Entries in a Policy Bank."

   Invoke:

   Optional. To invoke another policy bank, select the name of the policy label or virtual server policy bank that you want to invoke.

6. When you are done, click OK. A message in the status bar indicates that the policy is configured successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Invoking or Removing a Policy Label or Virtual Server Policy Bank

Nov 14, 2013

Unlike a policy, which can only be bound once, you can use a policy label or a virtual server's policy bank any number of times by invoking it. Invocation can be performed from two places:

- From the binding for a named policy in a policy bank.
- From the binding for a NOPOLICY "dummy" entry in a policy bank.

Typically, the policy label must be of the same type as the policy from which it is invoked. For example, you would invoke a responder policy label from a responder policy.

Note: When binding or unbinding a global NOPOLICY entry in a policy bank at the command line, you specify a priority to distinguish one NOPOLICY entry from another.

To invoke a rewrite or integrated caching policy label by using the command line interface

At the command prompt, type the one of the following commands to invoke a rewrite or integrated caching policy label and verify the configuration:

- bind cache global <policy> -priority <positive_integer> [**-gotoPriorityExpression** <expression>] **-type REQ_OVERRIDE**|**REQ_DEFAULT** |**RES_OVERRIDE**|**RES_DEFAULT**] **-invoke reqvserver**|**resvserver**|**policylabel** <label_name>
- bind rewrite global<policy> -priority <positive_integer> [**-gotoPriorityExpression** <expression>] **-type REQ_OVERRIDE**|**REQ_DEFAULT** |**RES_OVERRIDE**|**RES_DEFAULT**] **-invoke reqvserver**|**resvserver**|**policylabel** <label_name>
- show cache global|show rewrite global

  Example

  ```
  > bind cache global _nonPostReq2 -priority 100 -type req_override -invoke
  policylabel lbl-cache-pol
   Done
  > show cache global
  1)    Global bindpoint: REQ_DEFAULT
        Number of bound policies: 2

  2)    Global bindpoint: RES_DEFAULT
        Number of bound policies: 1

  3)    Global bindpoint: REQ_OVERRIDE
        Number of bound policies: 1

   Done
  ```

To invoke a responder policy label by using the command line interface

At the command prompt, type the following commands to invoke a responder policy label and verify the configuration:

- **bind responder global** <policy_Name> <priority_as_positive_integer> [<gotoPriorityExpression>] **-type REQ_OVERRIDE**|**REQ_DEFAULT**|**OVERRIDE**|**DEFAULT -invoke vserver**|**policylabel** <label_name>
- **show responder global**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

```
> bind responder global pol404Error1 300 -invoke policylabel lbl-respndr-pol
 Done
> show responder global
1)     Global bindpoint: REQ_DEFAULT
       Number of bound policies: 2

 Done
>
```

To invoke a Virtual Server Policy Bank by using the command line interface

At the command prompt, type the following commands to invoke a Virtual Server Policy Bank and verify the configuration:

- **bind lb vserver** <vserver_name> **-policyName** <policy_Name> **-priority** <positive_integer> [**-gotoPriorityExpression** <expression>] **-type REQUEST**|**RESPONSE -invoke reqvserver**|**resvserver**|**policylabel** <policy_Label_Name>
- **bind lb vserver** <vserver_name>

Example

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 100
 Done

> show lb vserver lbvip
      lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
      State: DOWN
      Last state change was at Wed Jul 15 05:54:24 2009 (+166 ms)
      Time since last state change: 28 days, 06:37:49.250
      Effective State: DOWN
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
      No. of Bound Services :  0 (Total)       0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Vserver IP and Port insertion: OFF
      Push: DISABLED  Push VServer:
      Push Multi Clients: NO
      Push Label Rule: none

1)     CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority: 100   Hits: 0

1)     Policy : pol-ssl Priority:0
2)     Policy : ns_cmp_msapp Priority:100
3)     Policy : cf-pol Priority:1      Inherited
 Done
>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To remove a rewrite or integrated caching policy label by using the command line interface

At the command prompt, type one of the following commands to remove a rewrite or integrated caching policy label and verify the configuration:

- unbind rewrite global <policyName> -priority <positiveInteger> -type **REQ_OVERRIDE**|**REQ_DEFAULT**|**RES_OVERRIDE**|**RES_DEFAULT**
- unbind cache global <policyName> -priority <positiveInteger> -type **REQ_OVERRIDE**|**REQ_DEFAULT**|**RES_OVERRIDE**|**RES_DEFAULT**
- show rewrite global|show cache global

  Example

```
> unbind rewrite global NOPOLICY -priority 100 -type REQ_OVERRIDE
 Done
> show rewrite global
1)     Global bindpoint: REQ_DEFAULT
       Number of bound policies: 1

 Done
```

To remove a responder policy label by using the command line interface

At the command prompt, type the following commands to remove a responder policy label and verify the configuration:

- unbind responder global <policyName> -priority <positiveInteger> **-type OVERRIDE**|**DEFAULT**
- **show responder global**

  Example

```
> unbind responder global NOPOLICY -priority 100 -type REQ_DEFAULT
 Done
> show responder global
1)     Global bindpoint: REQ_DEFAULT
       Number of bound policies: 1

 Done
```

To remove a Virtual Server policy label by using the command line interface

At the command prompt, type one of the following commands to remove a Virtual Server policy label and verify the configuration:

- unbind lb vserver <virtualServerName> -policyName **NOPOLICY-REWRITE**|**NOPOLICY-RESPONDER**|**NOPOLICY-CACHE -type REQUEST**|**RESPONSE -priority** <positiveInteger>
- unbind cs vserver <virtualServerName> -policyName **NOPOLICY-REWRITE**|**NOPOLICY-RESPONDER**|**NOPOLICY-CACHE -type REQUEST**|**RESPONSE -priority** <positiveInteger>
- show lb vserver|show cs vserver

  Example

```
> unbind lb vserver lbvip -policyName ns_cmp_msapp -priority 200
 Done
> show lb vserver lbvip
       lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
       State: DOWN
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Last state change was at Wed Jul 15 05:54:24 2009 (+161 ms)
Time since last state change: 28 days, 06:47:54.600
Effective State: DOWN
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Port Rewrite : DISABLED
No. of Bound Services :  0 (Total)      0 (Active)
Configured Method: LEASTCONNECTION
Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED  Push VServer:
Push Multi Clients: NO
Push Label Rule: none

1)     CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority: 100   Hits: 0

1)     Policy : pol-ssl Priority:0
2)     Policy : cf-pol Priority:1      Inherited
 Done
```

To invoke a policy label or virtual server policy bank by using the configuration utility

1. Bind a policy, as described in "Binding a Policy Globally", "Binding a Policy to a Virtual Server", or "Binding a Policy to a Policy Label." Alternatively, you can enter a NOPOLICY “dummy” entry instead of a policy name. You do this if you do not want to evaluate a policy before evaluating the policy bank.
2. In the Invoke field, select the name of the policy label or virtual server policy bank that you want to evaluate if traffic matches the bound policy. A message in the status bar indicates that the policy label or virtual server policy bank is invoked successfully.

To remove a policy label invocation by using the configuration utility

1. Open the policy and clear the Invoke field. Unbinding the policy also removes the invocation of the label. A message in the status bar indicates that the policy label is removed successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring and Binding Policies with the Policy Manager

Sep 02, 2013

Some applications provide a specialized Policy Manager in the NetScaler configuration utility to simplify configuring policy banks. It also lets you find and delete policies and actions that are not being used.

The Policy Manager is currently available for the Rewrite, Integrated Caching, Responder, and Compression features.

The following are keyboard equivalents for the procedures in this section:

- For editing a cell in the Policy Manager, you can tab to the cell and click F2 or press the SPACE bar on the keyboard.
- To select an entry in a drop-down menu, you can tab to the entry, press the space bar to view the drop-down menu, use the UP and DOWN ARROW keys to navigate to the entry that you want, and press the space bar again to select the entry.
- To cancel a selection in a drop-down menu, press the Escape key.
- To insert a policy, tab to the row above the insertion point and press Control + Insert, or click Insert Policy.
- To remove a policy, tab to the row that contains the policy and press Delete.

Note: Note that when you delete the policy, the NetScaler searches the Goto Expression values of other policies in the bank. If any of these Goto Expression values match the priority level of the deleted policy, they are removed.

To configure policy bindings by using the Policy Manager

1. In the navigation pane, click the feature for which you want to configure policies. The choices are Responder, Integrated Caching, Rewrite or Compression.
2. In the details pane, click Policy Manager.
3. If you are configuring classic policy bindings for compression, in the Compression Policy Manager dialog box, click **Switch to Classic Syntax**. The dialog box switches to the classic syntax view and displays the Switch to Default Syntax button. At any time before you complete configuring policy bindings, if you want to configure bindings for policies that use the default syntax, click the Switch to Default Syntax button.
4. For features other than Responder, to specify the bind point, click Request or Response, and then click one of the request-time or response-time bind points. The options are Override Global, LB Virtual Server, CS Virtual Server, Default Global, or Policy Label. If you are configuring the Responder, the Request and Response flow types are not available.
5. To bind a policy to this bind point, click Insert Policy, and select a previously configured policy, a NOPOLICY label, or the New policy option. Depending on the option that you select, you have the following choices:
   - **New policy:** Create the policy as described in "Creating or Modifying a Policy," and then configure the priority level, GoTo expression, and policy invocation as described in the table, "Binding Policies That Use the Default Syntax."
   - **Existing policy**, **NOPOLICY**, or **NOPOLICY<feature name>**: Configure the priority level, GoTo expression, and policy invocation as described in the table, "Binding Policies That Use the Default Syntax." The **NOPOLICY** or **NOPOLICY<feature name>** options are available only for policies that use default syntax expressions.
6. Repeat the preceding steps to add entries to this policy bank.
7. To modify the priority level for an entry, you can do any of the following:
   - Double-click the Priority field for an entry and edit the value.
   - Click and drag a policy to another row in the table.
   - Click Regenerate Priorities.

   In all three cases, priority levels of all other policies are modified as needed to accommodate the new value. Goto Expressions with integer values are also updated automatically. For example, if you change a priority value of 10 to 100,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

all policies with a Goto Expression value of 10 are updated to the value 100.

8. To change the policy, action, or policy bank invocation for an row in the table, click the down arrow to the right of the entry and do one of the following:
   - To change the policy, select another policy name or select New Policy and follow the steps in "Creating or Modifying a Policy."
   - To change the Goto Expression, select Next, End, USE_INVOCATION_RESULT, or select more and enter an expression whose result returns the priority level of another entry in this policy bank.
   - To modify an invocation, select an existing policy bank, or click New Policy Label and follow the steps in "Binding a Policy to a Policy Label."
9. To unbind a policy or a policy label invocation from this bank, click any field in the row that contains the policy or policy label, and then click Unbind Policy.
10. When you are done, click Apply Changes. A message in the status bar indicates that the policy is bound successfully.

To remove unused policies by using the Policy Manager

1. In the navigation pane, click the feature for which you want to configure the policy bank. The choices are Responder, Integrated Caching, or Rewrite.
2. In the details pane, click <Feature Name> policy manager.
3. In the <Feature Name> Policy Manager dialog box, click Cleanup Configuration.
4. In the Cleanup Configuration dialog box, select the items that you want to delete, and then click Remove.
5. In the Remove dialog box, click Yes.
6. Click Close. A message in the status bar indicates that the policy is removed successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Default Syntax Expressions: Getting Started

May 25, 2015

Default syntax policies evaluate data on the basis of information that you supply in default syntax expressions. A default syntax expression analyzes data elements (for example, HTTP headers, source IP addresses, the NetScaler system time, and POST body data). In addition to configuring a default syntax expression in a policy, in some NetScaler features you configure default syntax expressions outside of the context of a policy.

To create a default syntax expression, you select a prefix that identifies a piece of data that you want to analyze, and then you specify an operation to perform on the data. For example, an operation can match a piece of data with a text string that you specify, or it can transform a text string into an HTTP header. Other operations match a returned string with a set of strings or a string pattern. You configure compound expressions by specifying Boolean and arithmetic operators, and by using parentheses to control the order of evaluation.

Default syntax expressions can also contain classic expressions. You can assign a name to a frequently used expression to avoid having to build the expression repeatedly.

Policies and a few other entities include rules that the NetScaler uses to evaluate a packet in the traffic flowing through it, to extract data from the NetScaler system itself, to send a request (a “callout”) to an external application, or to analyze another piece of data. A rule takes the form of a logical expression that is compared against traffic and ultimately returns values of TRUE or FALSE.

The elements of the rule can themselves return TRUE or FALSE, string, or numeric values.

Before configuring a default syntax expression, you need to understand the characteristics of the data that the policy or other entity is to evaluate. For example, when working with the Integrated Caching feature, a policy determines what data can be stored in the cache. With Integrated Caching, you need to know the URLs, headers, and other data in the HTTP requests and responses that the NetScaler receives. With this knowledge, you can configure policies that match the actual data and enable the NetScaler to manage caching for HTTP traffic. This information helps you determine the type of expression that you need to configure in the policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Basic Elements of a Default Syntax Expression

Sep 20, 2017

A default syntax expression consists of, at a minimum, a prefix (or a single element used in place of a prefix). Most expressions also specify an operation to be performed on the data that the prefix identifies. You format an expression of up to 1,499 characters as follows:

<prefix>.<operation> [<compound-operator> <prefix>.<operation>. . .]

where

<prefix>

is an anchor point for starting an expression.

The prefix is a period-delimited key that identifies a unit of data. For example, the following prefix examines HTTP requests for the presence of a header named Content-Type:

```
http.req.header("Content-Type")
```

Prefixes can also be used on their own to return the value of the object that the prefix identifies.

<operation>

identifies an evaluation that is to be performed on the data identified by the prefix.

For example, consider the following expression:

```
http.req.header("Content-Type").eq("text/html")
```

In this expression, the following is the operator component:

```
eq("text/html")
```

This operator causes the NetScaler to evaluate any HTTP requests that contain a Content-Type header, and in particular, to determine if the value of this header is equal to the string “text/html.” For more information, see "Operations."

<compound-operator>

is a Boolean or arithmetic operator that forms a compound expression from multiple prefix or prefix.operation elements.

For example, consider the following expression:

```
http.req.header("Content-Type").eq("text/html") && http.req.url.contains(".html")
```

This document includes the following details:

- Prefixes
- Single-Element Expressions
- Operations
- Basic Operations on Expression Prefixes

Prefixes

Updated: 2013-09-30

An expression prefix represents a discrete piece of data. For example, an expression prefix can represent an HTTP URL, an HTTP Cookie header, or a string in the body of an HTTP POST request. An expression prefix can identify and return a wide

© 1999-2017 Citrix Systems, Inc. All rights reserved.

variety of data types, including the following:

- A client IP address in a TCP/IP packet
- NetScaler system time
- An external callout over HTTP
- A TCP or UDP record type

In most cases, an expression prefix begins with one of the following keywords:

CLIENT:

Identifies a characteristic of the client that is either sending a request or receiving a response, as in the following examples:

- The prefix client.ip.dst designates the destination IP address in the request or response.
- The prefix client.ip.src designates the source IP address.

HTTP:

Identifies an element in an HTTP request or a response, as in the following examples:

- The prefix http.req.body(integer) designates the body of the HTTP request as a multiline text object, up to the character position designated in integer.
- The prefix http.req.header("header_name") designates an HTTP header, as specified in header_name.
- The prefix http.req.url designates an HTTP URL in URL-encoded format.

SERVER:

Identifies an element in the server that is either processing a request or sending a response.

SYS:

Identifies a characteristic of the NetScaler that is processing the traffic.

Note: Note that DNS policies support only SYS, CLIENT, and SERVER objects.

In addition, in the NetScaler Gateway, the Clientless VPN function can use the following types of prefixes:

TEXT:

Identifies any text element in a request or a response.

TARGET:

Identifies the target of a connection.

URL:

Identifies an element in the URL portion of an HTTP request or response.

As a general rule of thumb, any expression prefix can be a self-contained expression. For example, the following prefix is a complete expression that returns the contents of the HTTP header specified in the string argument (enclosed in quotation marks):

```
http.res.header.("myheader")
```

Or you can combine prefixes with simple operations to determine TRUE and FALSE values. For example, the following returns a value of TRUE or FALSE:

```
http.res.header.("myheader").exists
```

You can also use complex operations on individual prefixes and multiple prefixes within an expression, as in the following example:

```
http.req.url.length + http.req.cookie.length <= 500
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Which expression prefixes you can specify depends on the NetScaler feature. The following table describes the expression prefixes that are of interest on a per-feature basis

Table 1. Permitted Types of Expression Prefixes in Various NetScaler Features

Feature	Types of Expression Prefix Used in the Feature
DNS	SYS, CLIENT, SERVER
Responder in Protection Features	HTTP, SYS, CLIENT
Content Switching	HTTP, SYS, CLIENT
Rewrite	HTTP, SYS, CLIENT, SERVER, URL, TEXT, TARGET, VPN
Integrated Caching	HTTP, SYS, CLIENT, SERVER
NetScaler Gateway, Clientless Access	HTTP, SYS, CLIENT, SERVER, URL, TEXT, TARGET, VPN

Note: For details on the permitted expression prefixes in a feature, see the documentation for that feature.

Single-Element Expressions

The simplest type of default syntax expression contains a single element. This element can be one of the following:

- true. A default syntax expression can consist simply of the value true. This type of expression always returns a value of TRUE. It is useful for chaining policy actions and triggering Goto expressions.
- false. A default syntax expression can consist simply of the value false. This type of expression always returns a value of FALSE.
- A prefix for a compound expression. For example, the prefix HTTP.REQ.HOSTNAME is a complete expression that returns a host name and HTTP.REQ.URL is a complete expression that returns a URL. The prefix could also be used in conjunction with operations and additional prefixes to form a compound expression.

Operations

In most expressions, you also specify an operation on the data that the prefix identifies. For example, suppose that you specify the following prefix:

http.req.url

This prefix extracts URLs in HTTP requests. This expression prefix does not require any operators to be used in an expression. However, when you configure an expression that processes HTTP request URLs, you can specify operations that analyze particular characteristics of the URL. Following are a few possibilities:

- Search for a particular host name in the URL.
- Search for a particular path in the URL.
- Evaluate the length of the URL.
- Search for a string in the URL that indicates a time stamp and convert it to GMT.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following is an example of a prefix that identifies an HTTP header named Server and an operation that searches for the string IIS in the header value:

http.res.header("Server").contains("IIS")

Following is an example of a prefix that identifies host names and an operation that searches for the string “www.mycompany.com” as the value of the name:

http.req.hostname.eq("www.mycompany.com")

Basic Operations on Expression Prefixes

The following table describes a few of the basic operations that can be performed on expression prefixes.

Table 2. Basic Operations for Expressions

Operation	Determines Whether or Not
CONTAINS(<string>)	The object matches <string>. Following is an example: http.req.header("Cache-Control").contains("no-cache")
EXISTS	A particular item is present in an object. Following is an example: http.res.header("MyHdr").exists
EQ(<text>)	A particular non-numeric value is present in an object. Following is an example: http.req.method.eq(post)
EQ(<integer>)	A particular numeric value is present in an object. Following is an example: client.ip.dst.eq(10.100.10.100)
LT(<integer>)	An object's value is less than a particular value. Following is an example: http.req.content_length.lt(5000)
GT(<integer>)	An object's value is greater than a particular value. Following is an example: http.req.content_length.gt(5)

The following table summarizes a few of the available types of operations.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 3. Basic Types of Operations

Operation Type	Description
Text operations	Match individual strings and sets of strings with any portion of a target. The target can be an entire string, the start of a string, or any portion of text in between the start and the end of the string. For example, you can extract the string "XYZ" from "XYZSomeText". Or, you can compare an HTTP header value with an array of different strings. You can also transform text into another type of data. Following are examples: ● Transform a string into an integer value ● Create a list from the query strings in a URL ● Transform a string into a time value
Numeric operations	Numeric operations include applying arithmetic operators, evaluating content length, the number of items in a list, dates, times, and IP addresses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Compound Default Syntax Expressions

Sep 20, 2017

You can configure a default syntax expression that contains Boolean or arithmetic operators and multiple atomic operations. The following compound expression contains a boolean AND:

```
http.req.hostname.eq("mycompany.com") && http.req.method.eq(post)
```

The following expression adds the value of two targets, and compares the result to a third value:

```
http.req.url.length + http.req.cookie.length <= 500
```

A compound expression can contain any number of logical and arithmetic operators. The following expression evaluates the length of an HTTP request on the basis of its URL and cookie, evaluates text in the header, and performs a Boolean AND on these two results:

```
http.req.url.length + http.req.cookie.length <= 500 && http.req.header.contains("some text")
```

You can use parentheses to control the order of evaluation in a compound expression.

This document includes the following details:

- Booleans in Compound Expressions
- Parentheses in Compound Expressions
- Compound Operations for Strings
- Compound Operations for Numbers

Booleans in Compound Expressions

You configure compound expressions with the following operators:

&&.

This operator is a logical AND. For the expression to evaluate to TRUE, all components that are joined by the And must evaluate to TRUE. Following is an example:

```
http.req.url.hostname.eq("myHost") && http.req.header("myHeader").exists
```

||.

This operator is a logical OR. If any component of the expression that is joined by the OR evaluates to TRUE, the entire expression is TRUE.

!.

Performs a logical NOT on the expression.

In some cases, the NetScaler configuration utility offers AND, NOT, and OR operators in the Add Expression dialog box. However, these are of limited use. Citrix recommends that you use the operators &&, ||, and ! to configure compound expressions that use Boolean logic.

Parentheses in Compound Expressions

You can use parentheses to control the order of evaluation of an expression. The following is an example:

```
http.req.url.contains("myCompany.com") || (http.req.url.hostname.eq("myHost") && http.req.header("myHeader").exists)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following is another example:

(http.req.header("Content-Type").exists && http.req.header("Content-Type").eq("text/html")) || (http.req.header("Transfer-Encoding").exists || http.req.header("Content-Length").exists)

Compound Operations for Strings

The following table describes operators that you can use to configure compound operations on string data.

Table 1. String-Based Operations for Compound Default Syntax Expressions

All string operations	
Operations that produce a string value	
str + str	Concatenates the value of the expression on the left of the operator with the value on the right. Following is an example: http.req.hostname + http.req.url.protocol
str + num	Concatenates the value of the expression on the left of the operator with a numeric value on the right. Following is an example: http.req.hostname + http.req.url.content_length
num + str	Concatenates the numeric value of the expression on the left side of the operator with a string value on the right. Following is an example: http.req.url.content_length + http.req.url.hostname
str + ip	Concatenates the string value of the expression on the left side of the operator with an IP address value on the right. Following is an example: http.req.hostname + 10.00.000.00
ip + str	Concatenates the IP address value of the expression on the left of the operator with a string value on the right.Following is an example: client.ip.dst + http.req.url.hostname
str1 ALT str2	Uses the string1 or string2 value that is derived from the expression on either side of the operator, as long as neither of these expressions is a compound expressions. Following is an example: http.req.hostname alt client.ip.src
Operations on strings that produce a result of TRUE or FALSE	

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

All string operations	
str == str	Evaluates whether the strings on either side of the operator are the same. Following is an example: http.req.header("myheader") == http.res.header("myheader")
str <= str	Evaluates whether the string on the left side of the operator is the same as the string on the right, or precedes it alphabetically.
str >= str	Evaluates whether the string on the left side of the operator is the same as the string on the right, or follows it alphabetically.
str < str	Evaluates whether the string on the left side of the operator precedes the string on the right alphabetically.
str > str	Evaluates whether the string on the left side of the operator follows the string on the right alphabetically.
str !!= str	Evaluates whether the strings on either side of the operator are different.
Logical operations on strings	
bool && bool	This operator is a logical AND. When evaluating the components of the compound expression, all components that are joined by the AND must evaluate to TRUE. Following is an example: http.req.method.eq(GET) && http.req.url.query.contains("viewReport && my_pagelabel")
bool || bool	This operator is a logical OR. When evaluating the components of the compound expression, if any component of the expression that is joined by the OR evaluates to TRUE, the entire expression is TRUE. Following is an example: http.req.url.contains(".js") || http.res.header.("Content-Type").contains("javascript")
!bool	Performs a logical NOT on the expression.

Compound Operations for Numbers

Updated: 2013-09-02

You can configure compound numeric expressions. For example, the following expression returns a numeric value that is the sum of an HTTP header length and a URL length:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http.req.header.length + http.req.url.length

The following tables describes operators that you can use to configure compound expressions for numeric data.

Table 2. Arithmetic Operations on Numbers

Operator	Description
num + num	Add the value of the expression on the left of the operator to the value of the expression on the right. Following is an example: http.req.content_length + http.req.url.length
num – num	Subtract the value of the expression on the right of the operator from the value of the expression on the left.
num * num	Multiply the value of the expression on the left of the operator with the value of the expression on the right. Following is an example: client.interface.rxthroughput * 9
num / num	Divide the value of the expression on the left of the operator by the value of the expression on the right.
num % num	Calculate the modulo, or the numeric remainder on a division of the value of the expression on the left of the operator by the value of the expression on the right. For example, the values "15 mod 4" equals 3, and "12 mod 4" equals 0.
~number	Returns a number after applying a bitwise logical negation of the number. The following example assumes that numeric.expression returns 12 (binary 1100): ~numeric.expression. The result of applying the ~ operator is -11 (a binary 1110011, 32 bits total with all ones to the left). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.
number ^ number	Compares two bit patterns of equal length and performs an XOR operation on each pair of corresponding bits in each number argument, returning 1 if the bits are different, and 0 if they are the same. Returns a number after applying a bitwise XOR to the integer argument and the current number value. If the values in the bitwise comparison are the same, the returned value is a 0. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010): numeric.expression1 ^ numeric.expression2

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operator	Description
	The result of applying the ^ operator to the entire expression is 6 (binary 0110). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.
number | number	Returns a number after applying a bitwise OR to the number values. If either value in the bitwise comparison is a 1, the returned value is a 1. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010): numeric.expression1 | numeric.expression2 The result of applying the | operator to the entire expression is 14 (binary 1110). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.
number & number	Compares two bit patterns of equal length and performs a bitwise AND operation on each pair of corresponding bits, returning 1 if both of the bits contains a value of 1, and 0 if either bits are 0. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010): numeric.expression1 & numeric.expression2 The whole expression evaluates to 8 (binary 1000). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.
num << num	Returns a number after a bitwise left shift of the number value by the right-side number argument number of bits. Note that the number of bits shifted is integer modulo 32. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 3: numeric.expression1 << numeric.expression2 The result of applying the LSHIFT operator is 96 (a binary 1100000). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.
num >> num	Returns a number after a bitwise right shift of the number value by the integer argument number of bits. Note that the number of bits shifted is integer modulo 32. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 3: numeric.expression1 >> numeric.expression2 The result of applying the RSHIFT operator is 1 (a binary 0001).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operator	Description
	Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Table 3. Numeric Operators That Produce a Result of TRUE or FALSE

Operator	Description
num == num	Determine if the value of the expression on the left of the operator is equal to the value of the expression on the right.
num != num	Determine if the value of the expression on the left of the operator is not equal to the value of the expression on the right.
num > num	Determine if the value of the expression on the left of the operator is greater than the value of the expression on the right.
num < num	Determine if the value of the expression on the left of the operator is less than the value of the expression on the right.
num >= num	Determine if the value of the expression on the left of the operator is greater than or equal to the value of the expression on the right.
num <= num	Determine if the value of the expression on the left of the operator is less than or equal to the value of the expression on the right

Functions for Data Types in the Policy Infrastructure

The NetScaler policy infrastructure supports the following numeric data types:

- Integer (32 bits)
- Unsigned long (64 bits)
- Double (64 bits)

Simple expressions can return all of these data types. Therefore, you can create compound expressions that use arithmetic operators and logical operators to evaluate or return values of these data types. Additionally, you can use all of these values in policy expressions. Literal constants of type unsigned long can be specified by appending the string ul to the number. Literal constants of type double contain a period (.), an exponent, or both.

Arithmetic Operators, Logical Operators, and Type Promotion

In compound expressions, the following standard arithmetic and logical operators can be used for the double and unsigned long data types:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- +, -, *, and /
- %, ~, ^, &, |, <<, and >> (do not apply to double)
- ==, !=, >, <, >=, and <=

All of these operators have the same meaning as in the C programming language.

In all cases of mixed operations between operands of type integer, unsigned long, and double, type promotion is performed so that the operation can be performed on operands of the same type. A type of lower precedence is automatically promoted to the type of the operand with the highest precedence involved in the operation. The order of precedence (higher to lower) is as follows:

- Double
- Unsigned long
- Integer

Therefore, an operation that returns a numeric result returns a result of the highest type involved in the operation.

For example, if the operands are of type integer and unsigned long, the integer operand is automatically converted to type unsigned long. This type conversion is performed even in simple expressions in which the type of data identified by the expression prefix does not match the type of data that is passed as the argument to the function. To illustrate such an example, in the operation HTTP.REQ.CONTENT_LENGTH.DIV(3ul), the integer returned by the prefix HTTP.REQ.CONTENT_LENGTH is automatically converted to unsigned long (the type of the data passed as the argument to the DIV() function), and an unsigned long division is performed. Similarly, the argument can be promoted in an expression. For example, HTTP.REQ.HEADER("myHeader").TYPECAST_DOUBLE_AT.DIV(5) promotes the integer 5 to type double and performs double-precision division.

The following table describes the arithmetic and Boolean functions that can be used with the integer, unsigned long, and double data types. For information about expressions for casting data of one type to data of another type, see "Typecasting Data."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Specifying the Character Set in Expressions

Jul 11, 2013

The policy infrastructure on the Citrix® NetScaler® appliance supports the ASCII and UTF-8 character sets. The default character set is ASCII. If the traffic for which you are configuring an expression consists of only ASCII characters, you need not specify the character set in the expression. However, you must specify the character set in every simple expression that is meant for UTF-8 traffic. To specify the UTF-8 character set in a simple expression, you must include the SET_CHAR_SET(<charset>) function, with <charset> specified as UTF_8, as shown in the following examples:

```
HTTP.REQ.BODY(10).SET_CHAR_SET(UTF_8).CONTAINS("ß")
```

```
HTTP.RES.BODY(100).SET_CHAR_SET(UTF_8).BEFORE_STR("Bücher").AFTER_STR("Wörterbuch")
```

In an expression, the SET_CHAR_SET() function must be introduced at the point in the expression after which data processing must be carried out in the specified character set. For example, in the expression HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX(re/In the preceding example/).CONTAINS_ANY("Greek_ alphabet"), if the strings stored in the pattern set "Greek_alphabet" are in UTF-8, you must include the SET_CHAR_SET(UTF_8) function immediately before the CONTAINS_ANY("<string>") function, as follows:

```
HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX(re/In the preceding
example/).SET_CHAR_SET(UTF_8).CONTAINS_ANY("Greek_ alphabet")
```

The SET_CHAR_SET() function sets the character set for all further processing (that is, for all subsequent functions) in the expression unless it is overridden later in the expression by another SET_CHAR_SET() function that changes the character set. Therefore, if all the functions in a given simple expression are intended for UTF-8, you can include the SET_CHAR_SET(UTF_8) function immediately after functions that identify text (for example, the HEADER("<name>") or BODY(<int>) functions). In the second example that follows the first paragraph above, if the ASCII arguments passed to the AFTER_REGEX() and BEFORE_REGEX() functions are changed to UTF-8 strings, you can include the SET_CHAR_SET(UTF_8) function immediately after the BODY(1000) function, as follows:

```
HTTP.REQ.BODY(1000).SET_CHAR_SET(UTF_8).AFTER_REGEX(re/Bücher/).BEFORE_REGEX(re/Wörterbuch/).CONTAINS_ANY("Greek_alphabet")
```

The UTF-8 character set is a superset of the ASCII character set, so expressions configured for the ASCII character set continue to work as expected if you change the character set to UTF-8.

Compound Expressions with Different Character Sets

In a compound expression, if one subset of expressions is configured to work with data in the ASCII character set and the rest of the expressions are configured to work with data in the UTF-8 character set, the character set specified for each individual expression is considered when the expressions are evaluated individually. However, when processing the compound expression, just before processing the operators, the appliance promotes the character set of the returned ASCII values to UTF-8. For example, in the following compound expression, the first simple expression evaluates data in the ASCII character set while the second simple expression evaluates data in the UTF-8 character set:

```
HTTP.REQ.HEADER("MyHeader") == HTTP.REQ.BODY(10).SET_CHAR_SET(UTF_8)
```

However, when processing the compound expression, just before evaluating the "is equal to" Boolean operator, the NetScaler appliance promotes the character set of the value returned by HTTP.REQ.HEADER("MyHeader") to UTF-8.

The first simple expression in the following example evaluates data in the ASCII character set. However, when the NetScaler appliance processes the compound expression, just before concatenating the results of the two simple expressions, the appliance promotes the character set of the value returned by HTTP.REQ.BODY(10) to UTF-8.

```
HTTP.REQ.BODY(10) + HTTP.REQ.HEADER("MyHeader").SET_CHAR_SET(UTF_8)
```

Consequently, the compound expression returns data in the UTF-8 character set.

Specifying the Character Set Based on the Character Set of Traffic

You can set the character set to UTF-8 on the basis of traffic characteristics. If you are not sure whether the character set of the traffic being evaluated is UTF-8, you can configure a compound expression in which the first expression checks for UTF-8 traffic and subsequent expressions set the character set to UTF-8. Following is an example of a compound expression that first checks the value of "charset" in the request's Content-Type header for "UTF-8" before checking whether the first 1000 bytes in the request contain the UTF-8 string Bücher:

```
HTTP.REQ.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).TYPECAST_NVLIST_T('=', ';', '"').VALUE("charset").EQ("UTF-8") &&
HTTP.REQ.BODY(1000).SET_CHAR_SET(UTF_8).CONTAINS("Bücher")
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

If you are sure that the character set of the traffic being evaluated is UTF-8, the second expression in the example is sufficient.

Character and String Literals in Expressions

During expression evaluation, even if the current character set is ASCII, character literals and string literals, which are enclosed in single quotation marks (') and quotation marks ("), respectively, are considered to be literals in the UTF-8 character set. In a given expression, if a function is operating on character or string literals in the ASCII character set and you include a non-ASCII character in the literal, an error is returned.

Values in Hexadecimal and Octal Formats

When configuring an expression, you can enter values in octal and hexadecimal formats. However, each hexadecimal or octal byte is considered a UTF-8 byte. Invalid UTF-8 bytes result in errors regardless of whether the value is entered manually or pasted from the clipboard. For example, "\xce\x20" is an invalid UTF-8 character because "c8" cannot be followed by "20" (each byte in a multi-byte UTF-8 string must have the high bit set). Another example of an invalid UTF-8 character is "\xce \xa9," since the hexadecimal characters are separated by a white-space character.

Functions That Return UTF-8 Strings

Only the <text>.XPATH and <text>.XPATH_JSON functions always return UTF-8 strings. The following MYSQL routines determine at runtime which character set to return, depending on the data in the protocol:

- MYSQL_CLIENT_T.USER
- MYSQL_CLIENT_T.DATABASE
- MYSQL_REQ_QUERY_T.COMMAND
- MYSQL_REQ_QUERY_T.TEXT
- MYSQL_REQ_QUERY_T.TEXT(<unsigned int>)
- MYSQL_RES_ERROR_T.SQLSTATE
- MYSQL_RES_ERROR_T.MESSAGE
- MYSQL_RES_FIELD_T.CATALOG
- MYSQL_RES_FIELD_T.DB
- MYSQL_RES_FIELD_T.TABLE
- MYSQL_RES_FIELD_T.ORIGINAL_TABLE
- MYSQL_RES_FIELD_T.NAME
- MYSQL_RES_FIELD_T.ORIGINAL_NAME
- MYSQL_RES_OK_T.MESSAGE
- MYSQL_RES_ROW_T.TEXT_ELEM(<unsigned int>)

Terminal Connection Settings for UTF-8

When you set up a connection to the NetScaler appliance by using a terminal connection (by using PuTTY, for example), you must set the character set for transmission of data to UTF-8.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Classic Expressions in Default Syntax Expressions

Mar 20, 2012

Classic expressions describe basic characteristics of traffic. In some cases, you may want to use a classic expression in a default syntax expression. You can do so with the default syntax expression configuration tool. This can be helpful when manually migrating the older classic expressions to the default syntax.

Note that when you upgrade the NetScaler to version 9.0 or higher, Integrated Caching policies are automatically upgraded to default syntax policies, and the expressions in these policies are upgraded to the default syntax.

The following is the syntax for all default syntax expressions that use a classic expression:

SYS.EVAL_CLASSIC_EXPR("expression")

Following are examples of the SYS.EVAL_CLASSIC_EXPR("expression") expression:

```
sys.eval_classic_expr("req.ssl.client.cipher.bits > 1000")
sys.eval_classic_expr("url contains abc")
sys.eval_classic_expr("req.ip.sourceip == 10.102.1.61 -netmask 255.255.255.255")
sys.eval_classic_expr("time >= *:30:00GMT")
sys.eval_classic_expr("e1 || e2")
sys.eval_classic_expr("req.http.urllen > 50")
sys.eval_classic_expr("dayofweek == wedGMT")
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Default Syntax Expressions in a Policy

Sep 02, 2013

You can configure a default syntax expression of up to 1,499 characters in a policy. The user interface for default syntax expressions depends to some extent on the feature for which you are configuring the expression, and on whether you are configuring an expression for a policy or for another use.

When configuring expressions on the command line, you delimit the expression by using quotation marks (“. . .” or '. . .'). Within an expression, you escape additional quotation marks by using a back-slash (\). For example, the following are standard methods for escaping quotation marks in an expression:

"\"abc\""

‘\"abc\"’

You must also use a backslash to escape question marks and other backslashes on the command line. For example, the expression http.req.url.contains(“\?”) requires a backslash so that the question mark is parsed. Note that the backslash character will not appear on the command line after you type the question mark. On the other hand, if you escape a backslash (for example, in the expression 'http.req.url.contains("\\\\http")'), the escape characters are echoed on the command line.

To make an entry more readable, you can escape the quotation marks for an entire expression. At the start of the expression you enter the escape sequence “q” plus one of the following special characters: /{<|~$^+=&%@`?.

You enter only the special character at the end of the expression, as follows:

q@http.req.url.contains("sometext") && http.req.cookie.exists@

q~http.req.url.contains("sometext") && http.req.cookie.exists~

Note that an expression that uses the { delimiter is closed with }.

For some features (for example, Integrated Caching and Responder), the policy configuration dialog box provides a secondary dialog box for configuring expressions. This dialog enables you to choose from drop-down lists that show the available choices at each point during expression configuration. You cannot use arithmetic operators when using these configuration dialogs, but most other default syntax expression features are available. To use arithmetic operators , write your expressions in free-form format.

To configure a default syntax rule by using the command line interface

At the command prompt, type the following commands to configure a default syntax rule and verify the configuration:

1. add cache|dns|rewrite|cs policy policyName **-rule** expression featureSpecificParameters **-action**
2. show cache|dns|rewrite|cs policy policyName

   Following is an example of configuring a caching policy:

   Example

   > add cache policy pol-cache -rule http.req.content_length.le(5) -action INVAL
   Done

   > show cache policy pol-cache

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Name: pol-cache
    Rule: http.req.content_length.le(5)
    CacheAction: INVAL
    Invalidate groups: DEFAULT
    UndefAction: Use Global
    Hits: 0
    Undef Hits: 0

Done
```

To configure a default syntax policy expression by using the configuration utility

1. In the navigation pane, click the name of the feature where you want to configure a policy, for example, you can select Integrated Caching, Responder, DNS, Rewrite, or Content Switching, and then click Policies.
2. Click Add.
3. For most features, click in the Expression field. For Content Switching, click Configure.
4. Click the Prefix icon (the house) and select the first expression prefix from the drop-down list. For example, in Responder, the options are HTTP, SYS, and CLIENT. The next set of applicable options appear in a drop-down list.
5. Double-click the next option to select it, and then type a period (.). Again, a set of applicable options appears in another drop-down list.
6. Continue selecting options until an entry field (signalled by parentheses) appears. When you see an entry field, enter an appropriate value in the parentheses. For example, if you select GT(int) (greater-than, integer format), you specify an integer in the parentheses. Text strings are delimited by quotation marks. Following is an example:
   HTTP.REQ.BODY(1000).BETWEEN("this","that")
7. To insert an operator between two parts of a compound expression, click the Operators icon (the sigma), and select the operator type. Following is an example of a configured expression with a Boolean OR (signalled by double vertical bars, ||):
   HTTP.REQ.URL.EQ("www.mycompany.com")||HTTP.REQ.BODY(1000).BETWEEN("this","that")
8. To insert a named expression, click the down arrow next to the Add icon (the plus sign) and select a named expression.
9. To configure an expression using drop-down menus, and to insert built-in expressions, click the Add icon (the plus sign). The Add Expression dialog box works in a similar way to the main dialog box, but it provides drop-down lists for selecting options, and it provides text fields for data entry instead of parentheses. This dialog box also provides a Frequently Used Expressions drop-down list that inserts commonly used expressions. When you are done adding the expression, click OK.
10. When finished, click Create. A message in the status bar indicates that the policy expression is configured successfully.

To test a default syntax expression by using the configuration utility

1. In the navigation pane, click the name of the feature for which you want to configure a policy (for example, you can select Integrated Caching, Responder, DNS, Rewrite, or Content Switching), and then click Policies.
2. Select a policy and click Open.
3. To test the expression, click the Evaluate icon (the check mark).
4. In the expression evaluator dialog box, select the Flow Type that matches the expression.
5. In the HTTP Request Data or HTTP Response Data field, paste the HTTP request or response that you want to parse with the expression, and click Evaluate. Note that you must supply a complete HTTP request or response, and the header and body should be separated by blank line. Some programs that trap HTTP headers do not also trap the response. If you are copying and pasting only the header, insert a blank line at the end of the header to form a complete HTTP request or response.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

6. Click Close to close this dialog box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Named Default Syntax Expressions

Oct 29, 2013

Instead of retyping the same expression multiple times in multiple policies, you can configure a named expression and refer to the name any time you want to use the expression in a policy. For example, you could create the following named expressions:

ThisExpression:

http.req.body(100).contains("this")

ThatExpression:

http.req.body(100).contains("that")

You can then use these named expressions in a policy expression. For example, the following is a legal expression based on the preceding examples:

ThisExpression || ThatExpression

You can use the name of a default syntax expression as the prefix to a function. The named expression can be either a simple expression or a compound expression. The function must be one that can operate on the type of data that is returned by the named expression.

Example 1: Simple Named Expression as a Prefix

The following simple named expression, which identifies a text string, can be used as a prefix to the AFTER_STR("<string>")function, which works with text data:

HTTP.REQ.BODY(1000)

If the name of the expression is top1KB, you can use top1KB.AFTER_STR("username") instead of HTTP.REQ.BODY(1000).AFTER_STR("username").

Example 2: Compound Named Expression as a Prefix

You can create a compound named expression called basic_header_value to concatenate the user name in a request, a colon (:), and the user's password, as follows:

add policy expression basic_header_value "HTTP.REQ.USER.NAME + \":\" + HTTP.REQ.USER.PASSWD"

You can then use the name of the expression in a rewrite action, as shown in the following example:

add rewrite action insert_b64encoded_authorization insert_http_header authorization '"Basic " + basic_header_value.b64encode' -bypassSafetyCheck YES

In the example, in the expression that is used to construct the value of the custom header, the B64 encoding algorithm is applied to the string returned by the compound named expression.

You can also use a named expression (either by itself or as a prefix to a function) to create the text expression for the replacement target in a rewrite.

To configure a named default syntax expression by using the command line interface

At the command prompt, type the following commands to configure a named expression and verify the configuration:

- add policy expression <name><value>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- show policy expression <name>

  Example

  ```
  > add policy expression myExp "http.req.body(100).contains(\"the other\")"
   Done

  > show policy expression myExp
  1)     Name: myExp  Expr: "http.req.body(100).contains("the other")"  Hits: 0 Type : ADVANCED
   Done
  ```

  The expression can be up to 1,499 characters.

To configure a named expression by using the configuration utility

1. In the navigation pane, expand AppExpert, and then click Expressions.
2. Click Advanced Expressions.
3. Click Add.
4. Enter a name and a description for the expression.
5. Configure the expression by using the process described in "To configure a default syntax policy expression by using the configuration utility." A message in the status bar indicates that the policy expression is configured successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Default Syntax Expressions Outside the Context of a Policy

Sep 20, 2017

A number of functions, including the following, can require a default syntax expression that is not part of a policy:

Integrated Caching selectors:

You define multiple non-compound expressions (selectlets) in the definition of the selector. Each selectlet is in an implicit logical AND relationship with the others.

Load Balancing:

You configure an expression for the TOKEN method of load balancing for a load balancing virtual server.

Rewrite actions:

Expressions define the location of the rewrite action and the type of rewriting to be performed, depending on the type of rewrite action that you are configuring. For example, a DELETE action only uses a target expression. A REPLACE action uses a target expression and an expression to configure the replacement text.

Rate-based policies:

You use default syntax expressions to configure Limit Selectors. You can use these selectors when configuring policies to throttle the rate of traffic to various servers. You define up to five non-compound expressions (selectlets) in the definition of the selector. Each selectlet is in an implicit logical AND with the others.

To configure a default syntax expression outside a policy by using the command line interface (cache selector example)

At the command prompt, type the following commands to configure a default syntax expression outside a policy and verify the configuration:

- add cache selector <selectorName> <rule>
- show cache selector <selectorName>

  Example

```
> add cache selector mainpageSelector "http.req.cookie.value(\"ABC_def\")"
"http.req.url.query.value(\"_ghi\")"selector "mainpageSelector" added
 Done
> show cache selector mainpageSelector
      Name: mainpageSelector
      Expressions:
            1) http.req.cookie.value("ABC_def")
            2) http.req.url.query.value("_ghi")
 Done
```

Following is an equivalent command that uses the more readable q delimiter, as described in "Configuring Default Syntax Expressions in a Policy":

```
> add cache selector mainpageSelector2 q~http.req.cookie.value("ABC_def")~
q~http.req.url.query.value("_ghi")~selector "mainpageSelector2" added
 Done
> show cache selector mainpageSelector2
      Name: mainpageSelector2
      Expressions:
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        1) http.req.cookie.value("ABC_def")
        2) http.req.url.query.value("_ghi")
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Default Syntax Expressions: Evaluating Text

Sep 02, 2013

You can configure a policy with a default syntax expression that evaluates text in a request or response. Default syntax text expressions can range from simple expressions that perform string matching in HTTP headers to complex expressions that encode and decode text. You can configure text expressions to be case sensitive or case insensitive and to use or ignore spaces. You can also configure complex text expressions by combining text expressions with Boolean operators

You can use expression prefixes and operators for evaluating HTTP requests, HTTP responses, and VPN and Clientless VPN data. However, text expression prefixes are not restricted to evaluating these elements of your traffic. For information about additional default syntax text expression prefixes and operators, see the following topics:

- Pattern Sets
- Regular Expressions
- Typecasting Data
- Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data
- Default Syntax Expressions: Parsing SSL Certificates
- Expressions for SSL Certificate Dates

© 1999-2017 Citrix Systems, Inc. All rights reserved.

About Text Expressions

Sep 02, 2013

You can configure various expressions for working with text that flows through the NetScaler appliance. Following are some examples of how you can parse text by using a default syntax expression:

- Determine that a particular HTTP header exists.
  For example, you may want to identify HTTP requests that contains a particular Accept-Language header for the purpose of directing the request to a particular server.
- Determine that a particular HTTP URL contains a particular string.
  For example, you may want to block requests for particular URLs. Note that the string can occur at the beginning, middle, or end of another string.
- Identify a POST request that is directed to a particular application.
  For example, you may want to identify all POST requests that are directed to a database application for the purpose of refreshing cached application data.

Note that there are specialized tools for viewing the data stream for HTTP requests and responses. For example, from the following URL, you can download a Firefox Web browser plug-in that displays HTTP request and response headers:

"https://addons.mozilla.org/en-US/firefox/addon/3829"

The following plug-in displays headers, query strings, POST data, and other information:

"https://addons.mozilla.org/en-US/firefox/addon/6647"

After you download these plug-ins, they are accessible from the Firefox Tools menu.

About Operations on Text

A text-based expression consists of at least one prefix to identify an element of data and usually (although not always) an operation on that prefix. Text-based operations can apply to any part of a request or a response. Basic operations on text include various types of string matches.

For example, the following expression compares a header value with a string:

```
http.req.header("myHeader").contains("some-text")
```

Following expressions are examples of matching a file type in a request:

```
http.req.url.suffix.contains("jpeg")
```

```
http.req.url.suffix.eq("jpeg")
```

In the preceding examples, the contains operator permits a partial match and the eq operator looks for an exact match.

Other operations are available to format the string before evaluating it. For example, you can use text operations to strip out quotes and white spaces, to convert the string to all lowercase, or to concatenate strings.

Note: Complex operations are available to perform matching based on patterns or to convert one type of text format to another type.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For more information, see the following topics:

- "Pattern Sets and Data Sets."
- "Regular Expressions."
- "Typecasting Data."

Compounding and Precedence in Text Expressions

You can apply various operators to combine text prefixes or expressions. For example, the following expression concatenates the returned values of each prefix:

```
http.req.hostname + http.req.url
```

Following is an example of a compound text expression that uses a logical AND. Both components of this expression must be TRUE for a request to match the expression:

```
http.req.method.eq(post) && http.req.body(1024).startswith("destination=")
```

Note: For more information on operators for compounding, see "Compound Default Syntax Expressions."

Categories of Text Expressions

The primary categories of text expressions that you can configure are:

- Information in HTTP headers, HTTP URLs, and the POST body in HTTP requests.
  For more information, see "Expression Prefixes for Text in HTTP Requests and Responses."

- Information regarding a VPN or a clientless VPN.
  For more information, see "Expression Prefixes for VPNs and Clientless VPNs."

- TCP payload information.
  For more information about TCP payload expressions, see "Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data."

- Text in a Secure Sockets Layer (SSL) certificate.
  For information about text expressions for SSL and SSL certificate data, see "Default Syntax Expressions: Parsing SSL Certificates" and "Expressions for SSL Certificate Dates."

Note: Parsing a document body, such as the body of a POST request, can affect performance. You may want to test the performance impact of policies that evaluate a document body.

Guidelines for Text Expressions

From a performance standpoint, it typically is best to use protocol-aware functions in an expression. For example, the following expression makes use of a protocol-aware function:

```
HTTP.REQ.URL.QUERY
```

The previous expression performs better than the following equivalent expression, which is based on string parsing:

```
HTTP.REQ.URL.AFTER_STR("?")
```

In the first case, the expression looks specifically at the URL query. In the second case, the expression scans the data for the first occurrence of a question mark.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

There is also a performance benefit from structured parsing of text, as in the following expression:

```
HTTP.REQ.HEADER("Example").TYPECAST_LIST_T(',').GET(1)
```

(For more information on typecasting, see "Typecasting Data.") The typecasting expression, which collects comma-delimited data and structures it into a list, typically would perform better than the following unstructured equivalent:

```
HTTP.REQ.HEADER("Example").AFTER_STR(",").BEFORE_STR(",")
```

Finally, unstructured text expressions typically have better performance than regular expressions. For example, the following is an unstructured text expression:

```
HTTP.REQ.HEADER("Example").AFTER_STR("more")
```

The previous expression would generally provide better performance than the following equivalent, which uses a regular expression:

```
HTTP.REQ.HEADER("Example").AFTER_REGEX(re/more/)
```

For more information on regular expressions, see "Regular Expressions."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefixes for Text in HTTP Requests and Responses

Oct 11, 2017

An HTTP request or response typically contains text, such as in the form of headers, header values, URLs, and POST body text. You can configure expressions to operate on one or more of these text-based items in an HTTP request or response.

The following table describes the expression prefixes that you can configure to extract text from different parts of an HTTP request or response.

Table 1. HTTP Expression Prefixes That Return Text

Prefix	Description
HTTP.REQ.BODY(<integer>)	Example: HTTP.REQ.BODY(100) It will return first 100 characters of HTTP Request body. If the length of the body is less than 100 then the whole body will result as output.
HTTP.REQ.HOSTNAME	Example: HTTP.REQ.HOSTNAME.EQ("abc.com") The above example returns true if the hostname is abc.com. It returns HTTP HostName object from this request. If the target hostname is present in the first line of the request then that is selected. Otherwise, the value in the last occurrence of the HOST header is selected. The format of the output is abc.foo.com:8080. For more information on typecasting, see "Typecasting Data."
HTTP.REQ.HOSTNAME.DOMAIN	Example: HTTP.REQ.HOSTNAME.DOMAIN.EQ("foobar.com") The above example returns true if domain name is foobar.com. It returns Domain name part of the hostname. If the hostname is www.foobar.com or www.foobar.com:8080, then domain is foobar.com.
HTTP.REQ.HOSTNAME.SERVER	Example: HTTP.REQ.HOSTNAME.SERVER.EQ("www.foobar.com") The above example returns true if the server name is www.foobar.com. If the hostname is www.foobar.com or www.foobar.com:8080, then the server is www.foobar.com.
HTTP.REQ.METHOD	Example: HTTP.REQ.METHOD.EQ("GET") The above example returns true if the method name is GET.
HTTP.REQ.URL	Example: HTTP.REQ.URL.EQ("http://www.google.com") The above example returns true if URL is http://www.google.com. It returns the HTTP URL object from request.
HTTP.REQ.URL.HOSTNAME	Example: HTTP.REQ.URL.HOSTNAME.EQ("abc.foo.com:8080") The above example returns true if hostname in URL is abc.foo.com:8080. It returns HTTP Host Name present in the URL. For more information on typecasting, see "Typecasting Data."
HTTP.REQ.URL.HOSTNAME.DOMAIN	Example: HTTP.REQ.URL.HOSTNAME.DOMAIN.EQ("foobar.com") The above example returns true if the domain name is foobar.com. It returns Domain

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	name part of the hostname. If the hostname is www.foobar.com or www.foobar.com:8080, then the domain is foobar.com.
HTTP.REQ.URL.HOSTNAME.SERVER	Example: HTTP.REQ.URL.HOSTNAME.SERVER.EQ("www.foobar.com") The above example returns true if the server name is www.foobar.com. If the hostname is www.foobar.com or www.foobar.com:8080, then the server is www.foobar.com.
HTTP.REQ.URL.HOSTNAME.PORT	Example: HTTP.REQ.URL.HOSTNAME.PORT.EQ(80) The above example returns true if the port is 80. It returns number on the port part of the hostname.
HTTP.REQ.URL.PATH	Example: HTTP.REQ.URL.PATH.GET(1) If the URL is http://www.foo.com/a/b/c/bar.html?a=1 then the operation will select /a/b/c/bar.html, then the above example will result in “a”. It returns/separated List on the path component of the URL.
HTTP.REQ.URL.PATH_AND_QUERY	Example: HTTP.REQ.URL.PATH_AND_QUERY If the URL is http://www.foo.com/a/b/c/bar.html?a=1 then it will return /a/b/c/bar.html?a=1. It returns the portion of the URL following the hostname
HTTP.REQ.URL.PROTOCOL	Example: HTTP.REQ.URL.PROTOCOL If the URL is http://www.foo.com/a/b/c/bar.html?a=1 then the operation will result in HTTP. It results in the protocol present in the URL.
HTTP.REQ.URL.QUERY	Example: HTTP.REQ.URL.QUERY If the URL is http://www.foo.com?abc=1&def=2 will result in abc=1&def=2. It results as Name-Value List (with delimiters = and &) in the query component of the URL.
HTTP.REQ.URL.QUERY.VALUE	Example: HTTP.REQ.URL.QUERY.VALUE(0) If the URL is http://www.foo.com?abc=1&def=2 will result in 1. It returns the value component of the specified name-value component in the list.
HTTP.REQ.URL.SUFFIX	Example: HTTP.REQ.URL.SUFFIX If the path is /a/b/c.html then this operation will result in html. It returns filename suffice of the URL.
HTTP.REQ.USER	Example: HTTP.REQ.USER.GROUPS('grp1:grp2') The above example will return list on the Group which is separated by given delimiter I.e. “:" It returns the AAA User associated with the current HTTP transaction.
HTTP.REQ.USER.EXTERNAL_GROUPS	Example: HTTP.REQ.USER.EXTERNAL_GROUPS The above example will list external groups which are separated by “,". IT returns a list of external groups which are separated by “,".
HTTP.REQ.USER.EXTERNAL_GROUPS.IGNORE_EMPTY_ELEMENTS	Example: HTTP.REQ.USER.EXTERNAL_GROUPS. IGNORE_EMPTY_ELEMENTS.COUNT If AAA User associated with the current HTTP transaction is part of some external groups : 123,,24, ,15 then HTTP.REQ.USER.EXTERNAL_GROUPS.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	IGNORE_EMPTY_ELEMENTS.COUNT gives 4, whereas HTTP.REQ.USER.EXTERNAL_GROUPS.COUNT gives 5. It ignores empty elements in the list.
HTTP.REQ.USER.EXTERNAL_GROUPS(sep)	Example: HTTP.REQ.USER.EXTERNAL_GROUPS(“:”) The above example will list external groups which are separated by “:” It returns a list of external groups which are separated by given delimiter.
HTTP.REQ.USER.GROUPS	Example: HTTP.REQ.USER.GROUPS The above example will list groups which are separated by “,". IT returns a list of groups which are separated by “,".
HTTP.REQ.USER.GROUPS.IGNORE_EMPTY_ELEMENTS	Example: HTTP.REQ.USER.GROUPS. IGNORE_EMPTY_ELEMENTS.COUNT If AAA User associated with the current HTTP transaction is part of some groups : 123,,24, ,15 then HTTP.REQ.USER.GROUPS. IGNORE_EMPTY_ELEMENTS.COUNT gives 4, whereas HTTP.REQ.USER.GROUPS.COUNT gives 5. It ignores empty elements in the list.
HTTP.REQ.USER.GROUPS(sep)	Example: HTTP.REQ.USER.GROUPS(“:”) The above example will list groups which are separated by “:" IT returns a list of groups which are separated by given delimiter.
HTTP.REQ.USER.INTERNAL_GROUPS	Example: HTTP.REQ.USER.INTERNAL_GROUPS The above example will list internal groups which are separated by “,". IT returns a list of internal groups which are separated by “,".
HTTP.REQ.USER.INTERNAL_GROUPS.IGNORE_EMPTY_ELEMENTS	Example: HTTP.REQ.USER.INTERNAL_GROUPS. IGNORE_EMPTY_ELEMENTS.COUNT If AAA User associated with the current HTTP transaction is part of some internal groups : 123,,24, ,15 then HTTP.REQ.USER.INTERNAL_GROUPS.IGNORE_EMPTY_ELEMENTS.COUNT gives 4, whereas HTTP.REQ.USER.INTERNAL_GROUPS.COUNT gives 5. It ignores empty elements in the list.
HTTP.REQ.USER.INTERNAL_GROUPS(sep)	Example: HTTP.REQ.USER.INTERNAL_GROUPS(“:”) The above example will list internal groups which are separated by “:" IT returns a list of internal groups which are separated by given delimiter.
HTTP.REQ.USER.IS_MEMBER_OF(group_name)	Example: HTTP.REQ.USER.IS_MEMBER_OF(grp1) The above example returns true is the current AAA user is a member of group grp1. It returns TRUE if the user is a member of the group group_name.
HTTP.REQ.USER.NAME	Example: HTTP.REQ.USER.NAME The above example will return the name of the user. It returns the name of the user. This is the name used by the user for login unless it is overridden by name from the external authentication server.
HTTP.REQ.USER.PASSWD	Example: HTTP.REQ.USER.PASSWD The above example will return password of the user. It returns the password of the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	user.
HTTP.REQ.VERSION	Example: HTTP.REQ.VERSION The above example returns HTTP version information.
HTTP.RES.BODY(<integer>)	Example: HTTP.RES.BODY(100) It will return first 100 characters of HTTP Response body. If the length of the body is less than 100 then the whole body will result as output.
HTTP.RES.STATUS_MSG	Example: HTTP.RES.STATUS_MSG The above example results status message of response. It can be “OK”, some error etc.
HTTP.RES.VERSION	Example: HTTP.RES.VERSION The above example returns HTTP version information.
HTTP.REQ.URL.HOSTNAME.EQ(<hostname>)	Example: HTTP.REQ.URL.HOSTNAME.EQ("abc.foo.com:8080") The above example returns true if hostname in URL is abc.foo.com:8080. It returns HTTP Host Name present in the URL.
HTTP.REQ.IS_NTLM_OR_NEGOTIATE	Example: HTTP.REQ.IS_NTLM_OR_NEGOTIATE The above example returns TRUE if the request is part of NTLM or NEGOTIATE connection.
HTTP.REQ.URL.PATH.IGNORE_EMPTY_ELEMENTS	Example: HTTP.REQ.URL.PATH. IGNORE_EMPTY_ELEMENTS.COUNT If request URL has path (/123//24//15) elements as : 123,,24, ,15 then HTTP.REQ.URL.PATH.IGNORE_EMPTY_ELEMENTS.COUNT gives 4, whereas HTTP.REQ.URL.PATHS.COUNT gives 5. It ignores empty elements in the list.
HTTP.REQ.URL.QUERY.IGNORE_EMPTY_ELEMENTS	Example: HTTP.REQ.URL.QUERY.IGNORE_EMPTY_ELEMENTS.COUNT If request URL has path as : abc=1&&def=2&g=3&h=6 then HTTP.REQ.URL.QUERY.IGNORE_EMPTY_ELEMENTS.COUNT gives 4, whereas HTTP.REQ.URL.QUERY.COUNT gives 5. It ignores empty elements in the list.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefixes for VPNs and Clientless VPNs

Aug 30, 2013

The default syntax expression engine provides prefixes that are specific to parsing VPN or Clientless VPN data. This data includes the following:

- Host names, domains, and URLs in VPN traffic.
- Protocols in the VPN traffic.
- Queries in the VPN traffic.

These text elements are often URLs and components of URLs. In addition to applying the text-based operations on these elements, you can parse these elements by using operations that are specific to parsing URLs. For more information, see "Expressions for Extracting Segments of URLs."

The following table describes the expression prefixes for this type of data.

Table 1. VPN and Clientless VPN Expression Prefixes That Return Text

VPN and Clientless VPN Expression	Description
VPN.BASEURL.CVPN_DECODE	Extracts the original URL from a clientless VPN URL.
VPN.BASEURL.CVPN_ENCODE	Converts a URL to clientless VPN format.
VPN.BASEURL.HOSTNAME	Extracts the HTTP host name from the host name in the URL. This prefix cannot be used in bidirectional policies.
VPN.BASEURL.HOSTNAME.DOMAIN	Extracts the domain name from the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:8080, this prefix extracts mycompany.com. This prefix returns incorrect results if the host name is an IP address. For information on expressions for IP addresses, see "Default Syntax Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs." All text operations after this prefix are case insensitive.
VPN.BASEURL.HOSTNAME.EQ (<hostname>)	Returns a Boolean TRUE if the host name matches <hostname>. The comparison is case insensitive. For example, if the host name is www.mycompany.com, the following returns TRUE: vpn.baseurl.hostname.eq("www.mycompany.com") If the text mode is URLENCODED, the host name is decoded before comparison. For more information, see "Operations for HTTP, HTML, and XML Encoding and "Safe" Characters."
VPN.BASEURL.HOSTNAME.SERVER	Evaluates the server portion of the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:8080, the server is www.mycompany.com. All text operations after this prefix are case insensitive.
VPN.BASEURL.PATH	Extracts a slash- (/) separated list from the path component of the URL. For example, this prefix extracts /a/b/c/mypage.html from the following URL: http://www.mycompany.com/a/b/c/mypage.html?a=1 The following expression selects just the "a":

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VPN and Clientless VPN Expression	Description
	http.req.url.path.get(1) For more information on the GET operation, see "Expressions for Extracting Segments of URLs."
VPN.BASEURL.PATH.IGNORE_EMPTY_ELEMENTS	This prefix ignores the elements in a list. For example, the following comma-separated list has an empty element after “a=10”: a=10,,b=11, ,c=89 The element following b=11 contains a space, and by default, is not considered an empty element. Consider the following HTTP header: Cust_Header : 123,,24, ,15 The following expression returns a count of 4 when evaluating this header: http.req.header("Cust_Header").typecase_list_t(',').ignore_empty_elements.count The following expression returns a count of 5 when evaluating this header: http.req.header("Cust_Header").typecase_list_t(','). count
VPN.BASEURL.PATH_AND_QUERY	Evaluates the text in the URL that follows the host name. For example, if the URL is http://www.mycompany.com/a/b/c/mypage.html?a=1, this prefix evaluates /a/b/c/mypage.html?a=1.
VPN.BASEURL.PROTOCOL	Evaluates the protocol in the URL. Do not use this prefix in bidirectional policies.
VPN.BASEURL.QUERY	Extracts a name-value list, using the “=” and “&” delimiters from the query string in a URL.
VPN.BASEURL.QUERY.IGNORE_EMPTY_ELEMENTS	This method ignores the empty elements in a name-value list. For example, in the following name-value list, there is an empty element following “a=10”: a=10;;b=11; ;c=89 The element following b=11 contains a space and is not considered an empty element. Consider the following HTTP header: Cust_Header : a=1;;b=2; ;c=3 The following expression produces a count of 4 after evaluating this header: http.req.header("Cust_Header").typecast_nvlist_t('=',';').ignore_empty_elements.count The following expression produces a count of 5 after evaluating the header: http.req.header("Cust_Header").typecast_nvlist_t('=',';').
VPN.BASEURL.SUFFIX	Evaluates the file name suffix in a URL. For example, if the path is /a/b/c/my.page.html, this operation selects “html.”
VPN.CLIENTLESS_BASEURL	Evaluates the clientless VPN base URL.
VPN.CLIENTLESS_BASEURL.CVPN_DECODE	Extracts the original URL from the clientless VPN formatted URL.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VPN and Clientless VPN Expression	Description
VPN.CLIENTLESS_BASEURL.CVPN_ENCODE	Converts a URL to the clientless VPN format.
VPN.CLIENTLESS_BASEURL.HOSTNAME	Evaluates the host name in the URL. Do not use this prefix in bidirectional policies.
VPN.CLIENTLESS_BASEURL.HOSTNAME.DOMAIN	Evaluates the domain name part of the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:8080, the domain is mycompany.com. This operation returns incorrect results if the host name is an IP address. For information on expressions for IP addresses, see "Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs." All text operations after this prefix are case insensitive.
VPN.CLIENTLESS_BASEURL.HOSTNAME.EQ(<hostname>)	Returns a Boolean TRUE if the host name matches <hostname>. For example, if the host name is www.mycompany.com or www.mycompany.com:8080, the following is true: vpn.clientless_baseurl. hostname.eq("www.mycompany.com") The comparison is case insensitive. If the textmode is URLENCODED, the host name is decoded before comparison. For more information, see "Operations for HTTP, HTML, and XML Encoding and "Safe" Characters."
VPN.CLIENTLESS_BASEURL.HOSTNAME.SERVER	Evaluates the server part of a host name. For example, if the host name is www.mycompany.com or www.mycompany.com:8080, the server is www.mycompany.com. All text operations after this prefix are case insensitive.
VPN.CLIENTLESS_BASEURL.PATH	Evaluates a slash- (/) separated list in the URL path. For example, this prefix selects /a/b/c/mypage.html from the following URL: http://www.mycompany.com/a/b/c/mypage.html?a=1 The following expression selects "a" from the preceding URL: http.req.url.path.get(1) For more information on the GET operation, see "Expressions for Extracting Segments of URLs."
VPN.CLIENTLESS_BASEURL.PATH.IGNORE_EMPTY_ELEMENTS	Ignores empty elements in a list. For example, if the list delimiter is a comma (,) the following list has an empty element following "a=10": a=10,b=11, ,c=89 The element following b=11 contains a space and is not considered an empty element. Consider the following HTTP header: Cust_Header : 123,,24, ,15 The following expression returns a value of 4 after evaluating this header: http.req.header("Cust_Header").typecast_list_t(',').ignore_empty_elements.count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VPN and Clientless VPN Expression	Description
	The following expression returns a value of 5 after evaluating this header: http.req.header("Cust_Header").typecast_list_t(',',).
VPN.CLIENTLESS_BASEURL.PATH_AND_QUERY	Evaluates the text following the host name in a URL. For example, this prefix selects /a/b/c/mypage.html?a=1 from the following URL: http://www.mycompany.com/a/b/c/mypage.html?a=1
VPN.CLIENTLESS_BASEURL.PROTOCOL	Evaluates the protocol in the URL. Do not use this prefix in bidirectional policies.
VPN.CLIENTLESS_BASEURL.QUERY	Extracts a name-value list that uses the delimiters “=” and “&” from a URL query string.
VPN.CLIENTLESS_BASEURL.QUERY.IGNORE_EMPTY_ELEMENTS	Ignores empty elements in a name-value list. For example, the following list contains an empty element after “a=10”: a=10;;b=11; ;c=89 The element following b=11 contains a space and is not considered an empty element. As another example, consider the following http header: Cust_Header : a=1;;b=2; ;c=3 The following expression returns a value of 4 after evaluating the preceding header: http.req.header("Cust_Header").typecast_nvlist_t('=',';').ignore_empty_elements.count The following expression returns a value of 5 after evaluating the preceding header: http.req.header("Cust_Header").typecast_nvlist_t('=',';')
VPN.CLIENTLESS_BASEURL.SUFFIX	Evaluates the file suffix in a URL. For example, if the URL path is /a/b/c/mypage.html then this operation selects html.
VPN.CLIENTLESS_HOSTURL	Selects the clientless VPN host URL.
VPN.CLIENTLESS_HOSTURL.CVPN_DECODE	Selects the original URL from the clientless VPN formatted URL.
VPN.CLIENTLESS_HOSTURL.CVPN_ENCODE	Converts a URL to clientless VPN format.
VPN.CLIENTLESS_HOSTURL.HOSTNAME	Extracts the host name in the URL. Do not use this prefix in bidirectional policies.
VPN.CLIENTLESS_HOSTURL.HOSTNAME.DOMAIN	Extracts the domain name from the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:8080, the domain is mycompany.com. This operation returns incorrect results if the host name contains an IP address. For information on expressions for IP addresses, see "Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs." All text operations after this prefix are case insensitive.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VPN and Clientless VPN Expression	Description
VPN.CLIENTLESS_HOSTURL.HOSTNAME.EQ(<hostname>)	Results in Boolean TRUE if the host name matches the <hostname> argument. The comparison is case insensitive. For example, if the host name is www.mycompany.com or www.mycompany.com., the following expression returns TRUE: vpn.clilentless_hosturl. hostname.eq("www.mycompany.com") If the text mode is URLENCODED, the host name is decoded before comparison. For more information, see "Operations for HTTP, HTML, and XML Encoding and "Safe" Characters."
VPN.CLIENTLESS_HOSTURL.HOSTNAME.SERVER	Evaluates the server part of the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:8080, the server is www.mycompany.com. The comparison is case insensitive, and all text operations after this method are case insensitive.
VPN.CLIENTLESS_HOSTURL.PATH	Evaluates a slash- (/) separated list on the path component of the URL. For example, consider the following URL: http://www.mycompany.com/a/b/c/mypage.html?a=1 This prefix selects /a/b/c/mypage.html from the preceding URL.
VPN.CLIENTLESS_HOSTURL.PATH.IGNORE_EMPTY_ELEMENTS	This method ignores the empty elements in a list. For example, if the delimiter in a list is "," the following list contains an empty element after the entry "a=10": a=10,b=11, ,c=89 The element following b=11 contains a space and is not considered an empty element. Consider the following header: Cust_Header : 123,,24, ,15 The following expression returns a value of 4 for this header: http.req.header("Cust_Header").typecast_list_t(',''). ignore_empty_elements.count The following expression returns a value of 5 for the same header: http.req.header("Cust_Header").typecast_list_t(','').
VPN.CLIENTLESS_HOSTURL.PATH_AND_QUERY	Evaluates the portion of the URL that follows the host name. For example, consider the following URL: http://www.mycompany.com/a/b/c/mypage.html?a=1 This prefix returns /a/b/c/mypage.html?a=1 from the preceding URL.
VPN.CLIENTLESS_HOSTURL.PROTOCOL	Evaluates the protocol in the URL. Do not use this prefix in bidirectional policies.
VPN.CLIENTLESS_HOSTURL.QUERY	Extracts a name-value list, using the "=" and "&" delimiters from a URL query string.
VPN.CLIENTLESS_HOSTURL.QUERY.IGNORE_EMPTY_ELEMENTS	Ignores empty elements in a name-value list. For example, the following list uses a semicolon (;) delimiter. This list contains an empty element after "a=10":

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VPN and Clientless VPN Expression	Description
	a=10;;b=11; ;c=89 In the preceding example, the element following b=11 is not considered an empty element. Consider the following header: Cust_Header : a=1;;b=2; ;c=3 The following expression returns a value of 4 after evaluating this header: http.req.header("Cust_Header").typecast_nvlist_t('=',';').ignore_empty_elements.count The following expression returns a value of 5 after evaluating the same header: http.req.header("Cust_Header").typecast_nvlist_t('=',';')
VPN.CLIENTLESS_HOSTURL.SUFFIX	Extracts a file name suffix in a URL. For example, if the path is /a/b/c/my.page.html, this prefix selects html.
VPN.HOST.DOMAIN	Extracts the domain name part of the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:8080, the domain is mycompany.com. This prefix returns incorrect results if the host name contains an IP address. For information on expressions for IP addresses, see "Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs." All text operations after this prefix case insensitive.
VPN.HOST.EQ(<hostname>)	Returns a Boolean TRUE value if the host name matches the <hostname>. The comparison is case insensitive. For example, if the host name is www.mycompany.com or www.mycompany.com:8080, the following returns TRUE: vpn.host.eq("www.mycompany.com") If the text mode is URLENCODED the host name is decoded before comparison. For more information, see "Operations for HTTP, HTML, and XML Encoding and "Safe" Characters."
VPN.HOST.SERVER	Extracts the server name part of the host name. For example, if the host name is www.mycompany.com or www.mycompany.com:8080, the server is www.mycompany.com. All text operations after this prefix are case insensitive.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Basic Operations on Text

Aug 30, 2013

Basic operations on text include operations for string matching, calculating the length of a string, and controlling case sensitivity. You can include white space in a string that is passed as an argument to an expression, but the string cannot exceed 255 characters.

String Comparison Functions

The following table lists basic string matching operations in which the functions return a Boolean TRUE or FALSE.

Table 1. String Comparison Functions

Function	Description
<text>.CONTAINS(<string>)	Returns a Boolean TRUE value if the target contains <string>. Following is an example: http.req.url.contains(".jpeg")
<text>.EQ(<string>)	Returns a Boolean TRUE value if the target is an exact match with <string>. For example, the following expression returns a Boolean TRUE for a URL with a host name of “myhostabc”: http.req.url.hostname.eq("myhostabc")
<text>.STARTSWITH(<string>)	Returns a Boolean TRUE value if the target begins with <string>. For example, the following expression returns a Boolean TRUE for a URL with a host name of “myhostabc”: http.req.url.hostname.startswith("myhost")
<text>.ENDSWITH(<string>)	Returns a Boolean TRUE value if the target ends with <string>. For example, the following expression returns a Boolean TRUE for a URL with a host name of “myhostabc”: http.req.url.hostname.endswith("abc")
<text>.NE(<string>)	Returns a Boolean TRUE value if the prefix is not equal to the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
<text>.GT(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically greater than the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.
<text>.GE(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically greater than or equal to the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.
<text>.LT(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically lesser than the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.
<text>.LE(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically lesser than or equal to the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.

Calculating the Length of a String

The <text>.LENGTH operation returns a numeric value that is equal to the number of characters (not bytes) in a string:

```
<text>.LENGTH
```

For example, you may want to identify request URLs that exceed a particular length. Following is an expression that implements this example:

```
HTTP.REQ.URL.LENGTH < 500
```

After taking a count of the characters or elements in a string, you can apply numeric operations to them. For more information, see "Default Syntax Expressions: Working with Dates, Times, and Numbers."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Considering, Ignoring, and Changing Text Case

The following functions operate on the case (upper-case or lower-case) of the characters in the string.

Table 2. Functions for Considering, Ignoring, and Changing Text Case

Function	Description
<text>.SET_TEXT_MODE(IGNORECASE|NOIGNORECASE)	This function turns case sensitivity on or off for all text operations.
<text>.TO_LOWER	Converts the target to lowercase for a text block of up to 2 kilobyte (KB). Returns UNDEF if the target exceeds 2 KB. For example, the string “ABCd:” is converted to “abcd:”.
<text>.TO_UPPER	Converts the target to uppercase. Returns UNDEF if the target exceeds 2 KB. For example, the string “abcD:” is converted to “ABCD:”.

Stripping Specific Characters from a String

You can use the STRIP_CHARS(<string>) function to remove specific characters from the text that is returned by a default syntax expression prefix (the input string). All instances of the characters that you specify in the argument are stripped from the input string. You can use any text method on the resulting string, including the methods used for matching the string with a pattern set.

For example, in the expression CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_"), the STRIP_CHARS(<string>) function strips all periods (.), hyphens (-), and underscores (_) from the domain name returned by the prefix CLIENT.UDP.DNS.DOMAIN. If the domain name that is returned is "a.dom_ai_n-name", the function returns the string "adomainname".

In the following example, the resulting string is compared with a pattern set called "listofdomains":

CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_").CONTAINS_ANY("listofdomains")

Note: You cannot perform a rewrite on the string that is returned by the STRIP_CHARS(<string>) function.
The following functions strip matching characters from the beginning and end of a given string input.

Table 3. Functions for Stripping Characters From the Beginning or End of a String

Function	Description
<text>.STRIP_START_CHARS(s)	Strips matching characters from the beginning of the input string until the first non-matching character is found and returns the remainder of the string. You must specify the characters that you want to strip as a single string within quotation marks.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	For example, if the name of a header is TestLang and ://_en_us: is its value, HTTP.RES.HEADER("TestLang").STRIP_START_CHARS(":/_") strips the specified characters from the beginning of the value of the header until the first non-matching character e is found and returns en_us: as a string.
<text>.STRIP_END_CHARS(s)	Strips matching characters from the end of the input string to the first non-matching character is found and returns the remainder of the string. You must specify the characters that you want to strip as a single string within quotation marks. For example, if the name of a header is TestLang and ://_en_us: is its value, HTTP.RES.HEADER("TestLang").STRIP_START_CHARS(":/_") strips the specified characters from the end of the value of the header until the first non-matching character s is found and returns ://_en_us as a string.

Appending a String to Another String

You can use the APPEND() function to append the string representation of the argument to the string representation of the value returned by the preceding function. The preceding function can be one that returns a number, unsigned long, double, time value, IPv4 address, or IPv6 address. The argument can be a text string, number, unsigned long, double, time value, IPv4 address, or IPv6 address. The resulting string value is the same string value that is obtained by using the + operator.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Complex Operations on Text

May 25, 2015

In addition to performing simple string matching, you can configure expressions that examine more complex aspects of text, including examining the length of a string and looking within a text block for patterns rather than specific strings.

Be aware of the following for any text-based operation:

- For any operation that takes a string argument, the string cannot exceed 255 characters.
- You can include white space when you specify a string in an expression.

This document includes the following details:

- Operations on the Length of a String
- Operations on a Portion of a String
- Operations for Comparing the Alphanumeric Order of Two Strings
- Extracting an Integer from a String of Bytes That Represent Text
- Converting Text to a Hash Value
- Encoding and Decoding Text by Applying the Base64 Encoding Algorithm
- Refining the Search in a Rewrite Action by Using the EXTEND Function
- Converting Text to Hexadecimal Format
- Encrypting and Decrypting Text

Operations on the Length of a String

The following operations extract strings on the basis of a character count.

Table 1. String Operations Based on a Character Count

Character Count Operation	Description
<text>.TRUNCATE(<count>)	Returns a string after truncating the end of the target by the number of characters in <count>. If the entire string is shorter than <count>, nothing is returned.
<text>.TRUNCATE(<character>, <count>)	Returns a string after truncating the text after <character> by the number of characters specified in <count>.
<text>.PREFIX(<character>, <count>)	Selects the longest prefix in the target that has at most <count> occurrences of <character>.
	<text>.SUFFIX(<character>, <count>) Selects the longest suffix in the target that has at most <count> occurrences of <character>. For example, consider the following response body: JLEwx

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Character Count Operation	Description
	The following expression returns a value of “JLEwx”: http.res.body(100).suffix('L',1) The following expression returns “LLEwx”: http.res.body(100).suffix('L',2)
<text>.SUBSTR(<starting_offset>, <length>)	Select a string with <length> number of characters from the target object. Begin extracting the string after the <starting_offset>. If the number of characters after the offset are fewer than the value of the <length> argument, select all the remaining characters.
<text>.SKIP(<character>, <count>)	Select a string from the target after skipping over the longest prefix that has at most <count> occurrences of <character>.

Operations on a Portion of a String

You can extract a subset of a larger string by using one of the operations in the following table.

Table 2. Basic Operations on a Portion of a String

Basic Text Operation	Description
<text>.BEFORE_STR(<string>)	Returns the text that precedes the first occurrence of <string>. If there is no match for <string>, the expression returns a text object of 0 length. Following is an example: http.res.body(1024).after_str("start_string").before_str("end_string").contains("https")
<text>.AFTER_STR(<string>)	Returns the text that follows the first occurrence of <string>. If there is no match for <string>, the expression returns a text object of 0 length. Following is an example: http.res.body(1024).after_str("start_string").before_str("end_string").contains("https")
<text>.BETWEEN(<starting string>, <ending string>)	Returns a Boolean TRUE value if the length of the text object is greater than or equal to the sum <starting string>, <ending string> argument lengths, and if a prefix of the target matches <starting string>, and if the suffix of the target matches <ending string>.
<text>.PREFIX(<prefix length>)	Returns the starting string from a target block of text that contains the number of characters in the <prefix length> argument.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Basic Text Operation	Description
	If the <prefix length> argument exceeds the number of characters in the target, the entire string is selected.
<text>.SUFFIX(<suffix length>)	Returns the ending string from a target block of text that contains the number of characters in the <suffix length> argument. If the <suffix length> argument exceeds the number characters in the target, the entire string is selected.
<text>.SUBSTR(<string>)	Select the first block of text in the target that matches the <string>.
<text>.SKIP(<prefix length>)	Selects the text in the target after skipping over a <prefix length> number of characters. If the entire target has fewer characters than <prefix length>, the entire target is skipped.
<text>.STRIP_END_WS	Selects the text after removing white space from the end of the target.
<text>.STRIP_START_WS	Selects the text after removing white space from the beginning of the target.
<text>.UNQUOTE(<character>)	Selects the <character>, removes white space that immediately precedes and follows the <character>, and if the remaining text is quoted by <character>, this prefix also removes the quotes. For example, the operation UNQUOTE("") changes the following text: "abc xyz def " To the following: abc xyz def

Operations for Comparing the Alphanumeric Order of Two Strings

The COMPARE operation examines the first nonmatching character of two different strings. This operation is based on lexicographic order, which is the method used when ordering terms in dictionaries.

This operation returns the arithmetic difference between the ASCII values of the first nonmatching characters in the compared strings. The following differences are examples:

- The difference between “abc” and “abd” is -1 (based on the third pair-wise character comparison).
- The difference between “@” and “abc” is -33.
- The difference between “1” and “abc” is -47.

Following is the syntax for the COMPARE operation.

<text>.COMPARE(<string>)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Extracting an Integer from a String of Bytes That Represent Text

You can use the following functions to treat a string of bytes that represent text as a sequence of bytes, extract 8, 16, or 32 bits from the sequence, and then convert the extracted bits to an integer.

Table 3. Operations for Extracting an Integer from a String of Bytes That Represent Text

Function	Description
<text>.GET_SIGNED8(<n>)	Treats the string of bytes represented by text as a sequence of 8-bit signed integers and returns the integer at byte offset n. If the offset makes all or part of the value outside of the current text, an UNDEF condition is raised.
<text>.GET_UNSIGNED8(<n>)	Treats the string of bytes represented by text as a sequence of 8-bit unsigned integers and returns the integer at byte offset n. If the offset makes all or part of the value outside of the current text, an UNDEF condition is raised.
<text>.GET_SIGNED16(<n>, <endianness>)	Treats the text string returned by the prefix as a string of bytes, extracts 16 bits starting at byte offset n, and converts the extracted bit sequence to a 16-bit signed integer. If the offset makes all or part of the value outside of the current text, an UNDEF condition is raised. The first parameter n is the byte offset from the current position in the text string. Providing a byte offset enables the function to handle items that are not aligned on the boundaries that are required by indexes. The second parameter, endianness, takes a mnemonic value of LITTLE_ENDIAN or BIG_ENDIAN. Note: In NetScaler 9.2, the parameter n was an index into an array of 16-bit items. In NetScaler 9.3, the parameter is a byte offset. Therefore, if you used this function in NetScaler 9.2, after you upgrade to NetScaler 9.3, you must change n to 2*n to obtain the same results as you did earlier. For example, if the value of n before the upgrade was 4, you must change the value of n to 8. The parameter endianness also no longer takes the values that it did in NetScaler 9.2, which were 0 and 1. Instead, endianness accepts the mnemonic values mentioned earlier. **Example** HTTP.REQ.BODY(100).GET_SIGNED16(8, BIG_ENDIAN)
<text>.GET_UNSIGNED16(<n>, <endianness>)	Treats the text string returned by the prefix as a string of bytes, extracts 16 bits starting at byte offset n, and converts the extracted bit sequence to a 16-bit unsigned integer. If the offset makes all or part of the value outside of the current text, an UNDEF condition is raised. The first parameter n is the byte offset from the current position in the text string. Providing a byte offset enables the function to handle items that are not aligned on the boundaries that are required by indexes. The second parameter, endianness, takes a mnemonic value of LITTLE_ENDIAN or BIG_ENDIAN.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	Note: In NetScaler 9.2, the parameter n was an index into an array of 16-bit items. In NetScaler 9.3, the parameter is a byte offset. Therefore, if you used this function in NetScaler 9.2, after you upgrade to NetScaler 9.3, you must change n to 2*n to obtain the same results as you did earlier. For example, if the value of n before the upgrade was 4, you must change the value of n to 8. The parameter endianness also no longer takes the values that it did in NetScaler 9.2, which were 0 and 1. Instead, endianness accepts the mnemonic values mentioned earlier. **Example** HTTP.REQ.BODY(100).GET_UNSIGNED16(8, LITTLE_ENDIAN)
<text>.GET_SIGNED32(<n>, <endianness>)	Treats the text string returned by the prefix as a string of bytes, extracts 32 bits starting at byte offset n, and converts the extracted bit sequence to a 32-bit signed integer. If the offset makes all or part of the value outside of the current text, an UNDEF condition is raised. The first parameter n is the byte offset from the current position in the text string. Providing a byte offset enables the function to handle items that are not aligned on the boundaries that are required by indexes. The second parameter, endianness, takes a mnemonic value of LITTLE_ENDIAN or BIG_ENDIAN. Note: In NetScaler 9.2, the parameter n was an index into an array of 32-bit items. In NetScaler 9.3, the parameter is a byte offset. Therefore, if you used this function in NetScaler 9.2, after you upgrade to NetScaler 9.3, you must change n to 4*n to obtain the same results as you did earlier. For example, if the value of n before the upgrade was 4, you must change the value of n to 16. The parameter endianness also no longer takes the values that it did in NetScaler 9.2, which were 0 and 1. Instead, endianness accepts the mnemonic values mentioned earlier. **Example** HTTP.REQ.BODY(1000).GET_SIGNED32(12, BIG_ENDIAN)
<text>.GET_UNSIGNED32(<n>, <endianness>)	Treats the text string returned by the prefix as a string of bytes, extracts 32 bits starting at byte offset n, and returns the extracted bit sequence as part of a 64-bit unsigned long integer. If the offset makes all or part of the value outside of the current text, an UNDEF condition is raised. The first parameter n is the byte offset from the current position in the text string. Providing a byte offset enables the function to handle items that are not aligned on the boundaries that are required by indexes. The second parameter, endianness, takes a mnemonic value of LITTLE_ENDIAN or BIG_ENDIAN. **Example** HTTP.REQ.BODY(1000).GET_UNSIGNED32(30, LITTLE_ENDIAN)

Converting Text to a Hash Value

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You can convert a text string to a hash value by using the HASH function. This function returns a 31-bit positive integer as a result of the operation. Following is the format of the expression:

<text>.HASH

This function ignores case and white spaces. For example, after the operation, the two strings Ab c and a bc would produce the same hash value.

Encoding and Decoding Text by Applying the Base64 Encoding Algorithm

The following two functions encode and decode a text string by applying the Base64 encoding algorithm

Table 4. Functions for Encoding and Decoding a Text String by Using Base64 Encoding

Function	Description
text.B64ENCODE	Encodes the text string (designated by text) by applying the Base64 encoding algorithm.
text.B64DECODE	Decodes the Base64-encoded string (designated by text) by applying the Base64 decoding algorithm. The operation raises an UNDEF if text is not in B64-encoded format.

Refining the Search in a Rewrite Action by Using the EXTEND Function

The EXTEND function is used in rewrite actions that specify patterns or pattern sets and target the bodies of HTTP packets. When a pattern match is found, the EXTEND function extends the scope of the search by a predefined number of bytes on both sides of the matching string. A regular expression can then be used to perform a rewrite on matches in this extended region. Rewrite actions that are configured with the EXTEND function perform rewrites faster than rewrite actions that evaluate entire HTTP bodies using only regular expressions.

The format of the EXTEND function is EXTEND(m,n), where m and n are the number of bytes by which the scope of the search is extended before and after the matching pattern, respectively. When a match is found, the new search scope comprises m bytes that immediately precede the matching string, the string itself, and the n bytes that follow the string. A regular expression can then be used to perform a rewrite on a portion of this new string.

The EXTEND function can be used only if the rewrite action in which it is used fulfills the following requirements:

- The search is performed by using patterns or patterns sets (not regular expressions)
- The rewrite action evaluates only the bodies of HTTP packets.

Additionally, the EXTEND function can be used only with the following types of rewrite actions:

- replace_all
- insert_after_all
- delete_all
- insert_before_all

For example, you might want to delete all instances of "http://exampleurl.com/" and "http://exampleurl.au/" in the first 1000 bytes of the body. To do this, you can configure a rewrite action to search for all instances of the string exampleurl, extend the scope of the search on both sides of the string when a match is found, and then use a regular expression to

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

perform the rewrite in the extended region. The following example extends the scope of the search by 20 bytes to the left and 50 bytes to the right of the matching string:

```
add rewrite action delurl_example delete_all 'HTTP.REQ.BODY(1000)' -pattern exampleurl -refineSearch
'extend(20,50).regex_select(re#http://exampleurl.(com|au)#)'
```

Converting Text to Hexadecimal Format

The following function converts text to hexadecimal format and extracts the resulting string:

```
<text>.BLOB_TO_HEX(<string>)
```

For example, this function converts the byte string “abc” to “61:62:63”.

Encrypting and Decrypting Text

Updated: 2013-09-02

In default syntax expressions, you can use the ENCRYPT and DECRYPT functions to encrypt and decrypt text. Data encrypted by the ENCRYPT function on a given NetScaler appliance or high availability (HA) pair is intended for decryption by the DECRYPT function on the same NetScaler appliance or HA pair. The appliance supports the RC4, DES3, AES128, AES192, and AES256 encryption methods. The key value that is required for encryption is not user-specifiable. When an encryption method is set, the appliance automatically generates a random key value that is appropriate for the specified method. The default method is AES256 encryption, which is the most secure encryption method and the one that Citrix recommends.

You do not need to configure encryption unless you want to change the encryption method or you want the appliance to generate a new key value for the current encryption method.

Note: You can also encrypt and decrypt XML payloads. For information about the functions for encrypting and decrypting XML payloads, see "Encrypting and Decrypting XML Payloads."

Configuring Encryption

Updated: 2013-09-02

During startup, the appliance runs the set ns encryptionParams command with, by default, the AES256 encryption method, and uses a randomly generated key value that is appropriate for AES256 encryption. The appliance also encrypts the key value and saves the command, with the encrypted key value, to the NetScaler configuration file. Consequently, the AES256 encryption method is enabled for the ENCRYPT and DECRYPT functions by default. The key value that is saved in the configuration file persists across reboots even though the appliance runs the command each time you restart it.

You can run the set ns encryptionParams command manually, or use the configuration utility, if you want to change the encryption method or if you want the appliance to generate a new key value for the current encryption method. To use the CLI to change the encryption method, set only the method parameter, as shown in "**Example 1: Changing the Encryption Method**." If you want the appliance to generate a new key value for the current encryption method, set the method parameter to the current encryption method and the keyValue parameter to an empty string ("""), as shown in "**Example 2: Generating a New Key Value for the Current Encryption Method**." After you generate a new key value, you must save the configuration. If you do not save the configuration, the appliance uses the newly generated key value only until the next restart, after which it reverts to the key value in the saved configuration.

To configure encryption by using the configuration utility

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to System > Settings.
2. In the Settings area, click Change Encryption parameters.
3. In the Change Encryption Parameters dialog box, do one of the following:
   - To change the encryption method, in the Method list, select the encryption method that you want.
   - To generate a new key value for the current encryption method, click Generate a new key for the selected method.
4. Click OK.

Using the ENCRYPT and DECRYPT Functions

You can use the ENCRYPT and DECRYPT functions with any expression prefix that returns text. For example, you can use the ENCRYPT and DECRYPT functions by rewrite policies for cookie encryption. In the following example, the rewrite actions encrypt a cookie named MyCookie, which is set by a back-end service, and decrypt the same cookie when it is returned by a client:

```
add rewrite action my-cookie-encrypt-action replace "HTTP.RES.SET_COOKIE.COOKIE(\"MyCookie\").VALUE(0)"
"HTTP.RES.SET_COOKIE.COOKIE(\"MyCookie\").VALUE(0).ENCRYPT" -bypassSafetyCheck YES
```

```
add rewrite action my-cookie-decrypt-action replace "HTTP.REQ.COOKIE.VALUE(\"MyCookie\")"
"HTTP.REQ.COOKIE.VALUE(\"MyCookie\").DECRYPT" -bypassSafetyCheck YES
```

After you configure policies for encryption and decryption, save the configuration to bring the policies into effect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Default Syntax Expressions: Working with Dates, Times, and Numbers

Sep 02, 2013

Most numeric data that the NetScaler appliance processes consists of dates and times. In addition to working with dates and times, the appliance processes other numeric data, such as the lengths of HTTP requests and responses. To process this data, you can configure default syntax expressions that process numbers.

A numeric expression consists of an expression prefix that returns a number and sometimes, but not always, an operator that can perform an operation on the number. Examples of expression prefixes that return numbers are SYS.TIME.DAY, HTTP.REQ.CONTENT_LENGTH, and HTTP.RES.BODY.LENGTH. Numeric operators can work with any prefix expression that returns data in numeric format. The GT(<int>) operator, for example, can be used with any prefix expression, such as HTTP.REQ.CONTENT_LENGTH, that returns an integer. Numeric expression prefixes and operators are also covered in "Compound Operations for Numbers" and "Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Format of Dates and Times in an Expression

Nov 14, 2013

When configuring a default syntax expression in a policy that works with dates and times (for example, the NetScaler system time or a date in an SSL certificate), you specify a time format as follows:

GMT|LOCAL [<yyyy>] [<month>] [<d>] [<h>] [<m>] [<s>]

Where:

- <yyyy> is a four-digit year after GMT or LOCAL.
- <month> is a three-character abbreviation for the month, for example, Jan, Dec.
- <d> is a day of the week or an integer for the date.
  You cannot specify the day as Monday, Tuesday, and so on. You specify either an integer for a specific day of the month, or you specify a date as the first, second, third weekday of the month, and so on. Following are examples of specifying a day of the week:
  - Sun_1 is the first Sunday of the month.
  - Sun_3 is the third Sunday of the month.
  - Wed_3 is the third Wednesday of the month.
  - 30 is an example of an exact date in a month.
- <h> is the hour, for example, 10h.
- <s> is the number of seconds, for example, 30s.

The following example expression is true if the date is between 2008 Jan and 2009 Jan, based on GMT.

http.req.date.between(GMT 2008 Jan, GMT 2009 Jan)

The following example expression is true for March and all months that follow March in the calendar year, based on GMT:

sys.time.ge(GMT 2008 Mar)

When you specify a date and time, note that the format is case sensitive and must preserve the exact number of blank spaces between entries.

Note: In an expression that requires two time values, both must use GMT or both must use LOCAL. You cannot mix the two in an expression.
Note: Unlike when you use the SYS.TIME prefix in a default syntax expression, if you specify SYS.TIME in a rewrite action, the NetScaler returns a string in conventional date format (for example, Sun, 06 Nov 1994 08:49:37 GMT). For example, the following rewrite action replaces the http.res.date header with the NetScaler system time in a conventional date format: add rewrite action sync_date replace http.res.date sys.time

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions for the NetScaler System Time

Mar 20, 2012

The SYS.TIME expression prefix extracts the NetScaler system time. You can configure expressions that establish whether a particular event occurred at a particular time or within a particular time range according to the NetScaler system time.

The following table describes the expressions that you can create by using the SYS.TIME prefix.

Table 1. Expressions That Return NetScaler System Dates and Times

NetScaler Time Operation	Description
SYS.TIME.BETWEEN(<time1>, <time2>)	Returns a Boolean TRUE if the returned value is later than <time1> and earlier than <time2>. You format the <time1>, <time2> arguments as follows: • They must both be GMT or both LOCAL. • <time2> must be later than <time1>. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following: • sys.time.between(GMT 2004, GMT 2006) • sys.time.between(GMT 2004 Jan, GMT 2006 Nov) • sys.time.between(GMT 2004 Jan, GMT 2006) • sys.time.between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) • sys.time.between(GMT 2005 May 1, GMT May 2005 1) • sys.time.between(LOCAL 2005 May 1, LOCAL May 2005 1)
SYS.TIME.DAY	Returns the current day of the month as a number from 1 through 31.
SYS.TIME.EQ(<time>)	Returns a Boolean TRUE if the current time is equal to the <time> argument. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.eq(GMT 2005) (TRUE in this example.) • sys.time.eq(GMT 2005 Dec) (FALSE in this example.) • sys.time.eq(LOCAL 2005 May) (Evaluates to TRUE or FALSE in this example, depending on the current time zone.) • sys.time.eq(GMT 10h) (TRUE in this example.) • sys.time.eq(GMT 10h 30s) (TRUE in this example.) • sys.time.eq(GMT May 10h) (TRUE in this example.) • sys.time.eq(GMT Sun) (TRUE in this example.) • sys.time.eq(GMT May Sun_1) (TRUE in this example.)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler Time Operation	Description
SYS.TIME.NE(<time>)	Returns a Boolean TRUE if the current time is not equal to the <time> argument.
SYS.TIME.GE(<time>)	Returns a Boolean TRUE if the current time is later than or equal to <time>. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.ge(GMT 2004) (TRUE in this example.) • sys.time.ge(GMT 2005 Jan) (TRUE in this example.) • sys.time.ge(LOCAL 2005 May) (TRUE or FALSE in this example, depending on the current time zone.) • sys.time.ge(GMT 8h) (TRUE in this example.) • sys.time.ge(GMT 30m) (FALSE in this example.) • sys.time.ge(GMT May 10h) (TRUE in this example.) • sys.time.ge(GMT May 10h 0m) (TRUE in this example.) • sys.time.ge(GMT Sun) (TRUE in this example.) • sys.time.ge(GMT May Sun_1) (TRUE in this example.)
SYS.TIME.GT(<time>)	Returns a Boolean TRUE if the time value is later than the <time> argument. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.gt(GMT 2004) (TRUE in this example.) • sys.time.gt(GMT 2005 Jan) (TRUE in this example.) • sys.time.gt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.) • sys.time.gt(GMT 8h) (TRUE in this example.) • sys.time.gt(GMT 30m) (FALSE in this example.) • sys.time.gt(GMT May 10h) (FALSE in this example.) • sys.time.gt(GMT May 10h 0m) (TRUE in this example.) • sys.time.gt(GMT Sun) (FALSE in this example.) • sys.time.gt(GMT May Sun_1) (FALSE in this example.)
SYS.TIME.HOURS	Returns the current hour as an integer from 0 to 23.
SYS.TIME.LE(<time>)	Returns a Boolean TRUE if the current time value precedes or is equal to the <time> argument. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler Time Operation	Description
	• sys.time.le(GMT 2006) (TRUE in this example.) • sys.time.le(GMT 2005 Dec) (TRUE in this example.) • sys.time.le(LOCAL 2005 May) (TRUE or FALSE depending on the current timezone.) • sys.time.le(GMT 8h) (FALSE in this example.) • sys.time.le(GMT 30m) (TRUE in this example.) • sys.time.le(GMT May 10h) (TRUE in this example.) • sys.time.le(GMT Jun 11h) (TRUE in this example.) • sys.time.le(GMT Wed) (TRUE in this example.) • sys.time.le(GMT May Sun_1) (TRUE in this example.)
SYS.TIME.LT(<time>)	Returns a Boolean TRUE if the current time value precedes the <time> argument. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.lt(GMT 2006) (TRUE in this example.) • sys.time.lt.time.lt(GMT 2005 Dec) (TRUE in this example.) • sys.time.lt(LOCAL 2005 May) (TRUE or FALSE depending on the current time zone.) • sys.time.lt(GMT 8h) (FALSE in this example.) • sys.time.lt(GMT 30m) (TRUE in this example.) • sys.time.lt(GMT May 10h) (FALSE in this example.) • sys.time.lt(GMT Jun 11h) (TRUE in this example.) • sys.time.lt(GMT Wed) (TRUE in this example.) • sys.time.lt(GMT May Sun_1) (FALSE in this example.)
SYS.TIME.MINUTES	Returns the current minute as an integer from 0 to 59.
SYS.TIME.MONTH	Extracts the current month and returns an integer from 1 (January) to 12 (December).
SYS.TIME.RELATIVE_BOOT	Calculates the number of seconds to the closest previous or scheduled reboot, and returns an integer. If the closest boot time is in the past, the integer is negative. If it is in the future, the integer is positive.
SYS.TIME.RELATIVE_NOW	Calculates the number of seconds between the current NetScaler system time and the specified time, and returns an integer showing the difference. If the designated time is in the past, the integer is negative; if it is in the future, the integer is positive.
SYS.TIME.SECONDS	Extracts the seconds from the current NetScaler system time, and returns that value

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler Time Operation	Description
SYS.TIME.WEEKDAY	Returns the current weekday as a value from 0 (Sunday) to 6 (Saturday).
SYS.TIME.WITHIN (<time1>, <time2>)	If you omit an element of time in <time1>, for example, the day or hour, it is assumed to have the lowest value in its range. If you omit an element in <time2>, it is assumed to have the highest value of its range. The ranges for the elements of time are as follows: month 1-12, day 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59. If you specify the year, you must do so in both <time1> and <time2>. For example, if the time is GMT 2005 May 10 10h 15m 30s, and it is the second Tuesday of the month, you can specify the following (evaluation results are shown in parentheses): • sys.time.within(GMT 2004, GMT 2006) (TRUE in this example.) • sys.time.within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, May is not in the range of January to March.) • sys.time.within(GMT Feb, GMT) (TRUE, May is in the range of February to December.) • sys.time.within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday is between the first Sunday and the third Sunday.) • sys.time.within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE in this example.) • sys.time.within(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the NetScaler system time zone.)
SYS.TIME.YEAR	Extracts the year from the current system time and returns that value as a four-digit integer.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions for SSL Certificate Dates

May 11, 2012

You can determine the validity period for SSL certificates by configuring an expression that contains the following prefix:

CLIENT.SSL.CLIENT_CERT

The following example expression matches a particular time for expiration with the information in the certificate:

client.ssl.client_cert.valid_not_after.eq(GMT 2009)

The following table describes time-based operations on SSL certificates. To obtain the expression you want, replace *certificate* in the expression in the first column with the prefix expression, "CLIENT.SSL.CLIENT_CERT".

Table 1. Operations on Certificate (client.ssl.client_cert) Dates and Times

SSL Certificate Operation	Description
<certificate>.VALID_NOT_AFTER	Returns the last day before certificate expiration. The return format is the number of seconds since GMT January 1, 1970 (0 hours, 0 minutes, 0 seconds).
<certificate>.VALID_NOT_AFTER.BETWEEN(<time1>, <time2>)	Returns a Boolean TRUE value if the certificate validity is between the <time1> and <time2> arguments. Both <time1> and <time2> must be fully specified. Following are examples: GMT 1995 Jan is fully specified. GMT Jan is not fully specified GMT 1995 20 is not fully specified. GMT Jan Mon_2 is not fully specified. The <time1> and <time2> arguments must be both GMT or both LOCAL, and <time2> must be greater than <time1>. For example, if it is GMT 2005 May 1 10h 15m 30s, and the first Sunday of the month, you can specify the following (evaluation results are in parentheses). • . . .between(GMT 2004, GMT 2006) (TRUE) • . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE) • . . .between(GMT 2004 Jan, GMT 2006) (TRUE) • . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE) • . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE) • . . .between(LOCAL 2005 May 1, LOCAL May 2005 1)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Certificate Operation	Description
	(TRUE or FALSE, depending on the NetScaler system time zone.)
<certificate>.VALID_NOT_AFTER.DAY	Extracts the last day of the month that the certificate is valid, and returns a number from 1 through 31, as appropriate for the date.
<certificate>.VALID_NOT_AFTER.EQ(<time>)	Returns a Boolean TRUE if the time is equal to the <time> argument. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results for this example are in parentheses): • . . .eq(GMT 2005) (TRUE) • . . .eq(GMT 2005 Dec) (FALSE) • . . .eq(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone) • . . .eq(GMT 10h) (TRUE) • . . .eq(GMT 10h 30s) (TRUE) • . . .eq(GMT May 10h) (TRUE) • . . .eq(GMT Sun) (TRUE) • . . .eq(GMT May Sun_1) (TRUE)
<certificate>.VALID_NOT_AFTER.GE(<time>)	Returns a Boolean TRUE if the time value is greater than or equal to the argument <time>. For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses): • . . .ge(GMT 2004) (TRUE) • . . .ge(GMT 2005 Jan) (TRUE) • . . .ge(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.) • . . .ge(GMT 8h) (TRUE) • . . .ge(GMT 30m) (FALSE) • . . .ge(GMT May 10h) (TRUE) • . . .ge(GMT May 10h 0m) (TRUE) • . . .ge(GMT Sun) (TRUE) • . . .ge(GMT May Sun_1) (TRUE)
<certificate>.VALID_NOT_AFTER.GT(<time>)	Returns a Boolean TRUE if the time value is greater than the argument <time>.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Certificate Operation	Description
	For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses): • . . .gt(GMT 2004) (TRUE) • . . .gt(GMT 2005 Jan) (TRUE) • . . .gt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.) • . . .gt(GMT 8h) (TRUE) • . . .gt(GMT 30m) (FALSE) • . . .gt(GMT May 10h) (FALSE) • . . .gt(GMT Sun) (FALSE) • . . .gt(GMT May Sun_1) (FALSE)
<certificate>.VALID_NOT_AFTER.HOURS	Extracts the last hour that the certificate is valid and returns that value as an integer from 0 to 23.
<certificate>.VALID_NOT_AFTER.LE(<time>)	Returns a Boolean TRUE if the time precedes or is equal to the <time> argument. For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses): • . . .le(GMT 2006) (TRUE) • . . .le(GMT 2005 Dec) (TRUE) • . . .le(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.) • . . .le(GMT 8h) (FALSE) • . . .le(GMT 30m) (TRUE) • . . .le(GMT May 10h) (TRUE) • . . .le(GMT Jun 11h) (TRUE) • . . .le(GMT Wed) (TRUE) • . . .le(GMT May Sun_1) (TRUE)
<certificate>.VALID_NOT_AFTER.LT(<time>)	Returns a Boolean TRUE if the time precedes the <time> argument. For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following: • . . .lt(GMT 2006) (TRUE) • . . .lt(GMT 2005 Dec) (TRUE) • . . .lt(LOCAL 2005 May) (TRUE or FALSE, depending on

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Certificate Operation	Description
	the current time zone.) ● . . .lt(GMT 8h) (FALSE) ● . . .lt(GMT 30m) (TRUE) ● . . .lt(GMT May 10h) (FALSE) ● . . .lt(GMT Jun 11h) (TRUE) ● . . .lt(GMT Wed) (TRUE) ● . . .lt(GMT May Sun_1) (FALSE)
<certificate>.VALID_NOT_AFTER.MINUTES	Extracts the last minute that the certificate is valid and returns that value as an integer from 0 to 59.
<certificate>.VALID_NOT_AFTER.MONTH	Extracts the last month that the certificate is valid and returns that value as an integer from 1 (January) to 12 (December).
<certificate>.VALID_NOT_AFTER.RELATIVE_BOOT	Calculates the number of seconds to the closest previous or scheduled reboot and returns an integer. If the closest boot time is in the past, the integer is negative. If it is in the future, the integer is positive.
<certificate>.VALID_NOT_AFTER.RELATIVE_NOW	Calculates the number of seconds between the current system time and the specified time and returns an integer. If the time is in the past, the integer is negative; if it is in the future, the integer is positive.
<certificate>.VALID_NOT_AFTER.SECONDS	Extracts the last second that the certificate is valid and returns that value as an integer from 0 to 59.
<certificate>.VALID_NOT_AFTER.WEEKDAY	Extracts the last weekday that the certificate is valid. Returns a number between 0 (Sunday) and 6 (Saturday) to give the weekday in the time value.
<certificate>.VALID_NOT_AFTER.WITHIN(<time1>, <time2>)	Returns a Boolean TRUE if the time lies within all the ranges defined by the elements in <time1> and <time2>. If you omit an element of time from <time1>, it is assumed to have the lowest value in its range. If you omit an element from <time2>, it is assumed to have the highest value of its range. If you specify a year in <time1>, you must specify it in <time2>. The ranges for elements of time are as follows: month 1-12, day 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Certificate Operation	Description
	0-59. For the result to be TRUE, each element in the time must exist in the corresponding range that you specify in <time1>, <time2>. For example, if time is GMT 2005 May 10 10h 15m 30s, and it is the second Tuesday of the month, you can specify the following (evaluation results are in parentheses): • . . .within(GMT 2004, GMT 2006) (TRUE) • . . .within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, May is not in the range of January to March.) • . . .within(GMT Feb, GMT) (TRUE, May is in the range for February to December) • . . .within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday lies within the range of the first Sunday through the third Sunday) • . . .within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE) • . . .within(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the NetScaler system time zone)
<certificate>.VALID_NOT_AFTER.YEAR	Extracts the last year that the certificate is valid and returns a four-digit integer.
<certificate>.VALID_NOT_BEFORE	Returns the date that the client certificate becomes valid. The return format is the number of seconds since GMT January 1, 1970 (0 hours, 0 minutes, 0 seconds).
<certificate>.VALID_NOT_BEFORE.BETWEEN(<time1>, <time2>)	Returns a Boolean TRUE if the time value is between the two time arguments. Both <time1> and <time2> arguments must be fully specified. Following are examples: • GMT 1995 Jan is fully specified. • GMT Jan is not fully specified. • GMT 1995 20 is not fully specified. • GMT Jan Mon_2 is not fully specified. The time arguments must be both GMT or both LOCAL, and <time2> must be greater than <time1>. For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Certificate Operation	Description
	• . . .between(GMT 2004, GMT 2006) (TRUE) • . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE) • . . .between(GMT 2004 Jan, GMT 2006) (TRUE) • . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE) • . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE) • . . .between(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the NetScaler system time zone.)
<certificate>.VALID_NOT_BEFORE.DAY	Extracts the last day of the month that the certificate is valid and returns that value as a number from 1 through 31 representing that day.
<certificate>.VALID_NOT_BEFORE.EQ(<time>)	Returns a Boolean TRUE if the time is equal to the <time> argument. For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses): • . . .eq(GMT 2005) (TRUE) • . . .eq(GMT 2005 Dec) (FALSE) • . . .eq(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.) • . . .eq(GMT 10h) (TRUE) • . . .eq(GMT 10h 30s) (TRUE) • . . .eq(GMT May 10h) (TRUE) • . . .eq(GMT Sun) (TRUE) • . . .eq(GMT May Sun_1) (TRUE)
<certificate>.VALID_NOT_BEFORE.GE(<time>)	Returns a Boolean TRUE if the time is greater than (after) or equal to the <time> argument. For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results are in parentheses): • . . .ge(GMT 2004) (TRUE) • . . .ge(GMT 2005 Jan) (TRUE) • . . .ge(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.) • . . .ge(GMT 8h) (TRUE) • . . .ge(GMT 30m) (FALSE)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Certificate Operation	Description
	• . . .ge(GMT May 10h) (TRUE) • . . .ge(GMT May 10h 0m) (TRUE) • . . .ge(GMT Sun) (TRUE) • . . .ge(GMT May Sun_1) (TRUE)
<certificate>.VALID_NOT_BEFORE.GT(<time>)	Returns a Boolean TRUE if the time occurs after the <time> argument. For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results are in parentheses): • . . .gt(GMT 2004) (TRUE) • . . .gt(GMT 2005 Jan) (TRUE) • . . .gt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.) • . . .gt(GMT 8h) (TRUE) • . . .gt(GMT 30m) (FALSE) • . . .gt(GMT May 10h) (FALSE) • . . .gt(GMT May 10h 0m) (TRUE) • . . .gt(GMT Sun) (FALSE) • . . .gt(GMT May Sun_1) (FALSE)
<certificate>.VALID_NOT_BEFORE.HOURS	Extracts the last hour that the certificate is valid and returns that value as an integer from 0 to 23.
<certificate>.VALID_NOT_BEFORE.LE(<time>)	Returns a Boolean TRUE if the time precedes or is equal to the <time> argument. For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses): • . . .le(GMT 2006) (TRUE) • . . .le(GMT 2005 Dec) (TRUE) • . . .le(LOCAL 2005 May) (TRUE or FALSE,depending on the current time zone.) • . . .le(GMT 8h) (FALSE) • . . .le(GMT 30m) (TRUE) • . . .le(GMT May 10h) (TRUE) • . . .le(GMT Jun 11h) (TRUE) • . . .le(GMT Wed) (TRUE) • . . .le(GMT May Sun_1) (TRUE)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Certificate Operation	Description
<certificate>.VALID_NOT_BEFORE.LT(<time>)	Returns a Boolean TRUE if the time precedes the <time> argument. For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses): • . . .lt(GMT 2006) (TRUE) • . . .lt(GMT 2005 Dec) (TRUE) • . . .lt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.) • . . .lt(GMT 8h) (FALSE) • . . .lt(GMT 30m) (TRUE) • . . .lt(GMT May 10h) (FALSE) • . . .lt(GMT Jun 11h) (TRUE) • . . .lt(GMT Wed) (TRUE) • . . .lt(GMT May Sun_1) (FALSE)
<certificate>.VALID_NOT_BEFORE.MINUTES	Extracts the last minute that the certificate is valid. Returns the current minute as an integer from 0 to 59.
<certificate>.VALID_NOT_BEFORE.MONTH	Extracts the last month that the certificate is valid. Returns the current month as an integer from 1 (January) to 12 (December).
<certificate>.VALID_NOT_BEFORE.RELATIVE_BOOT	Calculates the number of seconds to the closest previous or scheduled NetScaler reboot and returns an integer. If the closest boot time is in the past, the integer is negative; if it is in the future, the integer is positive.
<certificate>.VALID_NOT_BEFORE.RELATIVE_NOW	Returns the number of seconds between the current NetScaler system time and the specified time as an integer. If the designated time is in the past, the integer is negative. If it is in the future, the integer is positive.
<certificate>.VALID_NOT_BEFORE.SECONDS	Extracts the last second that the certificate is valid. Returns the current second as an integer from 0 to 59.
<certificate>.VALID_NOT_BEFORE.WEEKDAY	Extracts the last weekday that the certificate is valid. Returns the weekday as a number between 0 (Sunday) and 6 (Saturday).
<certificate>.VALID_NOT_BEFORE.WITHIN(<time1>,	Returns a Boolean TRUE if each element of time exists

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Certificate Operation	Description
<time2>)	within the range defined in the <time1>, <time2> arguments. If you omit an element of time from <time1>, it is assumed to have the lowest value in its range. If you omit an element of time from <time2>, it is assumed to have the highest value in its range. If you specify a year in <time1>, it must be specified in <time2>. The ranges for elements of time are as follows: month 1-12, day 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59. For example, if the time is GMT 2005 May 10 10h 15m 30s, and it is the second Tuesday of the month, you can specify the following (evaluation results are in parentheses): • . . .within(GMT 2004, GMT 2006) (TRUE) • . . .within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, May is not in the range of January to March.) • . . .within(GMT Feb, GMT) (TRUE, May is in the range of February to December.) • . . .within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday is between the first Sunday and the third Sunday.) • . . .within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE) • . . .within(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the NetScaler system time zone)
<certificate>.VALID_NOT_BEFORE.YEAR	Extracts the last year that the certificate is valid. Returns the current year as a four-digit integer.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions for HTTP Request and Response Dates

Sep 02, 2013

The following expression prefixes return the contents of the HTTP Date header as text or as a date object. These values can be evaluated as follows:

- As a number. The numeric value of an HTTP Date header is returned in the form of the number of seconds since Jan 1 1970.
  For example, the expression http.req.date.mod(86400) returns the number of seconds since the beginning of the day. These values can be evaluated using the same operations as other non-date-related numeric data. For more information, see "Expression Prefixes for Numeric Data Other Than Date and Time."
- As an HTTP header. Date headers can be evaluated using the same operations as other HTTP headers.
  For more information, see "Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data."
- As text. Date headers can be evaluated using the same operations as other strings.

For more information, see "Default Syntax Expressions: Evaluating Text."

Table 1. Prefixes That Evaluate HTTP Date Headers

Prefix	Description
HTTP.REQ.DATE	Returns the contents of the HTTP Date header as text or as a date object. The date formats recognized are: RFC822. Sun, 06 Jan 1980 08:49:37 GMT RFC850. Sunday, 06-Jan-80 09:49:37 GMT ASCTIME. Sun Jan 6 08:49:37 1980
HTTP.RES.DATE	Returns the contents of the HTTP Date header as text or as a date object. The date formats recognized are: RFC822. Sun, 06 Jan 1980 8:49:37 GMT RFC850. Sunday, 06-Jan-80 9:49:37 GMT ASCTIME. Sun Jan 6 08:49:37 1980

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Generating the Day of the Week, as a String, in Short and Long Formats

Mar 20, 2012

The functions, WEEKDAY_STRING_SHORT and WEEKDAY_STRING, generate the day of the week, as a string, in short and long formats, respectively. The strings that are returned are always in English. The prefix used with these functions must return the day of the week in integer format and the acceptable range for the value returned by the prefix is 0-6. Therefore, you can use any prefix that returns an integer in the acceptable range. An UNDEF condition is raised if the returned value is not in this range or if memory allocation fails.

Following are the descriptions of the functions:

Table 1. Functions That Generate the Day of the Week, as a String, in Short and Long Formats

Function	Description
<prefix>.WEEKDAY_STRING_SHORT	Returns the day of the week in short format. The short form is always 3 characters long with an initial capital and the remaining characters in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING_SHORT returns Sun if the value returned by the WEEKDAY function is 0 and Sat if the value returned by the prefix is 6.
<prefix>.WEEKDAY_STRING	Returns the day of the week in long format. The long form always has an initial capital, with the remaining characters in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING returns Sunday if the value returned by the WEEKDAY function is 0 and Saturday if the value returned by the prefix is 6.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefixes for Numeric Data Other Than Date and Time

Sep 02, 2013

In addition to configuring expressions that operate on time, you can configure expressions for the following types of numeric data:

- The length of HTTP requests, the number of HTTP headers in a request, and so on.
  For more information, see "Expressions for Numeric HTTP Payload Data Other Than Dates."
- IP and MAC addresses.
  For more information, see "Expressions for IP Addresses and IP Subnets."
- Client and server data in regard to interface IDs and transaction throughput rate.
  For more information, see "Expressions for Numeric Client and Server Data."
- Numeric data in client certificates other than dates.
  For information on these prefixes, including the number of days until certificate expiration and the encryption key size, see "Prefixes for Numeric Data in SSL Certificates."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Converting Numbers to Text

Sep 02, 2013

The following functions produce binary strings from a number returned by an expression prefix. These functions are particularly useful in the TCP rewrite feature as replacement strings for binary data. For more information about the TCP rewrite feature, see "Rewrite."

All the functions return a value of type text. The endianness that some of the functions accept as a parameter is either LITTLE_ENDIAN or BIG_ENDIAN.

Table 1. Functions That Produce a Binary String From a Number

Function	Description
<number>.SIGNED8_STRING	Produces an 8-bit signed binary string representing the number. If the value is out of range, an undef condition is raised. **Example** HTTP.REQ.BODY(100).GET_SIGNED8(16).SUB(3).SIGNED8_STRING
<number>.UNSIGNED8_STRING	Produces an 8-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. **Example** HTTP.REQ.BODY(100).GET_UNSIGNED8(31).ADD(3).UNSIGNED8_STRING
<number>.SIGNED16_STRING(<endianness>)	Produces a 16-bit signed binary string representing the number. If the value is out of range, an undef condition is raised. **Example** HTTP.REQ.BODY(100).SKIP(12).GET_SIGNED16(0, BIG_ENDIAN).SUB(4).SIGNED16_STRING(BIG_ENDIAN)
<number>.UNSIGNED16_STRING(<endianness>)	Produces a 16-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. **Example** HTTP.REQ.BODY(100).GET_UNSIGNED16(47, LITTLE_ENDIAN).ADD(7).UNSIGNED16_STRING(LITTLE_ENDIAN)
<number>.SIGNED32_STRING(<endianness>)	Produces a 32-bit signed binary string representing the number. **Example** HTTP.REQ.BODY(100).AFTER_STR("delim").GET_SIGNED32(0, BIG_ENDIAN).SUB(1).SIGNED32_STRING(BIG_ENDIAN)
<unsigned_long_number>.UNSIGNED8_STRING	Produces an 8-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. **Example** HTTP.REQ.BODY(100).GET_UNSIGNED8(24).TYPECAST_UNSIGNED_LONG_AT.ADD(12).UNSIGNED8_STRING
<unsigned_long_number>.UNSIGNED16_STRING(<endianness>)	Produces a 16-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. **Example** HTTP.REQ.BODY(100).GET_UNSIGNED16(23, LITTLE_ENDIAN).TYPECAST_UNSIGNED_LONG_AT.ADD(10).UNSIGNED16_STRING(LITTLE_ENDIAN)
<unsigned_long_number>.UNSIGNED32_STRING(<endianness>)	Produces a 32-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. **Example** HTTP.REQ.BODY(100).AFTER_STR("delim2").GET_UNSIGNED32(0, BIG_ENDIAN).ADD(2).UNSIGNED32_STRING(BIG_ENDIAN)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Virtual Server Based Expressions

Apr 17, 2012

The SYS.VSERVER("<vserver-name>") expression prefix enables you to identify a virtual server. You can use the following functions with this prefix to retrieve information related to the specified virtual server:

- **THROUGHPUT.** Returns the throughput of the virtual server in Mbps (Megabits per second). The value returned is an unsigned long number.
  Usage: SYS.VSERVER("vserver").THROUGHPUT

- **CONNECTIONS.** Returns the number of connections being managed by the virtual server. The value returned is an unsigned long number.
  Usage: SYS.VSERVER("vserver").CONNECTIONS

- **STATE.** Returns the state of the virtual server. The value returned is UP, DOWN, or OUT_OF_SERVICE. One of these values can therefore be passed as an argument to the EQ() operator to perform a comparison that results in a Boolean TRUE or FALSE.
  Usage: SYS.VSERVER("vserver").STATE

- **HEALTH.** Returns the percentage of services in an UP state for the specified virtual server. The value returned is an integer.
  Usage: SYS.VSERVER("vserver").HEALTH

- **RESPTIME.** Returns the response time as an integer representing the number of microseconds. Response time is the average TTFB (Time To First Byte) from all the services bound to the virtual server.
  Usage: SYS.VSERVER("vserver").RESPTIME

- **SURGECOUNT.** Returns the number of requests in the surge queue of the virtual server. The value returned is an integer.
  Usage: SYS.VSERVER("vserver").SURGECOUNT

Example 1

The following rewrite policy aborts rewrite processing if the number of connections at the load balancing virtual server LBvserver exceeds 10000:

```
add rewrite policy norewrite_pol sys.vserver("LBvserver").connections.gt(10000) norewrite
```

Example 2

The following rewrite action inserts a custom header, TP, whose value is the throughout at the virtual server LBvserver:

```
add rewrite action tp_header insert_http_header TP SYS.VSERVER("LBvserver").THROUGHPUT
```

Example 3

The following audit log message action writes the average TTFB of the services bound to a virtual server, to the newnslog log file:

```
add audit messageaction log_vserver_resptime_act INFORMATIONAL "\"NS Response Time to Servers:\" +
sys.vserver(\"ssllb\").resptime + \" millisec\"" -logtoNewnslog YES -bypassSafetyCheck YES
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Default Syntax Expressions: Parsing HTTP, TCP, and UDP Data

Sep 02, 2013

You can configure default syntax expressions to evaluate and process the payload in HTTP requests and responses. The payload associated with an HTTP connection includes the various HTTP headers (both standard and custom headers), the body, and other connection information such as the URL. Additionally, you can evaluate and process the payload in a TCP or UDP packet. For HTTP connections, for example, you can check whether a particular HTTP header is present or if the URL includes a particular query parameter.

You can configure expressions to transform the URL encoding and apply HTML or XML “safe” coding for subsequent evaluation. You can also use XPATH and JSON prefixes to evaluate date in XML and JSON files, respectively.

You can also use text-based and numeric default syntax expressions to evaluate HTTP request and response data. For more information, see "Default Syntax Expressions: Evaluating Text" and "Default Syntax Expressions: Working with Dates, Times, and Numbers."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

About Evaluating HTTP and TCP Payload

Sep 02, 2013

The payload of an HTTP request or response consists of HTTP protocol information such as headers, a URL, body content, and version and status information. When you configure a default syntax expression to evaluate HTTP payload, you use a default syntax expression prefix and, if necessary, an operator.

For example, you use the following expression, which includes the http.req.header(“<header_name>“) prefix and the exists operator, if you want to determine whether an HTTP connection includes a custom header named “myHeader”:

```
http.req.header("myHeader").exists
```

You can also combine multiple default syntax expressions with Boolean and arithmetic operators. For example, the following compound expression could be useful with various NetScaler features, such as Integrated Caching, Rewrite, and Responder. This expression first uses the && Boolean operator to determine whether an HTTP connection includes the Content-Type header with a value of "text/html." If that operation returns a value of FALSE, the expression determines whether the HTTP connection includes a "Transfer-Encoding" or "Content-Length" header.

```
(http.req.header("Content-Type").exists && http.req.header("Content-Type").eq("text/html")) || (http.req.header("Transfer-Encoding").exists) || (http.req.header("Content-Length").exists)
```

The payload of a TCP or UDP packet is the data portion of the packet. You can configure default syntax expressions to examine features of a TCP or UDP packet, including the following:

- Source and destination domains
- Source and destination ports
- The text in the payload
- Record types

The following expression prefixes extract text from the body of the payload:

- HTTP.REQ.BODY(integer). Returns the body of an HTTP request as a multiline text object, up to the character position designated in the integer argument. If there are fewer characters in the body than is specified in the argument, the entire body is returned.
- HTTP.RES.BODY(integer). Returns a portion of the HTTP response body. The length of the returned text is equal to the number in the integer argument. If there are fewer characters in the body than is specified in integer, the entire body is returned.
- CLIENT.TCP.PAYLOAD(integer). Returns TCP payload data as a string, starting with the first character in the payload and continuing for the number of characters in the integer argument.

Following is an example that evaluates to TRUE if a response body of 1024 bytes contains the string “https”, and this string occurs after the string “start string” and before the string “end string”:

```
http.res.body(1024).after_str("start_string").before_str("end_string").contains("https")
```

Note: You can apply any text operation to the payload body. For information on operations that you can apply to text, see "Default Syntax Expressions: Evaluating Text."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions for Identifying the Protocol in an Incoming IP Packet

May 21, 2015

The following table lists the expressions that you can use to identify the protocol in an incoming packet.

Expression	Description
CLIENT.IP.PROTOCOL	Identifies the protocol in IPv4 packets sent by clients.
CLIENT.IPV6.PROTOCOL	Identifies the protocol in IPv6 packets sent by clients.
SERVER.IP.PROTOCOL	Identifies the protocol in IPv4 packets sent by servers.
SERVER.IPV6.PROTOCOL	Identifies the protocol in IPv6 packets sent by servers.

Arguments to the PROTOCOL function

You can pass the Internet Assigned Numbers Authority (IANA) protocol number to the PROTOCOL function. For example, if you want to determine whether the protocol in an incoming packet is TCP, you can use CLIENT.IP.PROTOCOL.EQ(6), where 6 is the IANA-assigned protocol number for TCP. For some protocols, you can pass an enumeration value instead of the protocol number. For example, instead of CLIENT.IP.PROTOCOL.EQ(6), you can use CLIENT.IP.PROTOCOL.EQ(TCP). The following table lists the protocols for which you can use enumeration values, and the corresponding enumeration values for use with the PROTOCOL function.

Protocol	Enumeration value
Transmission Control Protocol (TCP)	TCP
User Datagram Protocol (UDP)	UDP
Internet Control Message Protocol (ICMP)	ICMP
IP Authentication Header (AH), for providing authentication services in IPv4 and IPv6	AH
Encapsulating Security Payload (ESP) protocol	ESP
General Routing Encapsulation (GRE)	GRE
IP-within-IP Encapsulation Protocol	IPIP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Protocol	Enumeration value
Internet Control Message Protocol for IPv6 (ICMPv6)	ICMPV6
Fragment Header for IPv6	FRAGMENT

Use Case Scenarios

The protocol expressions can be used in both request-based and response-based policies. You can use the expressions in various NetScaler features, such as load balancing, WAN optimization, content switching, rewrite, and listen policies. You can use the expressions with functions such as EQ() and NE(), to identify the protocol in a policy and perform an action.

Following are some use cases for the expressions:

- In Branch Repeater load balancing configurations, you can use the expressions in a listen policy for the wildcard virtual server. For example, you can configure the wildcard virtual server with the listen policy CLIENT.IP.PROTOCOL.EQ(TCP) so that the virtual server processes only TCP traffic and simply bridges all non-TCP traffic. Even though you can use an Access Control List instead of the listen policy, the listen policy provides better control over what traffic is processed.
- For content switching virtual servers of type ANY, you can configure content switching policies that switch requests on the basis of the protocol in incoming packets. For example, you can configure content switching policies to direct all TCP traffic to one load balancing virtual server and all non-TCP traffic to another load balancing virtual server.
- You can use the client-based expressions to configure persistence based on the protocol. For example, you can use CLIENT.IP.PROTOCOL to configure persistence on the basis of the protocols in incoming IPv4 packets.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions for HTTP and Cache-Control Headers

Sep 02, 2013

One common method of evaluating HTTP traffic is to examine the headers in a request or a response. A header can perform a number of functions, including the following:

- Provide cookies that contain data about the sender.
- Identify the type of data that is being transmitted.
- Identify the route that the data has traveled (the Via header).

Note: Note that if an operation is used to evaluate both header and text data, the header-based operation always overrides the text-based operation. For example, the AFTER_STR operation, when applied to a header, overrides text-based AFTER_STR operations for all instances of the current header type.

Prefixes for HTTP Headers

The following table describes expression prefixes that extract HTTP headers.

Table 1. Prefixes That Extract HTTP Headers

HTTP Header Prefix	Description
HTTP.REQ.HEADER("<header_name>")	Returns the contents of the HTTP header specified by the <header_name> argument. The header name cannot exceed 32 characters. Note that this prefix returns the value from the Host header by default. To use this value as a host name you need to typecast it as follows: http.req.header("host").typecast_http_hostname_t For more information on typecasting, see "Typecasting Data."
HTTP.REQ.FULL_HEADER	Returns the contents of the complete set of HTTP header fields including the request line (for example, "GET /brochures/index.html HTTP/1.1") and the terminating \r\n\r\n sequence.
HTTP.REQ.DATE	Returns the contents of the HTTP Date header. The following date formats are recognized: RFC822. Sun, 06 Jan 1980 8:49:37 GMT RFC850. Sunday, 06-Jan-80 9:49:37 GMT ASCII TIME. Sun Jan 6 08:49:37 1980 To evaluate a Date header as a date object, see "Default Syntax Expressions: Working with Dates, Times, and Numbers."
HTTP.REQ.COOKIE	(Name/Value List) Returns the contents of the HTTP Cookie header.
HTTP.REQ.TXID	Returns the HTTP transaction ID. The value is a function of an internal

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Header Prefix	Description
HTTP.RES.HEADER("<header_name>")	Returns the contents of the HTTP header specified by the <header_name> argument. The header name cannot exceed 32 characters.
HTTP.RES.FULL_HEADER	Returns the contents of the complete set of HTTP header fields including the status line (for example, "HTTP/1.1 200 OK") and the terminating \r\n\r\n sequence.
HTTP.RES.SET_COOKIE or HTTP.RES.SET_COOKIE2	Returns the HTTP Set-Cookie header object in a response.
HTTP.RES.SET_COOKIE("<name>") or HTTP.RES.SET_COOKIE2("<name>")	Returns the cookie of the specified name if it is present. If it is not present, returns a text object of length 0. Returns UNDEF if more than 15 Set-Cookie headers are present and the specified cookie was not found in these headers.
HTTP.RES.SET_COOKIE("<name>").DOMAIN or HTTP.RES.SET_COOKIE2("<name>").DOMAIN	Returns the value of the first Domain field in the cookie. For example, if the cookie is Set-Cookie : Customer = "ABC"; DOMAIN=".abc.com"; DOMAIN=.xyz.com, the following expression returns .abc.com: http.res.set_cookie.cookie("customer").domain A string of zero length is returned if the Domain field or its value is absent.
HTTP.RES.SET_COOKIE.EXISTS("<name>") or HTTP.RES.SET_COOKIE2.EXISTS("<name>")	Returns a Boolean TRUE if a Cookie with the name specified in the <name> argument exists in the Set-Cookie header. This prefix returns UNDEF if more than 15 Set-Cookie headers are present and the named cookie is not in the first 15 headers.
HTTP.RES.SET_COOKIE.COOKIE("<name>").EXPIRES or HTTP.RES.SET_COOKIE2.COOKIE("<name>").EXPIRES	Returns the Expires field of the cookie. This is a date string that can be evaluated as a number, as a time object, or as text. If multiple Expires fields are present, the first one is returned. If the Expires field is absent, a text object of length zero is returned. To evaluate the returned value as a time object, see "Default Syntax Expressions: Working with Dates, Times, and Numbers."
HTTP.RES.SET_COOKIE.COOKIE("<name>").PATH|PATH.GET(n) or HTTP.RES.SET_COOKIE2.COOKIE("<name>").PATH|PATH.GET(n)	Returns the value of Path field of the cookie as a slash- ("/") separated list. Multiple instances of a slash are treated as single slash. If multiple Path fields are present, the value of the first instance is returned. For example, the following is a cookie with two path fields: Set-Cookie : Customer = "ABC"; PATH="/a//b/c"; PATH= "/x/y/z"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Header Prefix	Description
	The following expression returns /a//b/c from this cookie: http.res.set_cookie.cookie("Customer").path The following expression returns b: http.res.set_cookie.cookie("Customer").path.get(2) Quotes are stripped from the returned value. A string of zero length is returned if the Path field or its value is absent.
HTTP.RES.SET_COOKIE.COOKIE("<name>").PATH.IGNORE_EMPTY_ELEMENTS or HTTP.RES.SET_COOKIE2.COOKIE("<name>").PATH.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list. For example, in the list a=10,b=11, ,c=89, the element delimiter in the list is , and the list has an empty element following a=10. The element following b=11 is not considered an empty element. As another example, in the following expression, if a request contains Cust_Header : 123,,24, ,15 the following expression returns a value of 4: http.req.header("Cust_Header").typecast_list_t(',').ignore_empty_elements.count The following expression returns a value of 5: http.req.header("Cust_Header").typecast_list_t(',').count
HTTP.RES.SET_COOKIE.COOKIE("<name>").PORT or HTTP.RES.SET_COOKIE2.COOKIE("<name>").PORT	Returns the value of Port field of the cookie. Operate as a comma-separated list. For example, the following expression returns 80. 2580 from Set-Cookie : Customer = "ABC"; PATH="/a/b/c"; PORT= "80, 2580": http.res.set_cookie.cookie("ABC").port A string of zero length is returned if the Port field or value is absent.
HTTP.RES.SET_COOKIE.COOKIE("<name>").PORT.IGNORE_EMPTY_ELEMENTS or HTTP.RES.SET_COOKIE2.COOKIE("<name>").PORT.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list. For example, in the list a=10,b=11, ,c=89, the element delimiter in the list is , and the list has an empty element following a=10. The element following b=11 is not considered an empty element. As another example, in the following expression, if a request contains Cust_Header : 123,,24, ,15 the following expression returns a value of 4: http.req.header("Cust_Header").typecast_list_t(',').ignore_empty_elements.count The following expression returns a value of 5: http.req.header("Cust_Header").typecast_list_t(',').count
HTTP.RES.SET_COOKIE.COOKIE("<name>").VERSION or HTTP.RES.SET_COOKIE2.COOKIE("<name>").VERSION	Returns the value of the first Version field in the cookie as a decimal integer. For example, the following expression returns 1 from the cookie Set-Cookie : Customer = "ABC"; VERSION = "1"; VERSION = "0" http.res.set_cookie.cookie("CUSTOMER").version A zero is returned if the Version field or its value is absent or if the value is not

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Header Prefix	Description
HTTP.RES.SET_COOKIE.COOKIE("<name>", <integer>) or HTTP.RES.SET_COOKIE2.COOKIE("<name>", <integer>)	Returns the nth instance (0-based) of the cookie with the specified name. If the cookie is absent, returns a text object of length 0. Returns UNDEF if more than 15 Set-Cookie headers are present and the cookie is not found.
HTTP.RES.SET_COOKIE.COOKIE("<name>", <integer>).DOMAIN or HTTP.RES.SET_COOKIE2.COOKIE("<name>", <integer>).DOMAIN	Returns the value of the Domain field of the first cookie with the specified name. For example, the following expression returns a value of abc.com from the cookie Set-Cookie : Customer = "ABC"; DOMAIN=".abc.com"; DOMAIN=.xyz.com http.res.set_cookie.cookie("CUSTOMER").domain A string of zero length is returned if the Domain field or its value is absent.
HTTP.RES.SET_COOKIE.COOKIE("<name>", <integer>).EXPIRES or HTTP.RES.SET_COOKIE2.COOKIE("<name>", <integer>).EXPIRES	Returns the *n*th instance (0-based) of the Expires field of the cookie with the specified name as a date string. The value can be operated upon as a time object that supports a number of date formats. If the Expires attribute is absent a string of length zero is returned.
HTTP.RES.SET_COOKIE.COOKIE("<name>", <integer>).PATH | PATH.GET(i) or HTTP.RES.SET_COOKIE2.COOKIE("<name>", <integer>).PATH | PATH.GET(i)	Returns the value of the Path field of the nth cookie, as a '/' separated list. Multiple /s are treated as a single /. For example, the following expression returns /a//b/c from the cookie Set-Cookie : Customer = "ABC"; PATH="/a//b/c"; PATH= "/x/y/z" http.res.set_cookie.cookie("CUSTOMER").path The following returns b: http.res.set_cookie.cookie("CUSTOMER").path.get(2) A string of zero length is returned if the Path field or its value is absent.
HTTP.RES.SET_COOKIE.COOKIE("<name>", <integer>).PATH.IGNORE_EMPTY_ELEMENTS or HTTP.RES.SET_COOKIE2.COOKIE("<name>", <integer>).PATH.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list. For example, in the list a=10,b=11, ,c=89, the element delimiter in the list is , and the list has an empty element following a=10. The element following b=11 is not considered an empty element. As another example, in the following expression, if a request contains Cust_Header : 123,,24, ,15 the following expression returns a value of 4: http.req.header("Cust_Header").typecast_list_t(',').ignore_empty_elements.count The following expression returns a value of 5: http.req.header("Cust_Header").typecast_list_t(',').count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Header Prefix	Description
HTTP.RES.SET_COOKIE.COOKIE("<name>", <integer>).PORT or HTTP.RES.SET_COOKIE2.COOKIE("<name>", <integer>).PORT	Returns the value or values of the Port field of the named cookie as a ',' separated list. For example, the following expression returns 80, 2580 from the cookie Set-Cookie : Customer = "ABC"; PATH="/a/b/c"; PORT= "80, 2580" http.res.set_cookie.cookie("ABC").port A string of zero length is returned if the Port field or its value is absent.
HTTP.RES.SET_COOKIE.COOKIE("<name>", <integer>).PORT.IGNORE_EMPTY_ELEMENTS or HTTP.RES.SET_COOKIE2.COOKIE("<name>", <integer>).PORT.IGNORE_EMPTY_ELEMENTS	Ignores the empty elements in the list. For example, in the list a=10,b=11, ,c=89, the element delimiter in the list is , and the list has an empty element following a=10. The element following b=11 is not considered an empty element. As another example, in the following expression, if a request contains Cust_Header : 123,,24, ,15 the following expression returns a value of 4: http.req.header("Cust_Header").typecast_list_t(',').ignore_empty_elements.count The following expression returns a value of 5: http.req.header("Cust_Header").typecast_list_t(',').count
HTTP.RES.SET_COOKIE.COOKIE("<name>", <integer>).VERSION or HTTP.RES.SET_COOKIE2.COOKIE("<name>", <integer>).VERSION	Returns the value of Version field of the *n*th cookie as a decimal integer. A string of zero length is returned if the Port field or its value is absent.
HTTP.RES.TXID	Returns the HTTP transaction ID. The value is a function of an internal transaction number, system boot time and system MAC address.

Operations for HTTP Headers

The following table describes operations that you can specify with the prefixes for HTTP headers.

Table 2. Operations That Evaluate HTTP Headers

HTTP Header Operation	Description
http header .EXISTS	Returns a Boolean TRUE if an instance of the specified header type exists. Following is an example: http.req.header("Cache-Control").exists
http header.CONTAINS" http header . CONTAINS(" <string>")	Returns a Boolean TRUE if the <string> argument appears in any instance of the header value. Note: This operation overrides any text-based Contains operations on all instances of the current header type. Following is an example of request with two headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	HTTP/1.1 200 OK\r\n MyHeader: abc\r\n Content-Length: 200\r\n MyHeader: def\r\n \r\n The following returns a Boolean TRUE: http.res.header("MyHeader"). contains("de") The following returns FALSE. Note that the NetScaler does not concatenate the different values. http.res.header("MyHeader"). contains("bcd")
http header .COUNT	Returns the number of headers in a request or response, to a maximum of 15 headers of the same type. The result is undefined if there are more than 15 instances of the header. Following is sample data in a request: HTTP/1.1 200 OK\r\n MyHeader: abc\r\n Content-Length: 200\r\n MyHeader: def\r\n \r\n When evaluating the preceding request, the following returns a count of 2: http.res.header("MyHeader").count
http header.AFTER_STR("<string>")	Extracts the text that follows the first occurrence of the <string> argument. The headers are evaluated from the last instance to the first. Following is an example of a request: HTTP/1.1 200 OK\r\n MyHeader: 111abc\r\n Content-Length: 200\r\n MyHeader: 111def\r\n \r\n The following extracts the string "def" from the last instance of MyHeader. This is value "111def." http.res.header("MyHeader").after_str("111") The following extracts the string "c" from the first instance of MyHeader. This is the value "abc111."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	http.res.header("MyHeader").after_str("1ab")
http header.BEFORE_STR("<string>")	Extracts the text that appears prior to the first occurrence of the input <string> argument. The headers are evaluated from the last instance to the first. Following is an example of a request that contains headers: HTTP/1.1 200 OK\r\n MyHeader: abc111\r\n Content-Length: 200\r\n MyHeader: def111\r\n \r\n The following extracts the string "def" from the last instance of MyHeader. This is the value "def111." http.res.header("MyHeader").before_str("111") The following extracts the string "a" from the first instance of MyHeader. This is the value "abc111." http.res.header("MyHeader").before_str("bc1")
http header.INSTANCE(<instance number>)	An HTTP header can occur multiple times in a request or a response. This operation returns the header that occurs <instance number> of places before the final instance. For example, instance(0) selects the last instance of the current type, instance(1) selects the next-to-last instance, and so on. This prefix cannot be used in bidirectional policies. The <instance number> argument cannot exceed 14. Following is an example of a request with two headers: HTTP/1.1 200 OK\r\n MyHeader: abc\r\n Content-Length: 200\r\n MyHeader: def\r\n \r\n The following returns a text object that refers to "MyHeader: abc\r\n": http.res.header("MyHeader").instance(1)
http header.SUBSTR("<string>")	Extracts the text that matches the <string> argument. The headers are evaluated from the last instance to the first. Following is an example of a request with two headers that contain the string “111”: HTTP/1.1 200 OK\r\n MyHeader: abc111\r\n

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	Content-Length: 200\r\n MyHeader: 111def\r\n \r\n The following returns "111" from the last instance of MyHeader. This is the header with the value "111def." http.res.header("MyHeader").substr("111")
http header.VALUE(<instance number>)	An HTTP header can occur multiple times in a request or a response. VALUE(0) selects the value in the last instance, VALUE(1) selects the value in the next-to-last instance, and so on. The <instance number> argument cannot exceed 14. Following is an example of a request with two headers: HTTP/1.1 200 OK\r\n MyHeader: abc\r\n Content-Length: 200\r\n MyHeader: def\r\n \r\n The following returns "abc\r\n": http.res.header("MyHeader").value(1)

Prefixes for Cache-Control Headers

The following prefixes apply specifically to Cache-Control headers.

Table 3. Prefixes That Extract Cache-Control Headers

HTTP Header Prefix	Description
HTTP.REQ.CACHE_CONTROL	Returns a Cache-Control header in an HTTP request.
HTTP.RES.CACHE_CONTROL	Returns a Cache-Control header in an HTTP response.

Operations for Cache-Control Headers

You can apply any of the operations for HTTP headers to Cache-Control headers. For more information, see "Operations for HTTP Headers."

In addition, the following operations identify specific types of Cache-Control headers. See RFC 2616 for information about these header types.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 4. Operations That Evaluate Cache-Control Headers

HTTP Header Operation	Description
Cache-Control header.NAME(<integer>)	Returns as a text value the name of the Cache-Control header that corresponds to the nth component in a name-value list, as specified by <integer>. The index of the name-value component is 0-based. If the <integer> that is specified by the integer argument is greater than the number of components in the list, a zero-length text object is returned. Following is an example: http.req.cache_control.name(3).contains("some_text")
Cache-Control header.IS_INVALID	Returns a Boolean TRUE if the Cache-Control header is not present in the request or response. Following is an example: http.req.cache_control.is_invalid
Cache-Control header.IS_PRIVATE	Returns a Boolean TRUE if the Cache-Control header has the value Private. Following is an example: http.req.cache_control.is_private
Cache-Control header.IS_PUBLIC	Returns a Boolean TRUE if the Cache-Control header has the value Private. Following is an example: http.req.cache_control.is_public
Cache-Control header.IS_NO_STORE	Returns a Boolean TRUE if the Cache-Control header has the value No-Store. Following is an example: http.req.cache_control.is_no_store
Cache-Control header.IS_NO_CACHE	Returns a Boolean TRUE if the Cache-Control header has the value No-Cache. Following is an example: http.req.cache_control.is_no_cache
Cache-Control header.IS_MAX_AGE	Returns a Boolean TRUE if the Cache-Control header has the value Max-Age. Following is an example: http.req.cache_control.is_max_age

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Header Operation	Description
Cache-Control header.IS_MIN_FRESH	Returns a Boolean TRUE if the Cache-Control header has the value Min-Fresh. Following is an example: http.req.cache_control.is_min_fresh
Cache-Control header.IS_MAX_STALE	Returns a Boolean TRUE if the Cache-Control header has the value Max-Stale. Following is an example: http.req.cache_control.is_max_stale
Cache-Control header.IS_MUST_REVALIDATE	Returns a Boolean TRUE if the Cache-Control header has the value Must-Revalidate. Following is an example: http.req.cache_control.is_must_revalidate
Cache-Control header.IS_NO_TRANSFORM	Returns a Boolean TRUE if the Cache-Control header has the value No-Transform. Following is an example: http.req.cache_control.is_no_transform
Cache-Control header.IS_ONLY_IF_CACHED	Returns a Boolean TRUE if the Cache-Control header has the value Only-If-Cached. Following is an example: http.req.cache_control.is_only_if_cached
Cache-Control header.IS_PROXY_REVALIDATE	Returns a Boolean TRUE if the Cache-Control header has the value Proxy-Revalidate. Following is an example: http.req.cache_control.is_proxy_revalidate
Cache-Control header.IS_S_MAXAGE	Returns a Boolean TRUE if the Cache-Control header has the value S-Maxage. Following is an example: http.req.cache_control.is_s_maxage
Cache-Control header.IS_UNKNOWN	Returns a Boolean TRUE if the Cache-Control header is of an unknown type. Following is an example: http.req.cache_control.is_unknown
Cache-Control header.MAX_AGE	Returns the value of the Cache-Control header Max-Age. If this header is absent or invalid, 0 is returned. Following is an example:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Header Operation	Description
Cache-Control header.MAX_STALE	Returns the value of the Cache-Control header Max-Stale. If this header is absent or invalid, 0 is returned. Following is an example: http.req.cache_control.max_stale.le(3)
Cache-Control header.MIN_FRESH	Returns the value of the Cache-Control header Min-Fresh. If this header is absent or invalid, 0 is returned. Following is an example: http.req.cache_control.min_fresh.le(3)
Cache-Control header.S_MAXAGE	Returns the value of the Cache-Control header S-Maxage. If this header is absent or invalid, 0 is returned. Following is an example: http.req.cache_control.s_maxage.eq(2)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions for Extracting Segments of URLs

Sep 02, 2013

You can extract URLs and portions of URLs, such as the host name, or a segment of the URL path. For example, the following expression identifies HTTP requests for image files by extracting image file suffixes from the URL:

http.req.url.suffix.eq("jpeg") || http.req.url.suffix.eq("gif")

Most expressions for URLs operate on text and are described in "Expression Prefixes for Text in HTTP Requests and Responses." This section discusses the GET operation. The GET operation extracts text when used with the following prefixes:

- HTTP.REQ.URL.PATH
- VPN.BASEURL.PATH
- VPN.CLIENTLESS_BASEURL.PATH

The following table describes prefixes for HTTP URLs.

Table 1. Prefixes That Extract URLs

URL Prefix	Description
HTTP.REQ.URL.PATH.GET(<n>)	Returns a slash- ("/") separated list from the URL path. For example, consider the following URL: http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1 The following expression returns dir1 from this URL: http.req.url.path.get(1) The following expression returns dir2: http.req.url.path.get(2)
HTTP.REQ.URL.PATH.GET_REVERSE(<n>)	Returns a slash- ("/") separated list from the URL path, starting from the end of the path. For example, consider the following URL: http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1 The following expression returns index.html from this URL: http.req.url.path.get_reverse(0) The following expression returns dir3: http.req.url.path.get_reverse(1)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions for HTTP Status Codes and Numeric HTTP Payload Data Other Than Dates

Mar 20, 2012

The following table describes prefixes for numeric values in HTTP data other than dates.

Table 1. Prefixes That Evaluate HTTP Request or Response Length

Prefix	Description
HTTP.REQ.CONTENT_LENGTH	Returns the length of an HTTP request as a number. Following is an example: http.req.content_length < 500
HTTP.RES.CONTENT_LENGTH	Returns the length of the HTTP response as a number. Following is an example: http.res.content_length <= 1000
HTTP.RES.STATUS	Returns the response status code
HTTP.RES.IS_REDIRECT	Returns a Boolean TRUE if the response code is associated with a redirect. Following are the redirect response codes: • 300 (Multiple Choices) • 301 (Moved Permanently) • 302 (Found) • 303 (See Other) • 305 (Use Proxy) • 307 (Temporary Redirect) Note: Status code 304 is not considered a redirect HTTP response status code. Status code 306 is unused. In the following example, the rewrite action replaces http in the Location header of an HTTP response with https if the response is associated with an HTTP redirect. add rewrite action redloc replace 'http.res.header("Location").before_regex(re#://#)' '"https"' add rewrite policy pol1 HTTP.RES.IS_REDIRECT red_location bind rewrite global pol1 100

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SIP Expressions

Sep 23, 2014

Introduction

The NetScaler default expressions language contains a number of expressions that operate on Session Initiation Protocol (SIP) connections. These expressions are intended to be used in policies for any supported protocol that operates on a request/response basis. These expressions can be used in content switching, rate limiting, responder, and rewrite policies.

Certain limitations apply to SIP expressions used with responder policies. Only the DROP, NOOP or RESPONDWITH actions are allowed on a SIP load balancing virtual server. Responder policies can be bound to a load balancing virtual server, an override global bind point, a default global bind point, or a sip_udp policy label.

The header format used by the SIP protocol is similar to that used by the HTTP protocol, so many of the new expressions look and function much like their HTTP analogs. Each SIP header consists of a line that includes the SIP method, the URL, and the version, followed by a series of name-value pairs that look like HTTP headers.

Following is a sample SIP header that is referred to in the expressions tables beneath it:

```
INVITE sip:16@www.sip.com:5060;transport=udp SIP/2.0
Record-Route: <sip:200.200.100.22;lr=on>
Via: SIP/2.0/UDP 200.200.100.22;branch=z9hG4bK444b.c8e103d1.0;rport=5060;
 received=10.102.84.18
Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;
 received=10.102.84.160
From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185
To: "16" <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185
Call-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180
Max-Forwards: 69CSeq: 101 INVITE
User-Agent: Cisco-CP7940G/8.0
Contact: <sip:12@10.102.84.180:5060;transport=udp>
Expires: 180
Accept: application/sdp
Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,UPDATE
Supported: replaces,join,norefersub
Content-Length: 277
Content-Type: application/sdp
Content-Disposition: session;handling=optiona
```

SIP Reference Tables

The following tables contain lists of expressions that operate on SIP headers. The first table contains expressions that apply to request headers. Most response-based expressions are nearly the same as the corresponding request-based expressions. To create a response expression from the corresponding request expression, you change the first two sections of the expression from SIP.REQ to SIP.RES, and make other obvious adjustments. The second table contains those response expressions that are unique to responses and have no request equivalents. You can use any element in the following tables as a complete expression on its own, or you can use various operators to combine these expression elements with others to form more complex expressions.

Table 1. SIP Request Expressions

Expression	Description
SIP.REQ.METHOD	Operates on the method of the SIP request. The supported SIP request methods are ACK, BYE, CANCEL, INFO, INVITE, MESSAGE, NOTIFY, OPTIONS, PRACK, PUBLISH, REFER, REGISTER, SUBSCRIBE, and UPDATE. This expression is a derivative of the text class, so all operations that are applicable to text are applicable to this

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Description
	method. For example, for a SIP request of INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0, this expression returns INVITE.
SIP.REQ.URL	Operates on the SIP request URL. This expression is a derivative of the text class, so all operations that are applicable to text are applicable to this method. For example, for a SIP request of INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0, this expression returns sip:16@10.102.84.181:5060;transport=udp.
SIP.REQ.URL.PROTOCOL	Returns the URL protocol. For example, for a SIP URL of sip:16@www.sip.com:5060;transport=udp, this expression returns sip.
SIP.REQ.URL.HOSTNAME	Returns the hostname portion of the SIP URL. For example, for a SIP URL of sip:16@www.sip.com:5060;transport=udp, this expression returns www.sip.com:5060.
SIP.REQ.URL.HOSTNAME.PORT	Returns the port portion of the SIP URL hostname. If no port is specified, this expression returns the default SIP port, 5060. For example, for a SIP hostname of www.sip.com:5060, this expression returns 5060.
SIP.REQ.URL.HOSTNAME.DOMAIN	Returns the domain name portion of the SIP URL hostname. If the host is an IP address, then this expression returns an incorrect result. For example, for a SIP hostname of www.sip.com:5060, this expression returns sip.com. For a SIP hostname of 192.168.43.15:5060, this expression returns an error.
SIP.REQ.URL.HOSTNAME.SERVER	Returns the server portion of the host. For example, for a SIP hostname of www.sip.com:5060, this expression returns www.
SIP.REQ.URL.USERNAME	Returns the username that precedes the @ character. For example, for a SIP URL of sip:16@www.sip.com:5060;transport=udp, this expression returns 16.
SIP.REQ.VERSION	Returns the SIP version number in the request. For example, for a SIP request of INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0, this expression returns SIP/2.0.
SIP.REQ.VERSION.MAJOR	Returns the major version number (the number to the left of the period). For example, for a SIP version number of SIP/2.0, this expression returns 2.
SIP.REQ.VERSION.MINOR	Returns the minor version number (the number to the right of the period). For example, for a SIP version number of SIP/2.0, this expression returns 0.
SIP.REQ.CONTENT_LENGTH	Returns the contents of the Content-Length header. This expression is a derivative of the sip_header_t class, so all operations that are available for SIP headers can be used. For example, for a SIP Content-Length header of Content-Length: 277, this expression returns 277.
SIP.REQ.TO	Returns the contents of the To header. For example, for a SIP To header of To: "16" <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns "16" <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Description
SIP.REQ.TO.ADDRESS	Returns the SIP URI, which is found in the sip_url object. All operations that are available for SIP URIs can be used. For example, for a SIP To header of To: "16" <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns sip:16@sip_example.com.
SIP.REQ.TO.DISPLAY_NAME	Returns the display name portion of the To header. For example, for a SIP To header of To: "16" <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 16.
SIP.REQ.TO.TAG	Returns the “tag” vale from the “tag” name value pair in the TO header. For example, for a SIP To header of To: "16" <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 00127f54ec85a6d90cc14f45-53cc0185.
SIP.REQ.FROM	Returns the contents of the From header. For example, for a SIP From header of From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns sip:12@sip_example.com.
SIP.REQ.FROM.ADDRESS	Returns the SIP URI, which is found in the sip_url object. All operations that are available for SIP URIs can be used. For example, for a SIP From header of From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns sip:12@sip_example.com.
SIP.REQ.FROM.DISPLAY_NAME	Returns the display name portion of the To header. For example, for a SIP From header of From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 12.
SIP.REQ.FROM.TAG	Returns the “tag” value from the “tag” name/value pair in the TO header. For example, for a SIP From header of From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 00127f54ec85a6d90cc14f45-53cc0185.
SIP.REQ.VIA	Returns the complete Via header. If there are multiple Via headers in the request, returns the last Via header. For example, for the two Via headers in the sample SIP header, this expression returns Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160.
SIP.REQ.VIA.SENTBY_ADDRESS	Returns the address that sent the request. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160, this expression returns 10.102.84.180.
SIP.REQ.VIA.SENTBY_PORT	Returns the port that sent the request. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160, this expression returns 5060.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Description
SIP.REQ.VIA.RPORT	Returns the value from the rport name/value pair. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160, this expression returns 5060.
SIP.REQ.VIA.BRANCH	Returns the value from the branch name/value pair. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160, this expression returns z9hG4bK03e76d0b.
SIP.REQ.VIA.RECEIVED	Returns the value from the received name/value pair. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160, this expression returns 10.102.84.160.
SIP.REQ.CALLID	Returns the contents of the Callid header. This expression is a derivative of the sip_header_t class, so all operations that are available for SIP headers can be used. For example, for a SIP Callid header of Call-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180, this expression returns 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180.
SIP.REQ.CSEQ	Returns the CSEQ number from the CSEQ, as an integer. For example, for a SIP CSEQ header of CSeq: 101 INVITE, this expression returns 101.
SIP.REQ.HEADER("<header_name>")	Returns the specified SIP header. For <header_name>, substitute the name of the header that you want. For example, to return the SIP From header, you would type SIP.REQ.HEADER("From").
SIP.REQ.HEADER("<header_name>").INSTANCE(<line_number>)	Returns the specified instance of the specified SIP header. Multiple instances of the same SIP header can occur. Where you want a specific instance of such a SIP header (for example, a specific Via header), you can specify that header by typing a number as the <line_number>. Header instances are matched from last (0) to first. In other words, SIP.REQ.HEADER("Via").INSTANCE(0) returns the last instance of the Via header, while SIP.REQ.HEADER("Via").INSTANCE(1) returns the last instance but one of the Via header, and so on. For example, if used on the example SIP header, SIP.REQ.HEADER("Via").INSTANCE(1) returns Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060.
SIP.REQ.HEADER("<header_name>").VALUE(<line_number>)	Returns the contents of the specified instance of the specified SIP header. The usage is nearly the same as the previous expression. For example, if used on the SIP header example in the preceding table entry, SIP.REQ.HEADER("Via").VALUE(1) returns SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060.
SIP.REQ.HEADER("<header_name>").COUNT	Returns the number of instances of a particular header as an integer. For example, if used on the SIP header example above, SIP.REQ.HEADER("Via").COUNT returns 2.
SIP.REQ.HEADER("<header_name>").EXISTS	Returns a boolean value of true or false, depending upon whether the specified header exists or not. For example, if used on the SIP header example above,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Description
	SIP.REQ.HEADER(“Expires”).EXISTS returns true, while SIP.REQ.HEADER(“Caller-ID”).EXISTS returns false.
SIP.REQ.HEADER("<header_name>").LIST	Returns the comma-separated parameter list in the specified header. For example, if used on the SIP header example above, SIP.REQ.HEADER(“Allow”).LIST returns ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,UPDATE. You can append the string .GET(<list_item_number>) to select a specific list item. For example, to get the first item (ACK) from the above list, you would type SIP.REQ.HEADER(“Allow”).LIST.GET(0). To extract the second item (BYE), you would type SIP.REQ.HEADER(“Allow”).LIST.GET(1). Note: If the specified header contains a list of name/value pairs, the entire name/value pair is returned.
SIP.REQ.HEADER("<header_name>").TYPECAST_SIP_HEADER_T("<in_header_name>")	Typecasts <header_name> to <in_header_name>. Any text can be typecasted to the sip_header_t class, after which all header-based operations can be used. After you perform this operation, you can apply all operations that can be used with <in_header_name>. For example, the expression SIP.REQ.CONTENT_LENGTH.TYPECAST_SIP_HEADER_T typecasts all instances of the Content-Length header. After you perform this operation, you can apply all header operations to all instances of the specified header.
SIP.REQ.HEADER("<header_name>").CONTAINS("<string>").	Returns boolean true if the specified text string is present in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER("<header_name>").EQUALS_ANY(<patset>)	Returns boolean true if any pattern associated with <patset> matches any content in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER("<header_name>").CONTAINS_ANY(<patset>)	Returns Boolean true if any pattern associated with <patset> matches any content in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER("<header_name>").CONTAINS_INDEX(<patset>)	Returns the index of the matching pattern associated with <patset> if that pattern matches any content in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER("<header_name>").EQUALS_INDEX(<patset>)	Returns the index of the matching pattern associated with <patset> if that pattern matches any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER("<header_name>").SUBSTR("<string>")	If the specified string is present in any instance of the specified header, this expression returns that string. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160”, SIP.REQ.HEADER("Via").SUBSTR("rport=5060") returns “rport=5060”. SIP.REQ.HEADER("Via").SUBSTR("rport=5061") returns an empty string.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Description
SIP.REQ.HEADER("<header_name>").AFTER_STR("<string>")	If the specified string is present in any instance of the specified header, this expression returns the string immediately after that string. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160, the expression SIP.REQ.HEADER("Via").AFTER_STR("rport=") returns 5060.
SIP.REQ.HEADER("<header_name>").REGEX_MATCH(<regex>)	Returns boolean true if the specified regular expression (*regex*) matches any instance of the specified header. You must specify the regular expression in the following format: re<delimiter>regular expression<same delimiter> The regular expression cannot be larger than 1499 characters in length. It must conform to the PCRE regular expression library. See http://www.pcre.org/pcre.txt for documentation on PCRE regular expression syntax. The pcrepattern man page also has useful information on specifying patterns by using PCRE regular expressions. The regular expression syntax supported in this expression has some differences from PCRE. Back references are not allowed. You should avoid recursive regular expressions; although some work, many do not. The dot (.) metacharacter matches newlines. Unicode is not supported. SET_TEXT_MODE(IGNORECASE) overrides the (?i) internal option specified in the regular expression.
SIP.REQ.HEADER("<header_name>").REGEX_SELECT(<regex>)	If the specified regex matches any text in any instance of the specified header, this expression returns the text. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160, the expression SIP.REQ.HEADER("Via").REGEX_SELECT("received=[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}") returns received=10.102.84.160.
SIP.REQ.HEADER("<header_name>").AFTER_REGEX(<regex>)	If the specified regex matches any text in any instance of the specified header, this expression returns the string immediately after that text. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160, the expression SIP.REQ.HEADER("Via").AFTER_REGEX("received=") returns 10.102.84.160.
SIP.REQ.HEADER("<header_name>").BEFORE_REGEX(<regex>)	If the specified regex matches any text in any instance of the specified header, this expression returns the string immediately before that text. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;received=10.102.84.160, the expression SIP.REQ.HEADER("Via").BEFORE_REGEX("[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}") returns received=.
SIP.REQ.FULL_HEADER	Returns the entire SIP header, including the terminating CR/LF.
SIP.REQ.IS_VALID	Returns boolean true if the request format is valid.
SIP.REQ.BODY(<length>)	Returns the request body, up to the specified length. If the specified length is greater than the length of the request body, this expression returns the entire request body.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Description
SIP.REQ.LB_VSERVER	Returns the name of the load balancing virtual server (*LB vserver*) that is serving the current request.
SIP.REQ.CS_VSERVER	Returns the name of the content switching virtual server (*CS vserver*) that is serving the current request.

Table 2. SIP Response Expressions

Expression	Description
SIP.RES.STATUS	Returns the SIP response status code. For example, if the first line of the response is SIP/2.0 100 Trying, this expression returns 100.
SIP.RES.STATUS_MSG	Returns the SIP response status message. For example, if the first line of the response is SIP/2.0 100 Trying, this expression returns Trying.
SIP.RES.IS_REDIRECT	Returns boolean true if the response code is a redirect.
SIP.RES.METHOD	Returns the response method extracted from the request method string in the CSeq header.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations for HTTP, HTML, and XML Encoding and “Safe” Characters

Nov 27, 2014

The following operations work with the encoding of HTML data in a request or response and XML data in a POST body.

Table 1. Operations That Evaluate HTML and XML Encoding

HTML or XML Operation	Description
<text>.HTML_XML_SAFE	Transforms special characters into XML safe format, as in the following examples: ● A left-pointing angle bracket (<) is converted to < ● A right-pointing angle bracket (>) is converted to > ● An ampersand (&) is converted to & This operation safeguards against cross-site scripting attacks. Maximum length of the transformed text is 2048 bytes. This is a read-only operation. After applying the transformation, additional operators that you specify in the expression are applied to the selected text. Following is an example: http.req.url.query.html_xml_safe.contains("myQueryString")
<text>.HTTP_HEADER_SAFE	Converts all new line ('\n') characters in the input text to '%0A' to enable the input to be used safely in HTTP headers. This operation safeguards against response-splitting attacks. The maximum length of the transformed text is 2048 bytes. This is a read-only operation.
<text>.HTTP_URL_SAFE	Converts unsafe URL characters to '%xx' values, where “xx” is a hex-based representation of the input character. For example, the ampersand (&) is represented as %26 in URL-safe encoding. The maximum length of the transformed text is 2048 bytes. This is a read-only operation. Following are URL safe characters. All others are unsafe: ● Alpha-numeric characters: a-z, A-Z, 0-9 ● Asterix: "*" ● Ampersand: "&" ● At-sign: "@" ● Colon: ":" ● Comma: "," ● Dollar: "$" ● Dot: "." ● Equals: "=" ● Exclamation mark: "!" ● Hyphen: "-" ● Open and close parentheses: "(", ")" ● Percent: "%" ● Plus: "+" ● Semicolon: ";"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTML or XML Operation	Description
	• Single quote: "'" • Slash: "/" • Question mark: "?" • Tilde: "~" • Underscore: "_"
<text>.MARK_SAFE	Marks the text as safe without applying any type of data transformation.
<text>.SET_TEXT_MODE(URLENCODED|NOURLENCODED)	Transforms all %HH encoding in the byte stream. This operation works with characters (not bytes). By default, a single byte represents a character in ASCII encoding. However, if you specify URLENCODED mode, three bytes can represent a character. In the following example, a PREFIX(3) operation selects the first 3 characters in a target. http.req.url.hostname.prefix(3) In the following example, the NetScaler can select up to 9 bytes from the target: http.req.url.hostname.set_text_mode(urlencoded).prefix(3)
<text>.SET_TEXT_MODE(PLUS_AS_SPACE|NO_PLUS_AS_SPACE)	Specifies how to treat the plus character (+). The PLUS_AS_SPACE option replaces a plus character with white space. For example, the text “hello+world” becomes “hello world.” The NO_PLUS_AS_SPACE option leaves plus characters as they are.
<text>.SET_TEXT_MODE(BACKSLASH_ENCODED|NO_BACKSLASH_ENCODED)	Specifies whether or not backslash decoding is performed on the text object represented by <text>. If BACKSLASH_ENCODED is specified, the SET_TEXT_MODE operator performs the following operations on the text object: • All occurrences of “\XXX” will be replaced with the character “Y” (where XXX represents a number in the octal system and Y represents the ASCII equivalent of XXX). The valid range of octal values for this type of encoding is 0 to 377. For example, the encoded text "http\72//" and "http\072//" will both be decoded to "http://", where the colon (:) is the ASCII equivalent of the octal value “72”. • All occurrences of “\xHH” will be replaced with the character “Y” (HH represents a number in the hexadecimal system and Y denotes the ASCII equivalent of HH. For example, the encoded text "http\x3a//" will be decoded to "http://", where the colon (:) is the ASCII equivalent of the hexadecimal value “3a“. • All occurrences of “\\uWWXX” will be replaced with the character sequence “YZ” (Where WW and XX represent two distinct hexadecimal values and Y and Z represent their ASCII equivalents of WW and XX respectively. For example, the encoded text "http%u3a2f/" and "http%u003a//" will both be decoded to "http://", where “3a” and “2f” are two hexadecimal values and the colon (:) and forward slash (“/”) represent their ASCII equivalents respectively. • All occurrences of "\b", "\n", "\t", "\f", and "\r" are

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTML or XML Operation	Description
	replaced with the corresponding ASCII characters. If NO_BACKSLASH_ENCODED is specified, backslash decoding is not performed on the text object.
<text>.SET_TEXT_MODE(BAD_ENCODE_RAISE_UNDEF|NO_BAD_ENCODE_RAISE_UNDEF)	Performs the associated undefined action if either the URLENCODED or the BACKSLASH_ENCODED mode is set and bad encoding corresponding to the specified encoding mode is encountered in the text object represented by <text>. If NO_BAD_ENCODE_RAISE_UNDEF is specified, the associated undefined action will not be performed when bad encoding is encountered in the text object represented by<text>.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions for TCP, UDP, and VLAN Data

Sep 02, 2013

TCP and UDP data take the form of a string or a number. For expression prefixes that return string values for TCP and UDP data, you can apply any text-based operations. For more information, see "Default Syntax Expressions: Evaluating Text."

For expression prefixes that return a numeric value, such as a source port, you can apply an arithmetic operation. For more information, see "Basic Operations on Expression Prefixes" and "Compound Operations for Numbers."

The following table describes prefixes that extract TCP and UDP data.

Table 1. Prefixes That Extract TCP and UDP Data

GET Operation	Description
CLIENT.TCP.PAYLOAD(<integer>)	Returns TCP payload data as a string, starting with the first character in the payload and continuing for the number of characters in the <integer> argument. You can apply any text-based operation to this prefix.
CLIENT.TCP.SRCPORT	Returns the ID of the current packet's source port as a number.
CLIENT.TCP.DSTPORT	Returns the ID of the current packet's destination port as a number.
CLIENT.TCP.OPTIONS	Returns the TCP options set by the client. Examples of TCP options are Maximum Segment Size (MSS), Window Scale, Selective Acknowledgements (SACK), and Time Stamp Option. The COUNT, TYPE(<type>), and TYPE_NAME(<m>) operators can be used with this prefix. For the TCP options set by the server, see the SERVER.TCP.OPTIONS prefix.
CLIENT.TCP.OPTIONS.COUNT	Returns the number of TCP options that the client has set.
CLIENT.TCP.OPTIONS.TYPE(<type>)	Returns the value of the TCP option whose type (or *option kind*) is specified as the argument. The value is returned as a string of bytes in big endian format (or *network byte order*). **Parameters:** type - Type value
CLIENT.TCP.OPTIONS.TYPE_NAME(<m>)	Returns the value of the TCP option whose enumeration constant is specified as the argument. The enumeration constants that you can

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GET Operation	Description
	pass as the argument are REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG. To specify the TCP option kind instead of these enumeration constants, use CLIENT.TCP.OPTIONS.TYPE(<type>). For other TCP options, you must use CLIENT.TCP.OPTIONS.TYPE(<type>). **Parameters:** m - TCP option enumeration constant
CLIENT.TCP.REPEATER_OPTION.EXISTS	Returns a Boolean TRUE if Repeater TCP options exist.
CLIENT.TCP.REPEATER_OPTION.IP	Returns the branch repeater's IPv4 address from the Repeater TCP options.
CLIENT.TCP.REPEATER_OPTION.MAC	Returns the branch repeater's MAC address from the Repeater TCP options.
CLIENT.UDP.DNS.DOMAIN	Returns the DNS domain name.
CLIENT.UDP.DNS.DOMAIN.EQ("<hostname>")	Returns a Boolean TRUE if the domain name matches the <hostname> argument. The comparison is case insensitive. Following is an example: client.udp.dns.domain.eq("www.mycompany.com")
CLIENT.UDP.DNS.IS_AAAAREC	Returns a Boolean TRUE if the record type is AAAA. These types of records indicate an IPv6 address in forward lookups.
CLIENT.UDP.DNS.IS_ANYREC	Returns a Boolean TRUE if it is of any record type.
CLIENT.UDP.DNS.IS_AREC	Returns a Boolean TRUE if the record is type A. Type A records provide the host address.
CLIENT.UDP.DNS.IS_CNAMEREC	Returns a Boolean TRUE if the record is of type CNAME. In systems that use multiple names to identify a resource, there is one canonical name and a number of aliases. The CNAME provides the canonical name.
CLIENT.UDP.DNS.IS_MXREC	Returns a Boolean TRUE if the record is of type MX (mail exchanger). This DNS record describes a priority and a host name. The MX records for the same domain name specify the email servers in the domain and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GET Operation	Description
	the priority for each server.
CLIENT.UDP.DNS.IS_NSREC	Returns a Boolean TRUE if the record is of type NS. This is a name server record that includes a host name with an associated A record. This enables locating the domain name that is associated with the NS record.
CLIENT.UDP.DNS.IS_PTRREC	Returns a Boolean TRUE if the record is of type PTR. This is a domain name pointer and is often used to associate a domain name with an IPv4 address.
CLIENT.UDP.DNS.IS_SOAREC	Returns a Boolean TRUE if the record is of type SOA. This is a start of authority record.
CLIENT.UDP.DNS.IS_SRVREC	Returns a Boolean TRUE if the record is of type SRV. This is a more general version of the MX record.
CLIENT.UDP.DSTPORT	Returns the numeric ID of the current packet's UDP destination port.
CLIENT.UDP.SRCPORT	Returns the numeric ID of the current packet's UDP source port.
CLIENT.UDP.RADIUS	Returns RADIUS data for the current packet.
CLIENT.UDP.RADIUS.ATTR_TYPE(<type>)	Returns the value for the attribute type specified as the argument.
CLIENT.UDP.RADIUS.USERNAME	Returns the RADIUS user name.
CLIENT.TCP.MSS	Returns the maximum segment size (MSS) for the current connection as a number.
CLIENT.VLAN.ID	Returns the numeric ID of the VLAN through which the current packet entered the NetScaler.
SERVER.TCP.DSTPORT	Returns the numeric ID of the current packet's destination port.
SERVER.TCP.SRCPORT	Returns the numeric ID of the current packet's source port.
SERVER.TCP.OPTIONS	Returns the TCP options set by the server. Examples of TCP options are Maximum Segment Size (MSS), Window Scale, Selective Acknowledgements (SACK), and Time Stamp Option. The COUNT, TYPE(<type>), and TYPE_NAME(<m>) operators can be used with this

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GET Operation	Description
	prefix. For the TCP options set by the client, see the CLIENT.TCP.OPTIONS prefix.
SERVER.TCP.OPTIONS.COUNT	Returns the number of TCP options that the server has set.
SERVER.TCP.OPTIONS.TYPE(<type>)	Returns the value of the TCP option whose type (or *option kind*) is specified as the argument. The value is returned as a string of bytes in big endian format (or *network byte order*). **Parameters:** type - Type value
SERVER.TCP.OPTIONS.TYPE_NAME(<m>)	Returns the value of the TCP option whose enumeration constant is specified as the argument. The enumeration constants that you can pass as the argument are REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG. To specify the TCP option kind instead of these enumeration constants, use CLIENT.TCP.OPTIONS.TYPE(<type>). For other TCP options, you must use CLIENT.TCP.OPTIONS.TYPE(<type>). **Parameters:** m - TCP option enumeration constant
SERVER.VLAN	Operates on the VLAN through which the current packet entered the NetScaler.
SERVER.VLAN.ID	Returns the numeric ID of the VLAN through which the current packet entered the NetScaler.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions for Evaluating a DNS Message and Identifying Its Carrier Protocol

Aug 30, 2013

You can evaluate DNS requests and responses by using expressions that begin with DNS.REQ and DNS.RES, respectively. You can also identify the transport layer protocol that is being used to send the DNS messages.

The following functions return the contents of a DNS query.

Table 1. Functions that return the contents of a DNS query

Function	Description
DNS.REQ.QUESTION.DOMAIN	Return the domain name (the value of the QNAME field) in the question section of the DNS query. The domain name is returned as a text string, which can be passed to EQ(), NE(), and any other functions that work with text.
DNS.REQ.QUESTION.TYPE	Return the query type (the value of the QTYPE field) in the DNS query. The field indicates the type of resource record (for example, A, NS, or CNAME) for which the name server is being queried. The returned value can be compared to one of the following values by using the EQ() and NE() functions: • A • AAAA • NS • SRV • PTR • CNAME • SOA • MX • ANY Note: You can use only the EQ() and NE() functions with the TYPE function. **Example:** DNS.REQ.QUESTION.TYPE.EQ(MX)

The following functions return the contents of a DNS response.

Table 2. Functions that return the contents of a DNS response

Function	Description
DNS.RES.HEADER.RCODE	Return the response code (the value of the RCODE field) in the header section of the DNS response. You can use only the EQ() and NE() functions with the RCODE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	function. Following are the possible values: • NOERROR • FORMERR • SERVFAIL • NXDOMAIN • NOTIMP • REFUSED
DNS.RES.QUESTION.DOMAIN	Return the domain name (the value of the QNAME field) in the question section of the DNS response. The domain name is returned as a text string, which can be passed to EQ(), NE(), and any other functions that work with text.
DNS.RES.QUESTION.TYPE	Return the query type (the value of the QTYPE field) in the question section of the DNS response. The field indicates the type of resource record (for example, A, NS, or CNAME) that is contained in the response. The returned value can be compared to one of the following values by using the EQ() and NE() functions: • A • AAAA • NS • SRV • PTR • CNAME • SOA • MX • ANY You can use only the EQ() and NE() functions with the TYPE function. **Example:** DNS.RES.QUESTION.TYPE.EQ(SOA)

The following functions return the transport layer protocol name.

Table 3. Functions that return the transport layer protocol name

Function	Description
DNS.REQ.TRANSPORT	Return the name of the transport layer protocol that was used to send the DNS query. Possible values returned are TCP and UDP. You can use only the EQ() and NE() functions with the TRANSPORT function. **Example:** DNS.REQ.TRANSPORT.EQ(TCP)
DNS.RES.TRANSPORT	Return the name of the transport layer protocol that was used for the DNS response.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	Possible values returned are TCP and UDP. You can use only the EQ() and NE() functions with the TRANSPORT function. **Example:** DNS.RES.TRANSPORT.EQ(TCP)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XPath and HTML, XML, or JSON Expressions

Mar 20, 2012

The default syntax expression engine supports expressions for evaluating and retrieving data from HTML, XML, and JavaScript Object Notation (JSON) files. This enables you to find specific nodes in an HTML, XML, or JSON document, determine if a node exists in the file, locate nodes in XML contexts (for example, nodes that have specific parents or a specific attribute with a given value), and return the contents of such nodes. Additionally, you can use XPath expressions in rewrite expressions.

The default syntax expression implementation for XPath comprises a default syntax expression prefix (such as "HTTP.REQ.BODY") that designates HTML or XML text, and the XPATH operator that takes the XPath expression as its argument.

HTML files are a largely free-form collection of tags and text elements. You can use the XPATH_HTML operator, which takes an XPath expression as its argument, to process HTML files. JSON files are either a collection of name/value pairs or an ordered list of values. You can use the XPATH_JSON operator, which takes an XPath expression as its argument, to process JSON files.

Table 1. XPath and JSON Expression Prefixes That Return Text

XPath Prefix	Description
<text>.XPATH(xpathex)	Operate on an XML file and return a Boolean value. For example, the following expression returns a Boolean TRUE if a node called "creator" exists under the node "Book" within the first 1000 bytes of the XML file. HTTP.REQ.BODY(1000).XPATH(xp%boolean(//Book/creator)%) Parameters: xpathex - XPath Boolean expression
<text>.XPATH(xpathex)	Operate on an XML file and return a value of data type "double." For example, the following expression converts the string "36" (a price value) to a value of data type "double" if the string is in the first 1000 bytes of the XML file: HTTP.REQ.BODY(1000).XPATH(xp%number(/Book/price)%) Parameters: xpathex - XPath numeric expression
<text>.XPATH(xpathex)	Operate on an XML file and return a node-set or a string. Node-sets are converted to corresponding strings by using the standard XPath string conversion routine. For example, the following expression selects all the nodes that are enclosed by "/Book/creator" (a node-set) in the first 1000 bytes of the body: HTTP.REQ.BODY(1000).XPATH(xp%/Book/creator%) Parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XPath Prefix	Description
<text>.XPATH_HTML(xpathex)	Operate on an HTML file and return a text value. For example, the following expression operates on an HTML file and returns the text enclosed in <title></title> tags if the title HTML element is found in the first 1000 bytes: HTTP.REQ.BODY(1000).XPATH_HTML(xp%/html/head/title%) Parameters: xpathex - XPath text expression
<text>.XPATH_HTML_WITH_MARKUP(xpathex)	Operate on an HTML file and return a string that contains the entire selected portion of the document, including markup such as including the enclosing element tags. The following expression operates on the HTML file and selects all content within the <title> tag, including markup. HTTP.REQ.BODY(1000).XPATH_HTML_WITH_MARKUP(xp%/html/head/title%) The portion of the HTML body that is selected by the expression is marked for further processing. Parameters: xpathex - XPath expression
<text>.XPATH_JSON(xpathex)	Operate on a JSON file and return a Boolean value. For example, consider the following JSON file: { "Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>' } } The following expression operates on the JSON file and returns a Boolean TRUE if the JSON file contains a node named “creator,” whose parent node is “Book,” in the first 1000 bytes: HTTP.REQ.BODY(1000).XPATH_JSON(xp%boolean(/Book/creator)%) Parameters: xpathex - XPath Boolean expression
<text>.XPATH_JSON(xpathex)	Operate on a JSON file and return a value of data type “double.” For example, consider the following JSON file: { "Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>', "price":"36" } }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XPath Prefix	Description
	The following expression operates on the JSON file and converts the string “36” to a value of data type “double” if the string is present in the first 1000 bytes of the JSON file. HTTP.REQ.BODY(1000).XPATH_JSON(xp%number(/Book/price)%) Parameters: xpathex - XPath numeric expression
<text>.XPATH_JSON(xpathex)	Operate on a JSON file and return a node-set or a string. Node-sets are converted to corresponding strings by using the standard XPath string conversion routine. For example, consider the following JSON file: { "Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>' } } The following expression selects all the nodes that are enclosed by “/Book" (a node-set) in the first 1000 bytes of the body of the JSON file and returns the corresponding string value, which is "<name><title>": HTTP.REQ.BODY(1000).XPATH_JSON(xp%/Book%) Parameters: xpathex - XPath expression
<text>.XPATH_JSON_WITH_MARKUP(xpathex)	Operate on an XML file and return a string that contains the entire portion of the document for the result node, including markup such as including the enclosing element tags. For example, consider the following JSON file: {"Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>' } } The following expression operates on the JSON file and selects all the nodes that are enclosed by “/Book/creator" in the first 1000 bytes of the body, which is “creator:{ person:{ name:'<name>' } }.” HTTP.REQ.BODY(1000).XPATH_JSON_WITH_MARKUP(xp%/Book/creator%) The portion of the JSON body that is selected by the expression is marked for further processing. Parameters: xpathex - XPath expression
<text>.XPATH_WITH_MARKUP(xpathex)	Operate on an XML file and return a string that contains the entire portion of the document for the result node, including markup such as including the enclosing element tags. For example, the following expression operates on an XML file and selects all the nodes enclosed by “/Book/creator" in the first 1000 bytes of the body.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XPath Prefix	HTTP.REQ.BODY(1000).XPATH_WITH_MARKUP(xp%/Book/creator%)
	Description The portion of the JSON body that is selected by the expression is marked for further processing. Parameters: xpathex - XPath expression

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Encrypting and Decrypting XML Payloads

Sep 02, 2013

You can use the XML_ENCRYPT() and XML_DECRYPT() functions in default syntax expressions to encrypt and decrypt, respectively, XML data. These functions conform to the W3C XML Encryption standard defined at "http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/." XML_ENCRYPT() and XML_DECRYPT() support a subset of the XML Encryption specification. In the subset, data encryption uses a bulk cipher method (RC4, DES3, AES128, AES192, or AES256), and an RSA public key is used to encrypt the bulk cipher key.

Note: If you want to encrypt and decrypt text in a payload, you must use the ENCRYPT and DECRYPT functions. For more information about these functions, see "Encrypting and Decrypting Text."

The XML_ENCRYPT() and XML_DECRYPT() functions are not dependent on the encryption/decryption service that is used by the ENCRYPT and DECRYPT commands for text. The cipher method is specified explicitly as an argument to the XML_ENCRYPT() function. The XML_DECRYPT() function obtains the information about the specified cipher method from the <xenc:EncryptedData> element. Following are synopses of the XML encryption and decryption functions:

- **XML_ENCRYPT(<certKeyName>, <method> [, <flags>])**. Returns an <xenc:EncryptedData> element that contains the encrypted input text and the encryption key, which is itself encrypted by using RSA.
- **XML_DECRYPT(<certKeyName>)**. Returns the decrypted text from the input <xenc:EncryptedData> element, which includes the cipher method and the RSA-encrypted key.

Note: The <xenc:EncryptedData> element is defined in the W3C XML Encryption specification.

Following are descriptions of the arguments:

certKeyName

Selects an X.509 certificate with an RSA public key for XML_ENCRYPT() or an RSA private key for XML_DECRYPT(). The certificate key must have been previously created by an add ssl certKey command.

method

Specifies which cipher method to use for encrypting the XML data. Possible values: RC4, DES3, AES128, AES192, AES256.

flags

A bitmask specifying the following optional key information (<ds:KeyInfo>) to be included in the <xenc:EncryptedData> element that is generated by XML_ENCRYPT():

- **1** - Include a KeyName element with the certKeyName. The element is <ds:KeyName>.
- **2** - Include a KeyValue element with the RSA public key from the certificate. The element is <ds:KeyValue>.
- **4** - Include an X509IssuerSerial element with the certificate serial number and issuer DN. The element is <ds:X509IssuserSerial>.
- **8** - Include an X509SubjectName element with the certificate subject DN. The element is <ds:X509SubjectName>.
- **16** - Include an X509Certificate element with the entire certificate. The element is <ds:X509Certificate>.

Using the XML_ENCRYPT() and XML_DECRYPT() Functions in Expressions

The XML encryption feature uses SSL certificate-key pairs to provide X.509 certificates (with RSA public keys) for key encryption and RSA private keys for key decryption. Therefore, before you use the XML_ENCRYPT() function in an expression, you must create an SSL certificate-key pair. The following command creates an SSL certificate-key pair, my-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

certkey, with the X.509 certificate, my-cert.pem, and the private key file, my-key.pem.

add ssl certKey my-certkey -cert my-cert.pem -key my-key.pem -passcrypt kxPeMRYnitY=

The following CLI commands create rewrite actions and policies for encrypting and decrypting XML content.

add rewrite action my-xml-encrypt-action replace "HTTP.RES.BODY(10000).XPATH_WITH_MARKUP(xp%/%)" "HTTP.RES.BODY(10000).XPATH_WITH_MARKUP(xp%/%).XML_ENCRYPT(\"my-certkey\", AES256, 31)" -bypassSafetyCheck YES

add rewrite action my-xml-decrypt-action replace "HTTP.REQ.BODY(10000).XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)" "HTTP.REQ.BODY(10000).XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%).XML_DECRYPT(\"my-certkey\")" -bypassSafetyCheck YES

add rewrite policy my-xml-encrypt-policy "HTTP.REQ.URL.CONTAINS(\"xml-encrypt\")" my-xml-encrypt-action

add rewrite policy my-xml-decrypt-policy "HTTP.REQ.BODY(10000).XPATH(xp%boolean(//xenc:EncryptedData)%)" my-xml-decrypt-action

bind rewrite global my-xml-encrypt-policy 30

bind rewrite global my-xml-decrypt-policy 30

In the above example, the rewrite action my-xml-encrypt-action encrypts the entire XML document (XPATH_WITH_MARKUP(xp%/%)) in the request by using the AES-256 bulk encryption method and the RSA public key from my-certkey to encrypt the bulk encryption key. The action replaces the document with an <xenc:EncryptedData> element containing the encrypted data and an encrypted key. The flags represented by 31 include all of the optional <ds:KeyInfo> elements.

The action my-xml-decrypt-action decrypts the first <xenc:EncryptedData> element in the response (XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)). This requires the prior addition of the xenc XML namespace by use of the following CLI command:

add ns xmlnamespace xenc http://www.w3.org/2001/04/xmlenc#

The my-xml-decrypt-action action uses the RSA private key in my-certkey to decrypt the encrypted key and then uses the bulk encryption method specified in the element to decrypt the encrypted contents. Finally, the action replaces the encrypted data element with the decrypted content.

The rewrite policy my-xml-encrypt-policy applies my-xml-encrypt-action to requests for URLs containing xml-encrypt. The action encrypts the entire response from a service configured on the NetScaler appliance.

The rewrite policy my-xml-decrypt-policy applies my-xml-decrypt-action to requests that contain an <xenc:EncryptedData> element ((XPATH(xp%//xenc:EncryptedData%) returns a non-empty string). The action decrypts the encrypted data in requests that are bound for a service configured on the NetScaler appliance.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Default Syntax Expressions: Parsing SSL Certificates

Sep 20, 2017

You can use default syntax expressions to evaluate X.509 Secure Sockets Layer (SSL) client certificates. A client certificate is an electronic document that can be used to authenticate a user's identity. A client certificate contains (at a minimum) version information, a serial number, a signature algorithm ID, an issuer name, a validity period, a subject (user) name, a public key, and signatures.

You can examine both SSL connections and data in client certificates. For example, you may want to send SSL requests that use low-strength ciphers to a particular load balancing virtual server farm. The following command is an example of a Content Switching policy that parses the cipher strength in a request and matches cipher strengths that are less than or equal to 40:

```
add cs policy p1 -rule "client.ssl.cipher_bits.le(40)"
```

As another example, you can configure a policy that determines whether a request contains a client certificate:

```
add cs policy p2 -rule "client.ssl.client_cert EXISTS"
```

Or, you might want to configure a policy that examines particular information in a client certificate. For example, the following policy verifies that the certificate has one or more days before expiration:

```
add cs policy p2 -rule "client.ssl.client_cert exists && client.ssl.client_cert.days_to_expire.ge(1)"
```

Note: For information on parsing dates and times in a certificate, see "Format of Dates and Times in an Expression" and "Expressions for SSL Certificate Dates."
This document includes the following details:

- Prefixes for Text-Based SSL and Certificate Data
- Prefixes for Numeric Data in SSL Certificates
- Expressions for SSL Certificates

Prefixes for Text-Based SSL and Certificate Data

The following table describes expression prefixes that identify text-based items in SSL transactions and client certificates.

Table 1. Prefixes That Return Text or Boolean Values for SSL and Client Certificate Data

Prefix	Description
CLIENT.SSL.CLIENT_CERT	Returns the SSL client certificate in the current SSL transaction.
CLIENT.SSL.CLIENT_CERT.TO_PEM	Returns the SSL client certificate in binary format.
CLIENT.SSL.CIPHER_EXPORTABLE	Returns a Boolean TRUE if the SSL cryptographic SSL cryptographic cipher is exportable.
CLIENT.SSL.CIPHER_NAME	Returns the name of the SSL Cipher if invoked from an SSL connection, and a NULL string if invoked from a non-SSL connection.
CLIENT.SSL.IS_SSL	Returns a Boolean TRUE if the current connection is SSL-based.

Prefixes for Numeric Data in SSL Certificates

Updated: 2015-06-17

The following table describes prefixes that evaluate numeric data other than dates in SSL certificates. These prefixes can be used with the operations that are described in "Basic Operations on Expression Prefixes" and "Compound Operations for Numbers."

Table 2. Prefixes That Evaluate Numeric Data Other Than Dates in SSL Certificates

Prefix	Description
CLIENT.SSL.CLIENT_CERT.DAYS_TO_EXPIRE	Returns the number of days that the certificate is valid, or returns -1 for expired certificates.
CLIENT.SSL.CLIENT_CERT.PK_SIZE	Returns the size of the public key used in the certificate.
CLIENT.SSL.CLIENT_CERT.VERSION	Returns the version number of the certificate. If the connection is not SSL-based, returns zero (0).
CLIENT.SSL.CIPHER_BITS	Returns the number of bits in the cryptograhic key. Returns 0 if the connection is not SSL-based.
CLIENT.SSL.VERSION	Returns a number that represents the SSL protocol version, as follows: • 0. The transaction is not SSL-based. • 0x002. The transaction is SSLv2. • 0x300. The transaction is SSLv3. • 0x301. The transaction is TLSv1. • 0x302. The transaction is TLSv1.1. • 0x303. The transaction is TLSv1.2.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: For expressions related to expiration dates in a certificate, see "Expressions for SSL Certificate Dates."

Expressions for SSL Certificates

Updated: 2013-09-02

You can parse SSL certificates by configuring expressions that use the following prefix:

CLIENT.SSL.CLIENT_CERT

This section discusses the expressions that you can configure for certificates, with the exception of expressions that examine certificate expiration. Time-based operations are described in "Default Syntax Expressions: Working with Dates, Times, and Numbers."

The following table describes operations that you can specify for the CLIENT.SSL.CLIENT_CERT prefix.

Table 3. Operations That Can Be Specified with the CLIENT.SSL.CLIENT_CERT Prefix

SSL Certificate Operation	Description
<certificate>.EXISTS	Returns a Boolean TRUE if the client has an SSL certificate.
<certificate>.ISSUER	Returns the Distinguished Name (DN) of the Issuer in the certificate as a name-value list. An equals sign ("=") is the delimiter for the name and the value, and the slash ("/") is the delimiter that separates the name-value pairs. Following is an example of the returned DN: /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycompany.c
<certificate>.ISSUER. IGNORE_EMPTY_ELEMENTS	Returns the Issuer and ignores the empty elements in a name-value list. For example, consider the following: Cert-Issuer: /c=in/st=kar//l=bangelore //o=mycompany/ou=sales/ /emailAddress=myuserid@mycompany.com The following Rewrite action returns a count of 6 based on the preceding Issuer definition: sh rewrite action insert_ssl_header Name: insert_ssl Operation: insert_http_header Target:Cert-Issuer Value:CLIENT.SSL.CLIENT_CERT.ISSUER.COUNT However, if you change the value to the following, the returned count is 9: CLIENT.SSL.CLIENT_CERT.ISSUER.IGNORE_EMPTY_ELEMENTS.COUNT
<certificate>.AUTH_KEYID	Returns a string that contains the Authority Key Identifier extension of the X.509 V3 certificate.
<certificate>.AUTH_KEYID.CERT_SERIALNUMBER	Returns the SerialNumber field of the Authority Key Identifier as a blob.
<certificate>.AUTH_KEYID.EXISTS	Returns a Boolean TRUE if the certificate contains an Authority Key Identifier extension.
<certificate>.AUTH_KEYID.ISSUER_NAME	Returns the Issuer Distinguished Name in the certificate as a name-value list. An equals sign ("=") is the delimiter for the name and the value, and the slash ("/") is the delimiter that separates the name-value pairs. Following is an example: /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycompany.c
<certificate>.AUTH_KEYID.ISSUER_NAME.IGNORE_EMPTY_ELEMENTS	Returns the Issuer Distinguished Name in the certificate as a name-value list and ignores the empty elements in the list For example, the following name-value list has an empty element following "a=10": a=10;;b=11; ;c=89 The element following b=11 is not considered an empty element.
<certificate>.AUTH_KEYID.KEYID	Returns the keyIdentifier field of the Authority Key Identifier as a blob.
<certificate>.CERT_POLICY	Returns a string that contains the client certificate policy. Note that this represents a sequence of certificate policies.
<certificate>.KEY_USAGE(string)	Returns a Boolean value to indicate whether the specified key usage extension bit value in the X.509 certificate is set. The string argument specifies which bit is checked. Following are valid arguments: • DIGITAL_SIGNATURE. Returns TRUE if the digital signature bit is set; otherwise, it returns FALSE. • NONREPUDIATION. Returns TRUE if the nonrepudiation bit is set; otherwise, it returns FALSE. • KEYENCIPHERMENT. Returns TRUE if the key encipherment bit is set; otherwise, it returns FALSE.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Certificate Operation	Description
	• DATAENCIPHERMENT. Returns TRUE if the data encipherment bit is set; otherwise, it returns FALSE. • KEYAGREEMENT. Returns TRUE if the key agreement bit is set; otherwise, it returns FALSE. • KEYCERTSIGN. Returns TRUE if the key cert sign bit is set; otherwise, it returns FALSE. • CRLSIGN. Returns TRUE if the CRL bit is set; otherwise, it returns FALSE. • ENCIPHERONLY. Returns TRUE if the encipher only bit is set; otherwise, it returns FALSE. • DECIPHERONLY. Returns TRUE if the decipher only bit is set; otherwise, it returns FALSE.
<certificate>.PK_ALGORITHM	Returns the name of the public key algorithm used by the certificate.
<certificate>.PK_SIZE	Returns the size of the public key used in the certificate.
<certificate>.SERIALNUMBER	Returns the serial number of the client certificate. If this is a non-SSL transaction or there is an error in the certificate, this operation returns an empty string.
<certificate>.SIGNATURE_ALGORITHM	Returns the name of the cryptographic algorithm used by the CA to sign this certificate.
<certificate>.SUBJECT	Returns the Distinguished Name of the Subject as a name-value. An equals sign ("=") separates names and values and a slash ("/") delimits name-value pairs. Following is an example: /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycompany.c
<certificate>.SUBJECT.IGNORE_EMPTY_ELEMENTS	Returns the Subject as a name-value list, but ignores the empty elements in the list. For example, consider the following Cert-Issuer: /c=in/st=kar//l=bangelore //o=mycompany/ou=sales/ /emailAddress=myuserid@mycompany.com The following Rewrite action returns a count of 6 based on the preceding Issuer definition: sh rewrite action insert_ssl_header Name: insert_ssl Operation: insert_http_header Target:Cert-Issuer Value:CLIENT.SSL.CLIENT_CERT.ISSUER.COUNT However, if you change the value to the following, the returned count is 9: CLIENT.SSL.CLIENT_CERT.ISSUER.IGNORE_EMPTY_ELEMENTS.COUNT
<certificate>.SUBJECT_KEYID	Returns the Subject KeyID of the client certificate. If there is no Subject KeyID, this operation returns a zero-length te object.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs

Sep 20, 2017

You can use default syntax expression prefixes that return IPv4 and IPv6 addresses, MAC addresses, IP subnets, useful client and server data such as the throughput rates at the interface ports (Rx, Tx, and RxTx), and the IDs of the VLANs through which packets are received. You can then use various operators to evaluate the data that is returned by these expression prefixes.

This document includes the following details:

- Expressions for IP Addresses and IP Subnets
- Expressions for MAC Addresses
- Expressions for Numeric Client and Server Data

Expressions for IP Addresses and IP Subnets

Updated: 2013-09-02

You can use default syntax expressions to evaluate addresses and subnets that are in Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) format. Expression prefixes for IPv6 addresses and subnets include IPv6 in the prefix. Expression prefixes for IPv4 addresses and subnets include IP in the prefix. Following is an example of an expression that identifies whether a request has originated from a particular IPv4 subnet.

```
client.ip.src.in_subnet(147.1.0.0/16)
```

Following are two examples of Rewrite policies that examine the subnet from which the packet is received and perform a rewrite action on the Host header. With these two policies configured, the rewrite action that is performed depends on the subnet in the request. These two policies evaluate IP addresses that are in the IPv4 address format.

```
add rewrite action URL1-rewrite-action replace "http.req.header(\"Host\")" "\"www.mycompany1.com\""
add rewrite policy URL1-rewrite-policy "http.req.header(\"Host\").contains(\"www.test1.com\") && client.ip.src.in_subnet(147.1.0.0/16)" URL1-rewrite-action
add rewrite action URL2-rewrite-action replace "http.req.header(\"Host\")" "\"www.mycompany2.com\""
add rewrite policy URL2-rewrite-policy "http.req.header(\"Host\").contains(\"www.test2.com\") && client.ip.src.in_subnet(10.202.0.0/16)" URL2-rewrite-action
```

Note: The preceding examples are commands that you type at the NetScaler command-line interface (CLI) and, therefore, each quotation mark must be preceded by a backslash (\). For more information, see "Configuring Default Syntax Expressions in a Policy."

Prefixes for IPV4 Addresses and IP Subnets

Updated: 2013-09-02

The following table describes prefixes that return IPv4 addresses and subnets, and segments of IPv4 addresses. You can use numeric operators and operators that are specific to IPv4 addresses with these prefixes. For more information about numeric operations, see "Basic Operations on Expression Prefixes" and "Compound Operations for Numbers."

Table 1. Prefixes That Evaluate IP and MAC Addresses

Prefix	Description
CLIENT.IP.SRC	Returns the source IP of the current packet as an IP address or as a number.
CLIENT.IP.DST	Returns the destination IP of the current packet as an IP address or as a number.
SERVER.IP.SRC	Returns the source IP of the current packet as an IP address or as a number.
SERVER.IP.DST	Returns the destination IP of the current packet as an IP address or as a number.

Operations for IPV4 Addresses

The following table describes the operators that can be used with prefixes that return an IPv4 address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 2. Operations on IPV4 Addresses

Prefix	Description
<ip address>.EQ(<address>)	Returns a Boolean TRUE if the IP address value is same as the <address> argument. The following example checks whether the client's destination IP address is equal to 10.100.10.100: client.ip.dst.eq(10.100.10.100)
<ip address>.GET1. . .GET4	Returns a portion of an IP address as a numeric value. For example, if the IP address value is 10.100.200.1, the following is returned: client.ip.src.get1 Returns 10 client.ip.src.get2 returns 100 client.ip.src.get3 returns 200
<ip address>.IN_SUBNET(<subnet>)	Returns a Boolean TRUE if the <subnet> argument matches the subnet of the IP address value. For example, the following determines whether the client's destination IP address subnet is 10.100.10.100/18: client.ip.dst.eq(10.100.10.100/18)
<ip address>.SUBNET(<n>)	Returns the IP address after applying the subnet mask specified as the argument. The subnet mask can take values between 0 and 32. For example: CLIENT.IP.SRC.SUBNET(24) returns 192.168.1.0 if the IP address represented by the prefix is 192.168.1.[0-255].
<ip address>.IS_IPV6	Returns a Boolean TRUE if this is an Internet Protocol version 6 (IPv6) host for the client or server. Following is an example: client.ip.src.is_ipv6
<ip address>.MATCHES(<hostname>)	Returns a Boolean TRUE if the IP address for the host specified in <hostname> matches the current IP address. The <hostname> cannot exceed 255 characters.
<ip address>.MATCHES_LOCATION(<location>)	Returns a Boolean TRUE if the location of the IP address matches the <location> argument. The Location string can take the following form: qual1.qual2.qual3.qual4.qual5.qual6, for example: NorthAmeria.CA.* Following is an example: client.ip.src.matches_location(\"Europe.GB.17.London.*.*\")

About IPv6 Expressions

The IPv6 address format allows more flexibility than the older IPv4 format. IPv6 addresses are in the hexadecimal format (RFC 2373). In the following examples, Example 1 is an IPv6 address, Example 2 is a URL that includes the IPv6 address, and Example 3 includes the IPv6 address and a port number.

Example 1:
9901:0ab1:22a2:88a3:3333:4a4b:5555:6666
Example 2:
http://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]/

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 3:
https://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]:8080/
In Example 3, the brackets separate the IP address from the port number (8080).

Note that you can only use the '+' operator to combine IPv6 expressions with other expressions. The output is a concatenation of the string values that are returned from the individual expressions. You cannot use any other arithmetic operator with an IPv6 expression. The following syntax is an example:

client.ipv6.src + server.ip.dst
For example, if the client source IPv6 address is ABCD:1234::ABCD, and the server destination IPv4 address is 10.100.10.100, the preceding expression returns "ABCD:1234::ABCD10.100.10.100".

Note that when the NetScaler appliance receives an IPv6 packet, it assigns a temporary IPv4 address from an unused IPv4 address range and changes the source address of the packet to this temporary address. At response time, the outgoing packet's source address is replaced with the original IPv6 address.

Note: You can combine an IPv6 expression with any other expression except an expression that produces a Boolean result.

Expression Prefixes for IPv6 Addresses

The IPv6 addresses that are returned by the expression prefixes in the following table can be treated as text data. For example, the prefix client.ipv6.dst returns the destination IPv6 address as a string that can be evaluated as text.

The following table describes expression prefixes that return an IPv6 address.

Table 3. IPv6 Expression Prefixes That Return Text

Prefix	Description
CLIENT.IPV6	Operates on the IPv6 address in with the current packet.
CLIENT.IPV6.DST	Returns the IPv6 address in the destination field of the IP header.
CLIENT.IPV6.SRC	Returns the IPv6 address in the source field of the IP header. Following are examples: client.ipv6.src.in_subnet(2007::2008/64) client.ipv6.src.get1.le(2008)
SERVER.IPV6	Operates on the IPv6 address in with the current packet.
SERVER.IPV6.DST	Returns the IPv6 address in the destination field of the IP header.
SERVER.IPV6.SRC	Returns the IPv6 address in the source field of the IP header. Following are examples: server.ipv6.src.in_subnet(2007::2008/64) server.ipv6.src.get1.le(2008)

Operations for IPV6 Prefixes

The following table describes the operators that can be used with prefixes that return an IPv6 address:

Table 4. Operations That Evaluate IPv6 Addresses

IPv6 Operation	Description
<ipv6>.EQ(<IPv6_address>)	Returns a Boolean TRUE if the IP address value is same as the <IPv6_address> argument. Following is an example:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

IPv6 Operation	Description
	client.ipv6.dst.eq(ABCD:1234::ABCD)
<ipv6>.GET1. . .GET8	Returns a segment of an IPv6 address as a number. The following example expressions retrieve segments from the ipv6 address 1000:1001:CD10:0000:0000:89AB:4567:CDEF: • client.ipv6.dst.get5 extracts 0000, which is the fifth set of bits in the address. • client.ipv6.dst.get6 extracts 89AB. • client.ipv6.dst.get7 extracts 4567. You can perform numeric operations on these segments. Note that you cannot perform numeric operations when you retrieve an entire IPv6 address. This is because expressions that return an entire IPv6 address, such as CLIENT.IPV6.SRC, return the address in text format.
<ipv6>.IN_SUBNET(<subnet>)	Returns a Boolean TRUE if the IPv6 address value is in the subnet specified by the <subnet> argument. Following is an example: client.ipv6.dst.eq(1000:1001:CD10:0000:0000:89AB:4567:CDEF/60)
<ipv6>.IS_IPV4	Returns a Boolean TRUE if this is an IPv4 client, and returns a Boolean FALSE if it is not.
<ipv6>.SUBNET(<n>)	Returns the IPv6 address after applying the subnet mask specified as the argument. The subnet mask can take values between 0 and 128. For example: CLIENT.IPV6.SRC.SUBNET(24)

Expressions for MAC Addresses

A MAC address consists of colon-delimited hexadecimal values in the format ##:##:##:##:##:##, where each “#” represents either a number from 0 through 9 or a letter from A through F. Default syntax expression prefixes and operators are available for evaluating source and destination MAC addresses.

Prefixes for MAC Addresses

The following table describes prefixes that return MAC addresses.

Table 5. Prefixes That Evaluate MAC Addresses

Prefix	Description
client.ether.dstmac	Returns the MAC address in the destination field of the Ethernet header.
client.ether.srcmac	Returns the MAC address in the source field of the Ethernet header.

Operations for MAC Addresses

The following table describes the operators that can be used with prefixes that return a MAC address.

Table 6. Operations on MAC Addresses

Prefix	Description
<mac address>.EQ(<address>)	Returns a Boolean TRUE if the MAC address value is same as the <address> argument.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Prefix	Description
<mac address>.GET1. . .GET4	Returns a numeric value extracted from the segment of the MAC address that is specified in the GET operation. For example, if the MAC address is 12:34:56:78:9a:bc, the following returns 34: client.ether.dstmac.get2

Expressions for Numeric Client and Server Data

The following table describes prefixes for working with numeric client and server data, including throughput, port numbers, and VLAN IDs.

Table 7. Prefixes That Evaluate Numeric Client and Server Data

Prefix	Description
client.interface.rxthroughput	Returns an integer representing the raw received traffic throughput in kilobytes per second (KBps) for the previous seven seconds.
client.interface.txthroughput	Returns an integer representing the raw transmitted traffic throughput in KBps for the previous seven seconds.
client.interface.rxtxthroughput	Returns an integer representing the raw received and transmitted traffic throughput in KBps for the previous seven seconds.
server.interface.rxthroughput	Returns an integer representing the raw received traffic throughput in KBps for the previous seven seconds.
server.interface.txthroughput	Returns an integer representing the raw transmitted traffic throughput in KBps for the previous seven seconds.
server.interface.rxtxthroughput	Returns an integer representing the raw received and transmitted traffic throughput in KBps for the previous seven seconds.
server.vlan.id	Returns a numeric ID of the VLAN through which the current packet entered the NetScaler.
client.vlan.id	Returns a numeric ID for the VLAN through which the current packet entered the NetScaler.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Default Syntax Expressions: Stream Analytics Functions

May 21, 2015

Stream Analytics expressions begin with the ANALYTICS.STREAM(<identifier_name>) prefix. The following list describes the functions that can be used with this prefix.

COLLECT_STATS

Collect statistical data from the requests that are evaluated against the policy and create a record for each request.

REQUESTS

Return the number of requests that exist for the specified record grouping. The value returned is of type unsigned long.

BANDWIDTH

Return the bandwidth statistic for the specified record grouping. The value returned is of type unsigned long.

RESPTIME

Return the response time statistic for the specified record grouping. The value returned is of type unsigned long.

CONNECTIONS

Return the number of concurrent connections that exist for the specified record grouping. The value returned is of type unsigned long.

IS_TOP(n)

Return a Boolean TRUE if the statistical value for the specified record grouping is one among the top n groups. Otherwise, return a Boolean FALSE.

CHECK_LIMIT

Return a Boolean TRUE if the statistic for the specified record grouping has hit the preconfigured limit. Otherwise, return a Boolean FALSE.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Default Syntax Expressions: DataStream

May 25, 2015

The policy infrastructure on the Citrix NetScaler appliance includes expressions that you can use to evaluate and process database server traffic when the appliance is deployed between a farm of application servers and their associated database servers.

This document includes the following details:

- Expressions for the MySQL Protocol
- Expressions for Evaluating Microsoft SQL Server Connections

Expressions for the MySQL Protocol

The following expressions evaluate traffic associated with MySQL database servers. You can use the request-based expressions (expressions that begin with MYSQL.CLIENT and MYSQL.REQ) in policies to make request switching decisions at the content switching virtual server bind point and the response-based expressions (expressions that begin with MYSQL.RES) to evaluate server responses to user-configured health monitors.

- **MYSQL.CLIENT.** Operates on the client properties of a MySQL connection.
- **MYSQL.CLIENT.CAPABILITIES.** Returns the set of flags that the client has set in the capabilities field of the handshake initialization packet during authentication. Examples of the flags that are set are CLIENT_FOUND_ROWS, CLIENT_COMPRESS, and CLIENT_SSL.
- **MYSQL.CLIENT.CHAR_SET.** Returns the enumeration constant assigned to the character set that the client uses. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. Following are the character set enumeration constants:
  - LATIN2_CZECH_CS
  - DEC8_SWEDISH_CI
  - CP850_GENERAL_CI
  - GREEK_GENERAL_CI
  - LATIN1_GERMAN1_CI
  - HP8_ENGLISH_CI
  - KOI8R_GENERAL_CI
  - LATIN1_SWEDISH_CI
  - LATIN2_GENERAL_CI
  - SWE7_SWEDISH_CI
  - ASCII_GENERAL_CI
  - CP1251_BULGARIAN_CI
  - LATIN1_DANISH_CI
  - HEBREW_GENERAL_CI
  - LATIN7_ESTONIAN_CS
  - LATIN2_HUNGARIAN_CI
  - KOI8U_GENERAL_CI
  - CP1251_UKRAINIAN_CI
  - CP1250_GENERAL_CI
  - LATIN2_CROATIAN_CI
  - CP1257_LITHUANIAN_CI
  - LATIN5_TURKISH_CI

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- LATIN1_GERMAN2_CI
- ARMSCII8_GENERAL_CI
- UTF8_GENERAL_CI
- CP1250_CZECH_CS
- CP866_GENERAL_CI
- KEYBCS2_GENERAL_CI
- MACCE_GENERAL_CI
- MACROMAN_GENERAL_CI
- CP852_GENERAL_CI
- LATIN7_GENERAL_CI
- LATIN7_GENERAL_CS
- MACCE_BIN
- CP1250_CROATIAN_CI
- LATIN1_BIN
- LATIN1_GENERAL_CI
- LATIN1_GENERAL_CS
- CP1251_BIN
- CP1251_GENERAL_CI
- CP1251_GENERAL_CS
- MACROMAN_BIN
- CP1256_GENERAL_CI
- CP1257_BIN
- CP1257_GENERAL_CI
- ARMSCII8_BIN
- ASCII_BIN
- CP1250_BIN
- CP1256_BIN
- CP866_BIN
- DEC8_BIN
- GREEK_BIN
- HEBREW_BIN
- HP8_BIN
- KEYBCS2_BIN
- KOI8R_BIN
- KOI8U_BIN
- LATIN2_BIN
- LATIN5_BIN
- LATIN7_BIN
- CP850_BIN
- CP852_BIN
- SWE7_BIN
- UTF8_BIN
- GEOSTD8_GENERAL_CI
- GEOSTD8_BIN
- LATIN1_SPANISH_CI
- UTF8_UNICODE_CI

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- UTF8_ICELANDIC_CI
- UTF8_LATVIAN_CI
- UTF8_ROMANIAN_CI
- UTF8_SLOVENIAN_CI
- UTF8_POLISH_CI
- UTF8_ESTONIAN_CI
- UTF8_SPANISH_CI
- UTF8_SWEDISH_CI
- UTF8_TURKISH_CI
- UTF8_CZECH_CI
- UTF8_DANISH_CI
- UTF8_LITHUANIAN_CI
- UTF8_SLOVAK_CI
- UTF8_SPANISH2_CI
- UTF8_ROMAN_CI
- UTF8_PERSIAN_CI
- UTF8_ESPERANTO_CI
- UTF8_HUNGARIAN_CI
- INVAL_CHARSET

- **MYSQL.CLIENT.DATABASE.** Returns the name of the database specified in the authentication packet that the client sends to the database server. This is the databasename attribute.
- MYSQL.CLIENT.USER. Returns the user name (in the authentication packet) with which the client is attempting to connect to the database. This is the user attribute.
- **MYSQL.REQ.** Operates on a MySQL request.
- **MYSQL.REQ.COMMAND.** Identifies the enumeration constant assigned to the type of command in the request. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. Following are the enumeration constant values:
  - SLEEP
  - QUIT
  - INIT_DB
  - QUERY
  - FIELD_LIST
  - CREATE_DB
  - DROP_DB
  - REFRESH
  - SHUTDOWN
  - STATISTICS
  - PROCESS_INFO
  - CONNECT
  - PROCESS_KILL
  - DEBUG
  - PING
  - TIME
  - DELAYED_INSERT
  - CHANGE_USER
  - BINLOG_DUMP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- TABLE_DUMP
- CONNECT_OUT
- REGISTER_SLAVE
- STMT_PREPARE
- STMT_EXECUTE
- STMT_SEND_LONG_DATA
- STMT_CLOSE
- STMT_RESET
- SET_OPTION
- STMT_FETCH

- **MYSQL.REQ.QUERY.** Identifies the query in the MySQL request.
- **MYSQL.REQ.QUERY.COMMAND.** Returns the first keyword in the MySQL query.
- **MYSQL.REQ.QUERY.SIZE.** Returns the size of the request query in integer format. The SIZE method is similar to the CONTENT_LENGTH method that returns the length of an HTTP request or response.
- **MYSQL.REQ.QUERY.TEXT.** Returns a string covering the entire query.
- **MYSQL.REQ.QUERY.TEXT(<n>).** Returns the first n bytes of the MySQL query as a string. This is similar to HTTP.BODY(<n>).
  Parameters:

  n - Number of bytes to be returned

- **MYSQL.RES.** Operates on a MySQL response.
- **MYSQL.RES.ATLEAST_ROWS_COUNT(<i>).** Checks whether the response has at least i number of rows and returns a Boolean TRUE or FALSE to indicate the result.
  Parameters:

  i - Number of rows

- **MYSQL.RES.ERROR.** Identifies the MySQL error object. The error object includes the error number and the error message.
- **MYSQL.RES.ERROR.MESSAGE.** Returns the error message that is retrieved from the server's error response.
- **MYSQL.RES.ERROR.NUM.** Returns the error number that is retrieved from the server's error response.
- **MYSQL.RES.ERROR.SQLSTATE.** Returns the value of the SQLSTATE field in the server's error response. The MySQL server translates error number values to SQLSTATE values.
- **MYSQL.RES.FIELD(<i>).** Identifies the packet that corresponds to the ith individual field in the server's response. Each field packet describes the properties of the associated column. The packet count (i) begins at 0.
  Parameters:

  i - Packet number

- **MYSQL.RES.FIELD(<i>).CATALOG.** Returns the catalog property of the field packet.
- **MYSQL.RES.FIELD(<i>).CHAR_SET.** Returns the character set of the column. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix.
- **MYSQL.RES.FIELD(<i>).DATATYPE.** Returns an enumeration constant that represents the data type of the column. This is the type (also called enum_field_type) attribute of the column. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. The possible values for the various data types are:
  - DECIMAL

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

    - TINY
    - SHORT
    - LONG
    - FLOAT
    - DOUBLE
    - NULL
    - TIMESTAMP
    - LONGLONG
    - INT24
    - DATE
    - TIME
    - DATETIME
    - YEAR
    - NEWDATE
    - VARCHAR (new in MySQL 5.0)
    - BIT (new in MySQL 5.0)
    - NEWDECIMAL (new in MySQL 5.0)
    - ENUM
    - SET
    - TINY_BLOB
    - MEDIUM_BLOB
    - LONG_BLOB
    - BLOB
    - VAR_STRING
    - STRING
    - GEOMETRY
- **MYSQL.RES.FIELD(<i>).DB.** Returns the database identifier (db) attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).DECIMALS.** Returns the number of positions after the decimal point if the type is DECIMAL or NUMERIC. This is the decimals attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).FLAGS.** Returns the flags property of the field packet. Following are the possible hexadecimal flag values:
    - 0001: NOT_NULL_FLAG
    - 0002: PRI_KEY_FLAG
    - 0004: UNIQUE_KEY_FLAG
    - 0008: MULTIPLE_KEY_FLAG
    - 0010: BLOB_FLAG
    - 0020: UNSIGNED_FLAG
    - 0040: ZEROFILL_FLAG
    - 0080: BINARY_FLAG
    - 0100: ENUM_FLAG
    - 0200: AUTO_INCREMENT_FLAG
    - 0400: TIMESTAMP_FLAG
    - 0800: SET_FLAG
- **MYSQL.RES.FIELD(<i>).LENGTH.** Returns the length of the column. This is the value of the length attribute of the field packet. The value that is returned might be larger than the actual value. For example, an instance of a VARCHAR(2) column might return a value of 2 even when it contains only one character.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- **MYSQL.RES.FIELD(<i>).NAME.** Returns the column identifier (the name after the AS clause, if any). This is the name attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).ORIGINAL_NAME.** Returns the original column identifier (before the AS clause, if any). This is the org_name attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).ORIGINAL_TABLE.** Returns the original table identifier of the column (before the AS clause, if any). This is the org_table attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).TABLE.** Returns the table identifier of the column (after the AS clause, if any). This is the table attribute of the field packet.
- **MYSQL.RES.FIELDS_COUNT.** Returns the number of field packets in the response (the field_count attribute of the OK packet).
- **MYSQL.RES.OK.** Identifies the OK packet sent by the database server.
- **MYSQL.RES.OK.AFFECTED_ROWS.** Returns the number of rows affected by an INSERT, UPDATE, or DELETE query. This is the value of the affected_rows attribute of the OK packet.
- **MYSQL.RES.OK.INSERT_ID.** Identifies the unique_id attribute of the OK packet. If an auto-increment identity is not generated by the current MySQL statement or query, the value of unique_id, and hence the value returned by the expression, is 0.
- **MYSQL.RES.OK.MESSAGE.** Returns the message property of the OK packet.
- **MYSQL.RES.OK.STATUS.** Identifies the bit string in the server_status attribute of the OK packet. Clients can use the server status to check whether the current command is a part of a running transaction. The bits in the server_status bit string correspond to the following fields (in the given order):
  - IN TRANSACTION
  - AUTO_COMMIT
  - MORE RESULTS
  - MULTI QUERY
  - BAD INDEX USED
  - NO INDEX USED
  - CURSOR EXISTS
  - LAST ROW SEEN
  - DATABASE DROPPED
  - NO BACKSLASH ESCAPES
- **MYSQL.RES.OK.WARNING_COUNT.** Returns the warning_count attribute of the OK packet.
- **MYSQL.RES.ROW(<i>).** Identifies the packet that corresponds to the ith individual row in the database server's response.

  Parameters:

  i - Row number

- **MYSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>).** Checks whether the jth column of the ith row of the table is NULL. Following C conventions, both indexes i and j start from 0. Therefore, row i and column j are actually the $(i+1)^{th}$ row and the $(j+1)^{th}$ column, respectively.

  Parameters:

  i - Row number

  j - Column number

- **MYSQL.RES.ROW(<i>).IS_NULL_ELEM(j).** Checks whether the jth column of the ith row of the table is NULL. Following C conventions, both indexes i and j start from 0. Therefore, row i and column j are actually the $(i+1)^{th}$ row and the $(j+1)^{th}$

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

column, respectively.
Parameters:

i - Row number

j - Column number

- **MYSQL.RES.ROW(<i>).NUM_ELEM(<j>).** Returns an integer value from the jth column of the ith row of the table. Following C conventions, both indexes i and j start from 0. Therefore, row i and column j are actually the $(i+1)^{th}$ row and the $(j+1)^{th}$ column, respectively.
Parameters:

i - Row number

j - Column number

- **MYSQL.RES.ROW(<i>).TEXT_ELEM(j).** Returns a string from the jth column of the ith row of the table. Following C conventions, both indexes i and j start from 0. Therefore, row i and column j are actually the $(i+1)^{th}$ row and the $(j+1)^{th}$ column, respectively.
Parameters:

i - Row number

j - Column number

- **MYSQL.RES.TYPE.** Returns an enumeration constant for the response type. Its values can be ERROR, OK, and RESULT_SET. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix.

Expressions for Evaluating Microsoft SQL Server Connections

The following expressions evaluate traffic associated with Microsoft SQL Server database servers. You can use the request-based expressions (expressions that begin with MSSQL.CLIENT and MSSQL.REQ) in policies to make request switching decisions at the content switching virtual server bind point and the response-based expressions (expressions that begin with MSSQL.RES) to evaluate server responses to user-configured health monitors.

Table 1. Expressions for Evaluating Microsoft SQL Server Connections

Expression	Description
MSSQL.CLIENT.CAPABILITIES	Returns the OptionFlags1, OptionFlags2, OptionFlags3, and TypeFlags fields of the LOGIN7 authentication packet, in that order, as a 4-byte integer. Each field is 1 byte long and specifies a set of client capabilities.
MSSQL.CLIENT.DATABASE	Returns the name of the client database. The value returned is of type text.
MSSQL.CLIENT.USER	Returns the user name with which the client authenticated. The value returned is of type text.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Description
MSSQL.REQ.COMMAND	Returns an enumeration constant that identifies the type of command in the request sent to a Microsoft SQL Server database server. The value returned is of type text. Examples of the values of the enumeration constant are QUERY, RESPONSE, RPC, and ATTENTION. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this expression.
MSSQL.REQ.QUERY.COMMAND	Returns the first keyword in the SQL query. The value returned is of type text.
MSSQL.REQ.QUERY.SIZE	Returns the size of the SQL query in the request. The value returned is a number.
MSSQL.REQ.QUERY.TEXT	Returns the entire SQL query as a string. The value returned is of type text.
MSSQL.REQ.QUERY.TEXT(<n>)	Returns the first n bytes of the SQL query. The value returned is of type text. **Parameters:** n - Number of bytes
MSSQL.REQ.RPC.NAME	Returns the name of the procedure that is being called in a remote procedure call (RPC) request. The name is returned as a string.
MSSQL.REQ.RPC.IS_PROCID	Returns a Boolean value that indicates whether the remote procedure call (RPC) request contains a procedure ID or an RPC name. A return value of TRUE indicates that the request contains a procedure ID and a return value of FALSE indicates that the request contains an RPC name.
MSSQL.REQ.RPC.PROCID	Returns the procedure ID of the remote procedure call (RPC) request as an integer.
MSSQL.REQ.RPC.BODY Note: Not available for releases before 10.1.	Returns the body of the SQL request as a string in the form of parameters represented as "a=b" clauses separated by commas, where "a" is the RPC parameter name and "b" is its value.
MSSQL.REQ.RPC.BODY(n) Note: Not available for releases before 10.1.	Returns part of the body of the SQL request as a string in the form of parameters represented as "a=b" clauses separated by commas, where

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Description
	"a" is the RPC parameter name and "b" is its value. Parameters are returned from only the first "n" bytes of the request, skipping the SQL header. Only complete name-value pairs are returned.
MSSQL.RES.ATLEAST_ROWS_COUNT(i)	Checks whether the response has at least i number of rows. The value returned is a Boolean TRUE or FALSE value. **Parameters:** i - Number of rows
MSSQL.RES.DONE.ROWCOUNT	Returns a count of the number of rows affected by an INSERT, UPDATE, or DELETE query. The value returned is of type unsigned long.
MSSQL.RES.DONE.STATUS	Returns the status field from the DONE token sent by a Microsoft SQL Server database server. The value returned is a number.
MSSQL.RES.ERROR.MESSAGE	Returns the error message from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the MsgText field in the ERROR token. The value returned is of type text.
MSSQL.RES.ERROR.NUM	Returns the error number from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the Number field in the ERROR token. The value returned is a number.
MSSQL.RES.ERROR.STATE	Returns the error state from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the State field in the ERROR token. The value returned is a number.
MSSQL.RES.FIELD(<i>).DATATYPE	Returns the data type of the ith field in the server response. The EQ(<m>) and NE(<m>) functions, which return Boolean values to indicate the result of a comparison, are used with this prefix. For example, the following expression returns a Boolean TRUE if the DATATYPE function returns a value of datetime for the third field in the response: MSSQL.RES.FIELD(<2>).DATATYPE.EQ(datetime) **Parameters:** i - Row number
MSSQL.RES.FIELD(<i>).LENGTH	Returns the maximum possible length of the ith field in the server

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Description
	response. The value returned is a number. **Parameters:** i - Row number
MSSQL.RES.FIELD(<i>).NAME	Returns the name of the ith field in the server response. The value returned is of type text. **Parameters:** i - Row number
MSSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>)	Returns a value of type double from the jth column of the ith row of the table. If the value is not a double value, an UNDEF condition is raised. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the $(i + 1)^{th}$ row and the $(j + 1)^{th}$ column, respectively. **Parameters:** i - Row number j - Column number
MSSQL.RES.ROW(<i>).NUM_ELEM(j)	Returns an integer value from the jth column of ith row of the table. If the value is not an integer value, an UNDEF condition is raised. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the $(i + 1)^{th}$ row and the $(j + 1)^{th}$ column, respectively. **Parameters:** i - Row number j - Column number
MSSQL.RES.ROW(<i>).IS_NULL_ELEM(j)	Checks whether the jth column of the ith row of the table is NULL and returns a Boolean TRUE or FALSE to indicate the result. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the $(i + 1)^{th}$ row and the $(j + 1)^{th}$ column, respectively. **Parameters:** i - Row number j - Column number

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Description
MSSQL.RES.ROW(<i>).TEXT_ELEM(j)	Returns a text string from the jth column of ith row of the table. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the $(i + 1)^{th}$ row and the $(j + 1)^{th}$ column, respectively. **Parameters:** i - Row number j - Column number
MSSQL.RES.TYPE	Returns an enumeration constant that identifies the response type. Following are the possible return values: • ERROR • OK • RESULT_SET The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this expression.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Typecasting Data

May 14, 2012

You can extract data of one type (for example, text or an integer) from requests and responses and transform it to data of another type. For example, you can extract a string and transform the string to time format. You can also extract a string from an HTTP request body and treat it like an HTTP header or extract a value from one type of request header and insert it in a response header of a different type.

After typecasting the data, you can apply any operation that is appropriate for the new data type. For example, if you typecast text to an HTTP header, you can apply any operation that is applicable to HTTP headers to the returned value.

The following table describes various typecasting operations.

Table 1. Typecasting Functions

Function	Description
<text>.TYPECAST_LIST_T(<separator>)	Treats the text in an HTTP request or response body as a list whose elements are delimited by the character in the <separator> argument. Index values in the list that is created start with zero (0). Text mode settings have no effect on the separator. For example, even if you set the text mode to IGNORECASE, and the separator is the letter “p,” an uppercase “P” is not treated as a separator. The following example creates a Rewrite action that constructs a list from an HTTP request body and extracts the fourth item in the list: add rewrite action myreplace_action REPLACE 'http.req.body(100)' 'http.req.body(100).typecast_list_t('?').get(4)' set rewrite policy myreplace_policy -action myreplace_action This policy returns the string “fourth item” from the following request: GET?first item?second item?third item?fourth item? The following example extracts the fourth-from-last item from the list. add rewrite action myreplace_action1 REPLACE 'http.req.body(100)' 'http.req.body(100).typecast_list_t('?').get_reverse(4)' set rewrite policy myreplace_policy1 -action myreplace_action1 This policy returns the string “first item” from the following request: GET?first item?second item?third item?fourth item.
<text>.TYPECAST_ NVLIST_T(<separator>, <delimiter>) or text.TYPECAST_ NVLIST_T(<separator>, <delimiter>, <quote>)	Treats the text as a name-value list. The <separator> argument identifies the character and separates the name and the value. The <delimiter> argument identifies the character that separates each name-value pair. The <quote> character is required when typecasting text into a name-value list that supports quoted strings. Any delimiters that appear within the quoted string are ignored. The text mode has no effect on the delimiters. For example, if the current text mode is IGNORECASE and you specify “p” as the delimiter, an uppercase “P” is not treated as a delimiter. For example, the following policy counts the number of name-value pairs and inserts

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	the result in a header named name-value-count: add rewrite action mycount_action insert_http_header name-value-count 'http.req.header("Cookie").typecast_nvlist_t('=',';').count' set rewrite policy mycount_policy -action mycount_action This policy can extract a count of arguments in Cookie headers and insert the count in a name-value-count header: Cookie: name=name1; rank=rank1
<text>.TYPECAST_TIME_T	Treats the designated text as a date string. The following formats are supported: ● RFC822: Sun, 06 Nov 1994 08:49:37 GMT ● RFC850: Sunday, 06-Nov-94 08:49:37 GMT ● ASCII TIME: Sun Nov 6 08:49:37 1994 ● HTTP Set-Cookie Expiry date: Sun, 06-Nov-1994 08:49:37 GMT For example, the following policy converts the string to a time value and then extracts the day. This policy matches all requests that have a day value lesser than or equal to 10. Add rewrite policy mytime_policy "http.req.body(100) .typecast_time_t.day.le(10)" mytime_action bind rewrite global mytime_policy 100
<numeric string>.TYPECAST_IP_ ADDRESS_T	Treats a numeric string as an IP address. For example, the following policy matches HTTP requests that contains Cookie headers with a value of: 12.34.56.78\r\n. set rewrite policy ip_check_policy -rule 'http.req.cookie .value("ip").typecast_ip_address_t.eq(12.34.56.78)' bind rewrite global ip_check_policy 200 -type req_default
<numeric string>.TYPECAST_IPV6_ADDRESS_T	Treats a string as an IPv6 address in the following format: 0000:0000:CD00:0000:0000:00AB:0000:CDEF
<text>.TYPECAST_HTTP_ URL_T	Treats the designated text as the URL in the first line of an HTTP request header. The supported format is [<protocol>://<hostname>]<path>?<query>, and the text mode is set to URLENCODED by default. For example, the following policy replaces a URL-encoded part of a string in an HTTP header named Test. add rewrite action replace_header_string replace "http.req.header(\"Test\").typecast_http_url_t.path .before_str(\"123\").after_str(\"ABC\")" "\"string\"" add rewrite policy rewrite_test_header_policy true replace_header_string bind rewrite global rewrite_test_header_policy 1 END -type res_override Consider the following header:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	Test: ABC%12123\r\n This policy would replace the preceding header with the value ABC%string123\r\n.
<text>.TYPECAST_HTTP_ HOSTNAME_T	Provides operations for parsing an HTTP host name as it appears in HTTP data. The format for a host name is abc.def.com:8080.
<text>.TYPECAST_HTTP_ METHOD_T	Converts text to an HTTP method. For example, the following policy matches any HTTP request that contains a Host header with a value equal to POST: Add rewrite policy method_policy "http.req.header(\"Host\") .typecast_http_method_t.eq(POST)" act1
<text>.TYPECAST_DNS_ DOMAIN_T	Enables the designated text to be parsed like a DNS domain name in the format ab.def.com.
<text>.TYPECAST_HTTP_ HEADER_T("<name>")	Converts the designated text to a multi-line HTTP header that you specify in a <name> argument. For example, the following expression converts “MyHeader” to “InHeader”: http.req.header("MyHeader").typcast_http_header_t("InHeader") Typically, text operations that you specify in this type of expression apply to only the last line of this header, with some exceptions. For example, the CONTAINS operation operates on values in all the lines in instances of this header type.
<text>.TYPECAST_COOKIE_T	Treats the designated text as an HTTP cookie as it appears in a Set-Cookie or Set-Cookie2 header. You can apply name-value list operations as well as text operations to the designated text. For example, you can designate equals (=) as the name-value delimiter and the semicolon (;) as the list element delimiter. If you apply name-value list operations, the list is parsed as if IGNORE_EMPTY_ELEMENTS were in effect. Each cookie begins with a cookie-name=cookie-value pair, optionally followed by attribute-value pairs that are separated by a semicolon, as follows: cookie1=value1;version=n.n;value;domain=value;path=value If the same attribute appears more than once in a cookie, the value for the first instance of the attribute is returned.
<number>.TYPECAST_DOUBLE_AT	Transforms the number to a value of data type double.
<number>.TYPECAST_IP_ADDRESS_AT	Converts the number to an IP address.
<number>.TYPECAST_TIME_AT	Converts the number to time format.
<number>.TYPECAST_TIME_AT.BETWEEN(<time1>, <time2>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is between the lower and upper time value arguments <time1> and <time2>.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	The following are prerequisites for this function: ● Both the lower and upper time arguments must be fully specified. For example, GMT 1995 Jan is fully specified. But GMT Jan, GMT 1995 20 and GMT Jan Mon_2 are not fully specified. ● Both arguments must be either GMT or Local. ● The day of the week must not be present in either argument. However, the day of the month can be specified as the first, second, third, or fourth weekday of the month (example Wed_3 is the third Wednesday of the month). ● The upper time argument, <time2>, must be bigger than the lower time argument, <time1>. The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day is the first Sunday of the month of May in 2005. The result of the evaluation is given after each example. BETWEEN(GMT 2004, GMT 2006): TRUE BETWEEN(GMT 2004 Jan, GMT 2006 Nov): TRUE BETWEEN(GMT 2004 Jan, GMT 2006): TRUE BETWEEN(GMT 2005 May Sun_1, GMT 2005 May Sun_3): TRUE BETWEEN(GMT 2005 May 1, GMT May 2005 1): TRUE BETWEEN(LOCAL 2005 May 1, LOCAL May 2005 1): The result depends on the NetScaler system's timezone. Parameters: <time1> - Lower time value <time2> - Upper time value
<number>.TYPECAST_TIME_AT.DAY	Extracts the day of the month from the current system time and returns the value as a number that corresponds to the day of the month. The returned value ranges from 1 to 31.
<number>.TYPECAST_TIME_AT.EQ(<t>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is equal to the time value argument <t>. The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day is the 1st Sunday of the month of May in 2005. The result of the evaluation is given after each example. EQ(GMT 2005): TRUE EQ(GMT 2005 Dec): FALSE EQ(Local 2005 May): TRUE or FALSE, depending on the time zone. EQ(GMT 10h): TRUE EQ(GMT 10h 30s): TRUE EQ(GMT May 10h): TRUE EQ(GMT Sun): TRUE EQ(GMT May Sun_1): TRUE Parameters: <t> - Time
<number>.TYPECAST_TIME_AT.GE(<t>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is greater than or equal to the time value argument <t>. The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day is the 1st Sunday of the month of May in 2005. The result of

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	the evaluation is given after each example. GE(GMT 2004): TRUE GE(GMT 2005 Jan): TRUE GE(Local 2005 May): TRUE or FALSE, depending on the time zone. GE(GMT 8h): TRUE GE(GMT 30m): FALSE GE(GMT May 10h): TRUE GE(GMT May 10h 0m): TRUE GE(GMT Sun): TRUE GE(GMT May Sun_1): TRUE Parameters: <t> - Time
<number>.TYPECAST_TIME_AT.GT(<t>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is greater than the time value argument <t>. The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day is the 1st Sunday of the month of May in 2005. The result of the evaluation is given after each example. GT(GMT 2004): TRUE GT(GMT 2005 Jan): TRUE GT(Local 2005 May): TRUE or FALSE, depending on the time zone. GT(GMT 8h): TRUE GT(GMT 30m): FALSE GT(GMT May 10h): FALSE GT(GMT May 10h 0m): TRUE GT(GMT Sun): FALSE GT(GMT May Sun_1): FALSE Parameters: <t> - Time
<number>.TYPECAST_TIME_AT.HOURS	Extracts the hour from the current system time and returns the corresponding value as an integer that can range from 0 to 23.
<number>.TYPECAST_TIME_AT.LE(<t>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is lesser than or equal to the time value argument <t>. The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day is the 1st Sunday of the month of May in 2005. The result of the evaluation is given after each example. LE(GMT 2006): TRUE LE(GMT 2005 Dec): TRUE LE(Local 2005 May): TRUE or FALSE, depending on the time zone. LE(GMT 8h): FALSE LE(GMT 30m): TRUE LE(GMT May 10h): TRUE LE(GMT Jun 11h): TRUE LE(GMT Wed): TRUE LE(GMT May Sun_1): TRUE Parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
<number>.TYPECAST_TIME_AT.LT(<t>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> is lesser than the time value argument <t>. The following examples assume that the current time value is GMT 2005 May 1 10h 15m 30s and that the day is the 1st Sunday of the month of May in 2005. The result of the evaluation is given after each example. LT(GMT 2006): TRUE LT(GMT 2005 Dec): TRUE LT(Local 2005 May): TRUE or FALSE, depending on the time zone. LT(GMT 8h): FALSE LT(GMT 30m): TRUE LT(GMT May 10h): FALSE LT(GMT Jun 11h): TRUE LT(GMT Wed): TRUE LT(GMT May Sun_1): FALSE Parameters: <t> - Time
<number>.TYPECAST_TIME_AT.MINUTES	Extracts the minute from the current system time and returns the value as an integer that can range from 0 to 59.
<number>.TYPECAST_TIME_AT.MONTH	Extracts the month from the current system time and returns the value as an integer that can range from 1 (January) to 12 (December).
<number>.TYPECAST_TIME_AT.RELATIVE_BOOT	Calculates the number of seconds that have elapsed after the most recent reboot or the number of seconds to the next scheduled reboot, depending on which is closer to the current time, and returns an integer. If the closest boot time is in the past, the integer is negative. If the closest boot time is in the future (scheduled reboot time), the integer is positive.
<number>.TYPECAST_TIME_AT.RELATIVE_NOW	Calculates the number of seconds between the current system time and the specified time, and returns the value as an integer. If the designated time is in the past, the integer is negative. If it is in the future, the integer is positive.
<number>.TYPECAST_TIME_AT.SECONDS	Extracts the seconds from the current system time and returns the value as an integer that can range from 0 to 59.
<number>.TYPECAST_TIME_AT.WEEKDAY	Returns an integer that corresponds to the day of the week; 0 for Sunday and 6 for Saturday.
<number>.TYPECAST_TIME_AT.WITHIN(<time1>, <time2>)	Returns a Boolean value (TRUE or FALSE) that indicates whether the time value designated by <number> lies within all the ranges defined by lower and upper time value arguments <time1> and <time2>. If an element of time such as the day or the hour is left unspecified in the lower argument, <time1>, then it is assumed to have the lowest value possible for its range. If an element is left unspecified in the upper argument , <time2>, then it is assumed to have the highest value possible for its range.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	If the year is specified in one of the arguments, then it must be specified in the other argument as well. Following are the ranges for different elements of time: ● month: 1-12 ● day: 1-31 ● weekday: 0-6 ● hour: 0-23 ● minutes: 0-59 ● seconds: 0-59. Each element of time in the lower time value argument defines a range in combination with the corresponding element in the upper time value argument. For the result to be TRUE, each element of time in the time value designated by <number> must lie in the corresponding range specified by the lower and upper arguments. The following examples assume that the current time value is GMT 2005 May 10 10h 15m 30s.and that the day is the second Tuesday of the month. The result of the evaluation is given after each example. WITHIN(GMT 2004, GMT 2006): TRUE WITHIN(GMT 2004 Jan, GMT 2006 Mar): FALSE (May doesn't fall in the Jan-Mar range.) WITHIN(GMT Feb, GMT): TRUE (May falls in the Feb-Dec range.) WITHIN(GMT Sun_1, GMT Sun_3): TRUE (2nd Tuesday lies within 1st Sunday and the 3rd Sunday.) WITHIN(GMT 2005 May 1 10h, GMT May 2005 1 17h): TRUE WITHIN(LOCAL 2005 May 1, LOCAL May 2005 1): The result depends on the NetScaler system's timezone. Parameters: <time1> - Lower time value <time2> - Upper time value
<number>.TYPECAST_TIME_AT.YEAR	Extracts the year from the current system time and returns the value as a four-digit integer.
<prefix>.TYPECAST_NUM_T(<type>)	Casts numeric string data to a signed 32-bit number. The argument <type> can be one of the following: ● DECIMAL. Treat the string as a decimal number and cast to a signed 32-bit number. ● HEX. Treat the string as a hexadecimal number and cast to a signed 32-bit number. ● DECIMAL_PREFIX. Consider the part of the string up to the first occurrence of a character that is not a valid decimal character and cast to a signed 32-bit number. ● HEX_PREFIX. Consider the part of the string up to the first occurrence of a character that is not a valid hexadecimal character and cast to a signed 32-bit number. For example, the following policy extracts a numeric portion of a query string, adds 4 to the number, and inserts an HTTP header named Company with the resulting decimal value. add rewrite action myadd_action insert_http_header Company "http.req.url.query.typecast_num_t(decimal).add(4)"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	add rewrite policy myadd_policy true myadd_action bind rewrite global myadd_policy 300 END -type RES_DEFAULT For example, this policy would extract “4444” from the following URL stub: /test/file.html?4444 The action that is associated with the policy would insert the following HTTP response header: Company: 4448\r\n
<prefix>.TYPECAST_NUM_AT	Casts a number of any data type to a number of data type integer.
<prefix>.TYPECAST_DOUBLE_AT	Casts a number of any data type to a number of data type double.
<prefix>.TYPECAST_UNSIGNED_LONG_AT	Casts a number of any data type to a number of data type unsigned long.
<prefix>.TYPECAST_NUM_T(<type>,<default>)	Casts string data to a signed 32-bit number. If the typecasting operation raises an undefined (UNDEF) condition, the function returns the value specified for default. The type argument takes the values specified for TYPECAST_NUM_T(<type>).
<prefix>.TYPECAST_UNSIGNED_LONG_T(<type>)	Casts string data to data of type unsigned long. The argument can be one of the following: ● DECIMAL. Treat the string as a decimal number and cast to unsigned long. ● HEX. Treat the string as a hexadecimal number and cast to unsigned long. ● DECIMAL_PREFIX. Consider the part of the string up to the first occurrence of a character that is not a valid decimal character and cast to unsigned long. ● HEX_PREFIX. Consider the part of the string up to the first occurrence of a character that is not a valid hexadecimal character and cast to unsigned long.
<prefix>.TYPECAST_UNSIGNED_LONG_T(<type>, <default>)	Casts string data to data of type unsigned long. If the typecasting operation raises an undefined (UNDEF) condition, the function returns the value specified for default. The type argument takes the values specified for TYPECAST_UNSIGNED_LONG_T(<type>).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Regular Expressions

Sep 02, 2013

When you want to perform string matching operations that are more complex than the operations that you perform with the CONTAINS(“<string>”) or EQ(“<string>“) operators, you use regular expressions. The policy infrastructure on the Citrix® NetScaler® appliance includes operators to which you can pass regular expressions as arguments for text matching. The names of the operators that work with regular expressions include the string REGEX. The regular expressions that you pass as arguments must conform to the regular expression syntax that is described in "http://www.pcre.org/pcre.txt." You can learn more about regular expressions at "http://www.regular-expressions.info/quickstart.html" and at "http://www.silverstones.com/thebat/Regex.html."

The target text for an operator that works with regular expressions can be either text or the value of an HTTP header. Following is the format of a default syntax expression that uses a regular expression operator to operate on text:

<text>.<regex_operator>(re<delimiter><regex_pattern><delimiter>)

The string <text> represents the default syntax expression prefix that identifies a text string in a packet (for example, HTTP.REQ.URL). The string <regex_operator> represents the regular expression operator. The regular expression always begins with the string re. A pair of matching delimiters, represented by <delimiter>, enclose the string <regex_pattern>, which represents the regular expression.

The following example expression checks whether the URL in an HTTP packet contains the string *.jpeg (where * is a wildcard) and returns a Boolean TRUE or FALSE to indicate the result. The regular expression is enclosed within a pair of slash marks (/), which act as delimiters.

http.req.url.regex_match(re/*.jpeg/)

Regular expression operators can be combined to define or refine the scope of a search. For example, <text>.AFTER_REGEX(re/regex_pattern1/).BEFORE_REGEX(re/regex_pattern2/) specifies that the target for string matching is the text between the patterns regex_pattern1 and regex_pattern2. You can use a text operator on the scope that is defined by the regular expression operators. For example, you can use the CONTAINS(“<string>”) operator to check whether the defined scope contains the string abc:

<text>.AFTER_REGEX(re/regex_pattern1/).BEFORE_REGEX(re/regex_pattern2/).CONTAINS(“abc”)

Note: The process of evaluating a regular expression inherently takes more time than that for an operator such as CONTAINS(“<string>”) or EQ(“<string>”), which work with simple string arguments. You should use regular expressions only if your requirement is beyond the scope of other operators.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Basic Characteristics of Regular Expressions

Jul 10, 2013

Following are notable characteristics of regular expressions as defined on the NetScaler appliance:

- A regular expression always begins with the string “re” followed by a pair of delimiting characters (called delimiters) that enclose the regular expression that you want to use.
  For example, re#<regex_pattern># uses the number sign (#) as a delimiter.
- A regular expression cannot exceed 1499 characters.
- Digit matching can be done by using the string \d (a backslash followed by d).
- White space can be represented by using \s (a backslash followed by s).
- A regular expression can contain white spaces.

Following are the differences between the NetScaler syntax and the PCRE syntax:

- The NetScaler does not allow back references in regular expressions.
- You should not use recursive regular expressions.
- The dot meta-character also matches the newline character.
- Unicode is not supported.
- The operation SET_TEXT_MODE(IGNORECASE) overrides the (?i) internal option in the regular expression.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations for Regular Expressions

Jul 10, 2013

The following table describes the operators that work with regular expressions. The operation performed by a regular expression operator in a given default syntax expression depends on whether the expression prefix identifies text or HTTP headers. Operations that evaluate headers override any text-based operations for all instances of the specified header type. When you use an operator, replace <text> with the default syntax expression prefix that you want to configure for identifying text.

Table 1. Default Syntax Expression Operators That Work with Regular Expressions

Regular Expression Operation	Description
<text>.BEFORE_REGEX(<regular expression>)	Selects the text that precedes the string that matches the <regular expression> argument. If the regular expression does not match any data in the target, the expression returns a text object of length 0. The following expression selects the string "text" from "text/plain". http.res.header("content-type").before_regex(re#/#)
<text>.AFTER_REGEX(<regular expression>)	Selects the text that follows the string that matches the <regular expression> argument. If the regular expression does not match any text in the target, the expression returns a text object of length 0. The following expression extracts "Example" from "myExample": http.req.header("etag").after_regex(re/my/)
<text>.REGEX_SELECT(<regular expression>)	Selects a string that matches the <regular expression> argument. If the regular expression does not match the target, a text object of length 0 is returned. The following example extracts the string "NS-CACHE-9.0: 90" from a Via header: http.req.header("via").regex_select(re!NS-CACHE-\d\.\d:\s*\d{1,3}!)
<text>.REGEX_MATCH(<regular expression>)	Returns TRUE if the target matches a <regular expression> argument of up to 1499 characters. The regular expression must be of the following format: re<delimiter>regular expression< delimiter> Both delimiters must be the same. Additionally, the regular expression must conform to the Perl-compatible (PCRE) regular expression library syntax. For more information, go to http://www.pcre.org/pcre.txt. In particular, see the pcrepattern man page. However, note the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Regular Expression Operation	Description
	• Back-references are not allowed. • Recursive regular expressions are not recommended. • The dot metacharacter also matches the newline character. • The Unicode character set is not supported. • SET_TEXT_MODE(IGNORECASE) overrides the (?i) internal option specified in the regular expression. The following are examples: http.req.hostname.regex_match(re/[[:alpha:]]+(abc){2,3}/) http.req.url.set_text_mode(urlencoded).regex_match(re#(a*b+c*)#) The following example matches ab and aB: http.req.url.regex_match(re/a(?i)b/) The following example matches ab, aB, Ab and AB: http.req.url.set_text_mode(ignorecase).regex_match(re/ab/) The following example performs a case-insensitive, multiline match in which the dot meta-character also matches a newline character: http.req.body.regex_match(re/(?ixm) (^ab (.*) cd$) /)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Classic Policies and Expressions

May 25, 2015

Some NetScaler features use classic policies and classic expressions. As with default syntax policies, classic policies can be either global or specific to a virtual server. However, to a certain extent, the configuration method and bind points for classic policies are different from those of default syntax policies. As with default syntax expressions, you can configure named expressions and use a named expression in multiple classic policies.

The following table summarizes NetScaler features that can be configured by using classic policies.

Table 1. Policy Type and Bind Points for Policies in Features That Use Classic Policies

Feature	Virtual Servers	Supported Policies	Policy Bind Points	How You Use the Policies
System features, Authentication	None	Authentication policies	Global	For the Authentication feature, policies contain authentication schemes for different authentication methods. For example, you can configure LDAP and certificate-based authentication schemes.
SSL	None	SSL policies	• Global • Load Balancing virtual server	To determine when to apply an encryption function and add certificate information to clear text. To provide end-to-end security. After a message is decrypted, the SSL feature re-encrypts clear text and uses SSL to communicate with back-end Web servers.
Content Switching (Can use either classic or default syntax policies, but not both)	Content Switching virtual server	Content Switching policies	• Content Switching virtual server • Cache Redirection virtual server	To determine what server or group of servers is responsible for serving responses, based on characteristics of an incoming request. Request characteristics include device type, language, cookies, HTTP method, content type and associated cache server.
Compression	None	HTTP Compression policies	• Global • Content Switching virtual server • Load Balancing virtual server	To determine what type of HTTP traffic is compressed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature	Virtual Servers	Supported Policies	Policy Bind Points	How You Use the Policies
			• SSL Offload virtual server • Service	
Protection features, Filter	None	Content Filtering policies	• Global • Content Switching virtual server • Load Balancing virtual server • SSL Offload virtual server • Service	To configure the behavior of the filter function.
Protection features, SureConnect	None	SureConnect policies	• Load Balancing virtual server • SSL Offload virtual server • Service	To configure the behavior of the SureConnect function.
Protection features, Priority Queuing	None	Priority Queuing policies	• Load Balancing virtual server • SSL Offload virtual server	To configure the behavior of the Priority Queuing function.
HTML Injection	None	HTML Injection Policies	• Global • Load Balancing virtual server • Content Switching virtual server • SSL Offload virtual server	To enable the NetScaler to insert text or scripts into an HTTP response that it serves to a client.
AAA - Traffic Management	None	Authentication, Authorization, Auditing, and Session policies	• Authentication virtual server (authentication, session, and auditing policies) • Load Balancing or Content Switching virtual server (authorization	To configure rules for user access to specific sessions and auditing of user access.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature	Virtual Servers	Supported Policies	Policy Bind Points	How You Use the Policies
			and auditing policies) • Global (session and audit policies) • AAA group or user (session, auditing, and authorization policies)	
Cache Redirection	Cache Redirection virtual server	Cache Redirection policies Map policies	Cache Redirection virtual server	To determine whether HTTP responses are served from a cache or an origin server.
Application firewall	None	Application firewall policies	Global	To identify characteristics of traffic and data that should or should not be admitted through the firewall.
NetScaler Gateway	VPN server	Pre-Authentication policies	• AAA Global • VPN vserver	To determine how the NetScaler Gateway performs authentication, authorization, auditing, and other functions, and to define rewrite rules for general Web access using the NetScaler Gateway.
		Authentication policies	• System Global • AAA Global • VPN vserver	
		Auditing policies	• User • User group • VPN vserver	
		Session policies	• VPN Global • User • User Group • VPN vserver	
		Authorization policies	• User • User Group	
		Traffic policies	• VPN Global • User	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature	Virtual Servers	Supported Policies	Policy Bind Points	How You Use the Policies
			• User Group • VPN vserver	
		TCP Compression policies	VPN Global	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Classic Policy

Oct 29, 2013

You can configure classic policies and classic expressions by using either the configuration utility or the command-line interface. A policy rule cannot exceed 1,499 characters. When configuring the policy rule, you can use named classic expressions. For more information about named expressions, see "Creating Named Classic Expressions." After configuring the policy, you bind it either globally or to a virtual server.

Note that there are small variations in the policy configuration methods for various NetScaler features.

Note: You can embed a classic expression in a default syntax expression by using the syntax SYS.EVAL_CLASSIC_EXPR**(classic_expression)**, specifying the classic_expression as the argument.

To create a classic policy by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add cmp policy <name> -rule <expression> -action <action>
- show cmp policy [<policyName>]

Example

The following commands first create a compression action and then create a compression policy that applies the action:

```
> add cmp action cmp-act-compress compress
 Done
> show cmp action cmp-act-compress
1)     Name: cmp-act-compress  Compression Type: compress
 Done
> add cmp pol cmp-pol-compress -rule ExpCheckIp -resAction cmp-act-compress
 Done
> show cmp pol cmp-pol-compress
1)     Name: cmp-pol-compress  Rule: ExpCheckIp
       Response action: cmp-act-compress       Hits: 0
 Done
>
```

To create a policy with classic expressions by using the configuration utility

1. In the navigation pane, expand the feature for which you want to configure a policy and, depending on the feature, do the following:
   - For Content Switching, Cache Redirection, and the application firewall, click Policies.
   - For SSL, click Policies, and then in the details pane, click the Policies tab.
   - For System Authentication, click Authentication, and then in the details pane, click the Policies tab.
   - For Filter, SureConnect, and Priority Queuing, expand Protection Features, select the desired function, and then in the details pane, click the Policies tab.
   - For the NetScaler Gateway, expand NetScaler Gateway, expand Policies, select the desired function, and then in the details pane, click the Policies tab.
2. For most features, click the Add button.
3. In the Create <feature name> Policy dialog box, in the Name* text box, enter a name for the policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: Note: You must begin a policy name with a letter or underscore. A policy name can consist of 1 to 31 characters, including letters, numbers, hyphen (-), period (.), pound sign (#), space ( ), and underscore (_).

4. For most features, you associate an action or a profile. For example, you may be required to select an action, or, in the case of an NetScaler Gateway or application firewall policy, you select a profile to associate with the policy. A profile is a set of configuration options that operate as a set of actions that are applied when the data being analyzed matches the policy rule.
5. Create an expression that describes the type of data that you want this policy to match.

   Depending on the type of policy you want to create, you can choose a predefined expression, or you can create a new expression. For instructions on how to create an expression for most types of classic policies, see "Configuring a Classic Expression."

   Named expressions are predefined expressions that you can reference by name in a policy rule. For more information about named expressions, see "Creating Named Classic Expressions." For a list of all the default named expressions and a definition of each, see "Expressions Reference."

6. Click Create to create your new policy.
7. Click Close to return to the Policies screen for the type of policy you were creating.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Classic Expression

Oct 29, 2013

Classic expressions consist of the following expression elements, listed in hierarchical order:

- **Flow Type.** Specifies whether the connection is incoming or outgoing. The flow type is REQ for incoming connections and RES for outgoing connections.
- **Protocol.** Specifies the protocol, the choices for which are HTTP, SSL, TCP, and IP.
- **Qualifier.** The protocol attribute, which depends on the selected protocol.
- **Operator.** The type of test you want to perform on the connection data. Your choice of operator depends upon the connection information you are testing. If the connection information you are testing is text, you use text operators. If it is a number, you use standard numeric operators.
- **Value.** The string or number against which the connection data element—defined by the flow type, protocol, and qualifier—is tested. The value can be either a literal or an expression. The literal or expression must match the data type of the connection data element.

In a policy, classic expressions can be combined to create more complex expressions using Boolean and comparative operators.

Expression elements are parsed from left to right. The leftmost element is either REQ or RES and designates a request or a response, respectively. Successive terms define a specific connection type and a specific attribute for that connection type. Each term is separated from any preceding or following term by a period. Arguments appear in parentheses and follow the expression element to which they are passed.

The following classic expression fragment returns the client source IP for an incoming connection.

```
REQ.IP.SOURCEIP
```

The example identifies an IP address in a request. The expression element SOURCEIP designates the source IP address. This expression fragment may not be useful by itself. You can use an additional expression element, an operator, to determine whether the returned value meets specific criteria. The following expression tests whether the client IP is in the subnet 200.0.0.0/8 and returns a Boolean TRUE or FALSE:

```
REQ.IP.SOURCEIP == 200.0.0.0 -netmask 255.0.0.0
```

To create a classic policy expression by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- set appfw policy <name> -rule <expression> -action <action>
- show appfw policy <name>

Example

```
> set appfw policy GenericApplicationSSL_ 'HTTP.REQ.METHOD.EQ("get")' APPFW_DROP
 Done
> show appfw policy GenericApplicationSSL_
      Name: GenericApplicationSSL_    Rule: HTTP.REQ.METHOD.EQ("get")
      Profile: APPFW_DROP     Hits: 0
      Undef Hits: 0
      Policy is bound to following entities
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
1) REQ VSERVER app_u_GenericApplicationSSLPortalPages   PRIORITY : 100
Done
```

To add an expression for a classic policy by using the configuration utility

This procedure documents the Add Expression dialog box. Depending on the feature for which you are configuring a policy, the route by which you arrive at this dialog box may be different.

1. Perform steps 1-4 in "To create a policy with classic expressions by using the configuration utility."
2. In the Add Expression dialog box, in Expression Type, click the type of expression you want to create.
3. Under Flow Type, click the down arrow and choose a flow type.

   The flow type is typically REQ or RES. The REQ option specifies that the policy applies to all incoming connections or requests. The RES option applies the policy to all outgoing connections or responses.

   For Application Firewall policies, you should leave the expression type set to General Expression, and the flow type set to REQ. The Application Firewall treats each request and response as a single paired entity, so all Application Firewall policies begin with REQ.
4. Under Protocol, click the down arrow and choose the protocol you want for your policy expression. Your choices are:
   - HTTP. Evaluates HTTP requests that are sent to a Web server. For classic expressions, HTTP includes HTTPS requests.
   - SSL. Evaluates SSL data associated with the current connection.
   - TCP. Evaluates the TCP data associated with the current connection.
   - IP. Evaluates the IP addresses associated with the current connection.
5. Under Qualifier, click the down arrow and choose a qualifier for your policy.

   The qualifier defines the type of data to be evaluated. The list of qualifiers that appears depends on which protocol you selected in step 4.

   The following list describes the qualifier choices for the HTTP protocol. For a complete list of protocols and qualifiers, see "Classic Expressions."

   The following choices appear for the HTTP protocol:

   - METHOD. Filters HTTP requests that use a particular HTTP method.
   - URL. Filters HTTP requests for a specific Web page.
   - URLQUERY. Filters HTTP requests that contain a particular query string.
   - VERSION. Filters HTTP requests on the basis of the specified HTTP protocol version.
   - HEADER. Filters on the basis of a particular HTTP header.
   - URLLEN. Filters on the basis of the length of the URL.
   - URLQUERY. Filters on the basis of the query portion of the URL.
   - URLQUERYLEN. Filters on the basis of the length of the query portion of the URL only.
6. Under Operator, click the down arrow and choose the operator for your policy expression. For a complete list of choices see the “Operators” table in "Classic Expressions." Some common operators are:

Operator	Description
==	Matches the specified value exactly or is exactly equal to the specified value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operator	Description
!=	Does not match the specified value.
>	Is greater than the specified value.
<	Is less than the specified value.
>=	Is greater than or equal to the specified value.
<=	Is less than or equal to the specified value.
CONTAINS	Contains the specified value.
CONTENTS	Returns the contents of the designated header, URL, or URL query.
EXISTS	The specified header or query exists.
NOTCONTAINS	Does not contain the specified value.
NOTEXISTS	The specified header or query does not exist.

7. If a Value text box appears, type a string or numeric value, as appropriate. For example, chose REQ as the Flow Type, HTTP as the Protocol, and HEADER as the qualifier, and then type the value of the header string in the Value field and the header type for which you want to match the string in the Header Name text box.
8. Click OK.
9. To create a compound expression, click Add. Note that the type of compounding that is done depends on the following choices in the Create Policy dialog box:
   - **Match Any Expression.** The expressions are in a logical OR relationship.
   - **Match All Expressions.** The expressions are in a logical AND relationship.
   - **Tabular Expressions.** Click the AND, OR, and parentheses buttons to control evaluation.
   - **Advanced Free-Form.** Enter the expressions components directly into the Expression field, and click the AND, OR, and parentheses buttons to control evaluation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding a Classic Policy

Oct 29, 2013

Depending on the policy type, you can bind a classic policy either globally or to a virtual server. Policy bind points are described in the table, "Policy Type and Bind Points for Policies in Features That Use Classic Policies."

Note: You can bind a classic policy to multiple bind points.

To bind a classic policy globally by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- bind cmp global <policyName> [-priority <positive_integer>]
- show cmp global

Example

```
> bind cmp global cmp-pol-compress -priority 2
 Done
> show cmp global
1)    Policy Name: cmp-pol-compress   Priority: 2
2)    Policy Name: ns_nocmp_xml_ie    Priority: 8700
3)    Policy Name: ns_nocmp_mozilla_47       Priority: 8800
4)    Policy Name: ns_cmp_mscss       Priority: 8900
5)    Policy Name: ns_cmp_msapp       Priority: 9000
6)    Policy Name: ns_cmp_content_type       Priority: 10000
 Done
>
```

To bind a classic policy to a virtual server by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- bind lb vserver <name> [<targetVserver>] [-policyName <string> [-priority <positive_integer>]
- show lb vserver<name>

Example

```
> bind lb vserver lbtemp -policyName cmp-pol-compress -priority 1
 Done
> show lb vserver lbtemp
      lbtemp (10.102.29.101:80) - HTTP       Type: ADDRESS
      State: UP
      Last state change was at Tue Oct 27 06:40:38 2009 (+557 ms)
      Time since last state change: 0 days, 02:00:40.330
      Effective State: UP
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
      No. of Bound Services :  1 (Total)      1 (Active)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Configured Method: LEASTCONNECTION
    Current Method: Round Robin, Reason: Bound service's state changed to UP
    Group: vserver-grp
    Mode: IP
    Persistence: COOKIEINSERT (version 0)   Persistence Backup: SOURCEIP   Persistence Mask: 255.255.255.255
    Persistence Timeout: 2 min      Backup Persistence Timeout: 2 min
    Vserver IP and Port insertion: OFF
    Push: DISABLED  Push VServer:
    Push Multi Clients: NO
    Push Label Rule: none
1) http-one (10.102.29.252: 80) - HTTP State: UP        Weight: 1
    Persistence Cookie Value : NSC_wtfswfs-hsq=ffffffff096e03ed45525d5f4f58455e445a4a423660
1)     Policy : cmp-pol-compress Priority:1
 Done
>
```

To bind a classic policy globally by using the configuration utility

Note: This procedure documents the Global Bindings dialog box. Depending on the feature for which you want to globally bind a policy, the route by which you arrive at this dialog box may be different.

1. In the navigation pane, expand the feature for which you want to globally bind a classic policy, and then locate the policy that you want to bind globally.

   Note: You cannot globally bind policies for Content Switching, Cache Redirection, SureConnect, Priority Queuing, or NetScaler Gateway Authorization.
2. In the details pane, click Global Bindings.
3. In the Bind/Unbind <feature name> Policy(s) to Global dialog box, click Insert Policy.
4. In the Policy Name column, click the name of an existing policy that you want to globally bind, or click New Policy to open the Create <feature name> Policy dialog box.
5. After you have selected the policy or created a new policy, in the Priority column, type the priority value.

   The lower the number, the sooner this policy is applied relative to other policies. For example, a policy assigned a priority of 10 is applied before a policy with a priority of 100. You can use the same priority for different policies. All features that use classic policies implement only the first policy that a connection matches, so policy priority is important for getting the results you intend.

   As a best practice, leave room to add policies by setting priorities with intervals of 50 (or 100) between each policy.

6. Click OK.

To bind a classic policy to a virtual server by using the configuration utility

1. In the navigation pane, expand the feature that contains the virtual server to which you want to bind a classic policy (for example, if you want to bind a classic policy to a content switching virtual server, expand Traffic Management > Content Switching), and then click Virtual Servers.
2. In the details pane, select the virtual server, and then click Open.
3. In the Configure <Feature> Virtual Server dialog box, on the Policies tab, click the feature icon for the type policy that you want, and then click Insert Policy.
4. In the Policy Name column, click the name of an existing policy that you want to bind to a virtual server, or click A to open the Create <feature name> Policy dialog box.
5. After you have selected the policy or created a new policy, in the Priority column, set the priority.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

If you are binding a policy to a content switching virtual server, in the Target column, select a load balancing virtual server to which traffic that matches the policy should be sent.

6. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing Classic Policies

Oct 29, 2013

You can view classic policies by using either the configuration utility or the command line. You can view details such as the policy's name, expression, and bindings.

To view a classic policy and its binding information by using the command line interface

At the command prompt, type the following commands to view a classic policy and its binding information:

show <featureName> policy [policyName]

Example

```
> show appfw policy GenericApplicationSSL_
    Name: GenericApplicationSSL_    Rule: ns_only_get_adv
    Profile: GenericApplicationSSL_Prof1    Hits: 0
    Undef Hits: 0
    Policy is bound to following entities
    1) REQ VSERVER app_u_GenericApplicationSSLPortalPages   PRIORITY : 100
 Done
```

Note: If you omit the policy name, all policies are listed without the binding details.

To view classic policies and policy bindings by using the configuration utility

1. In the navigation pane, expand the feature whose policies you want to view, (for example, if you want to view application firewall policies, expand Application Firewall), and then click Policies.
2. In the details pane, do one or more of the following:
   - To view details for a specific policy, click the policy. Details appear in the Details area of the configuration pane.
   - To view bindings for a specific policy, click the policy, and then click Show Bindings.
   - To view global bindings, click the policy, and then click Global Bindings. Note that you cannot bind a Content Switching, Cache Redirection, SureConnect, Priority Queuing, or NetScaler Gateway Authorization policy globally.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating Named Classic Expressions

Sep 30, 2013

A named classic expression is a classic expression that can be referenced through an assigned name. Often, you need to configure classic expressions that are large or complex and form a part of a larger compound expression. You might also configure classic expressions that you need to use frequently and in multiple compound expressions or classic policies. In these scenarios, you can create the classic expression you want, save it with a name of your choice, and then reference the expression from compound expressions or policies through its name. This saves configuration time and improves the readability of complex compound expressions. Additionally, any modifications to a named classic expression need to be made only once.

Some named expressions are built-in, and a subset of these are read-only. Built-in named expressions are divided into four categories: General, Anti-Virus, Personal Firewall, and Internet Security. General named expressions have a wide variety of uses. For example, from the General category, you can use the expressions ns_true and ns_false to specify a value of TRUE or FALSE, respectively, to be returned for all traffic. You can also identify data of a particular type (for example, HTM, DOC, or GIF files), determine whether caching headers are present, or determine whether the round trip time for packets between a client and the NetScaler is high (over 80 milliseconds).

Anti-Virus, Personal Firewall, and Internet Security named expressions test clients for the presence of a particular program and version and are used primarily in NetScaler Gateway policies.

For descriptions of the built-in named expressions, see "Classic Expressions."

Note: You cannot modify or delete built-in named expressions.

To create a named classic expression by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- add expression <name> <value> [-comment <string>] [-clientSecurityMessage <string>]
- show expression [<name> | -type CLASSIC

Example

```
> add expression classic_ne "REQ.HTTP.URL CONTAINS www.example1.com" -comment "Checking the URL for www.example1.com"
 Done
> show expression classic_ne
1)     Name: classic_ne  Expr: REQ.HTTP.URL CONTAINS www.example1.com  Hits: 0 Type : CLASSIC
      Comment: "Checking the URL for www.example1.com"
 Done
>
```

To create a named classic expression by using the configuration utility

1. In the navigation pane, expand AppExpert, expand Expressions, and then click Classic Expressions.
2. In the details pane, click Add.
   Note: Some of the built-in expressions in the Expressions list are read-only.

3. In the Create Policy Expression dialog box, specify values for the following parameters:
   - Expression Name*—name
   - Client Security Message—clientSecurityMessage
   - Comments—comment

   * A required parameter

4. To create the expression, do one of the following:
   - You can choose inputs to this expression from the Named Expressions drop-down list.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- You can create a new expression, as described in "To add an expression for a classic policy by using the configuration utility."

5. When you are done, click Close. Verify that your new expression was created by scrolling to the bottom of the Classic Expressions list to view it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions Reference-Default Syntax Expressions

May 26, 2015

The following table is a listing of default syntax expression prefixes, with cross-references to descriptions of these prefixes and the operators that you can specify for them. Note that some prefixes can work with multiple types of operators. For example, a cookie can be parsed by using operators for text or operators for HTTP headers.

You can use any element in the following tables as a complete expression on its own, or you can use various operators to combine these expression elements with others to form more complex expressions.

Note: The Description column in the following table contains cross-references to additional information about prefix usage and applicable operators for the prefix.

Expression Prefix	**Links to Relevant Information, with Applicable Notes and Operator Descriptions**
CLIENT.ETHER	"Prefixes for MAC Addresses." "Operations for MAC Addresses."
CLIENT.ETHER.[DSTMAC | SRCMAC]	"Prefixes for MAC Addresses." "Operations for MAC Addresses."
CLIENT.INTERFACE	Designates an expression that refers to the ID of the network interface through which the current packet entered the Application Switch. See the other CLIENT.INTERFACE prefix descriptions in this table.
CLIENT.INTERFACE.ID	Extracts the ID of the network interface that received the current packet of data. See the other CLIENT.INTERFACE prefix descriptions in this table.
CLIENT.INTERFACE.ID.EQ("id")	Returns Boolean TRUE if the interface's ID matches the ID that is passed as the argument. For example: CLIENT.INTERFACE.ID.EQ("1/1") See "Booleans in Compound Expressions."
CLIENT.INTERFACE.[RXTHROU GHPUT | RXTXTHROUGHPUT |	"Expressions for Numeric Client and Server

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
TXTHROUGHPUT]	Data]." "Compound Operations for Numbers."
CLIENT.IP	Operates on the IP protocol data associated with the current packet. See the other CLIENT.IP prefixes in this table.
CLIENT.IP.DST	"[Prefixes for IPV4 Addresses and IP Subnets]." "[Operations for IPV4 Addresses]." "Compound Operations for Numbers."
CLIENT.IP.SRC	"[Prefixes for IPV4 Addresses and IP Subnets]." "[Operations for IPV4 Addresses]." "Compound Operations for Numbers."
CLIENT.IPV6	Operates on IPv6 protocol data. See the other CLIENT.IPV6 prefixes in this table.
CLIENT.IPV6.DST	"[Expression Prefixes for IPv6 Addresses]." "[Operations for IPV6 Prefixes]."
CLIENT.IPV6.SRC	"[Expression Prefixes for IPv6 Addresses]." "[Operations for IPV6 Prefixes]."
CLIENT.SSL	Operates on the SSL protocol data for the current packet. See the other CLIENT.SSL prefixes in this table.
CLIENT.SSL.CIPHER_BITS	"[Prefixes for Numeric Data in SSL Certificates]." "Compound Operations for Numbers."
CLIENT.SSL.CIPHER_EXPORTA BLE	"[Prefixes for Text-Based SSL and Certificate Data]." "Booleans in Compound Expressions".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
CLIENT.SSL.CLIENT_CERT	"Expressions for SSL Certificates." "Expressions for SSL Certificate Dates."
CLIENT.SSL.IS_SSL	"Prefixes for Text-Based SSL and Certificate Data." "Booleans in Compound Expressions."
CLIENT.SSL.VERSION	"Prefixes for Numeric Data in SSL Certificates." "Compound Operations for Numbers."
CLIENT.TCP	Operates on TCP protocol data. See the other CLIENT.TCP prefixes in this table.
CLIENT.TCP.[DSTPORT | MSS | SRCPORT]	"Expressions for TCP, UDP, and VLAN Data." "Compound Operations for Numbers."
CLIENT.TCP.PAYLOAD(integer)	"Expressions for TCP, UDP, and VLAN Data." "Default Syntax Expressions: Evaluating Text."
CLIENT.UDP	Operates on the UDP protocol data associated with the current packet. See the other CLIENT.UDP prefixes in this table.
CLIENT.UDP.DNS.DOMAIN	"Expressions for TCP, UDP, and VLAN Data." "Default Syntax Expressions: Evaluating Text."
CLIENT.UDP.DNS.DOMAIN.EQ("hostname")	"Expressions for TCP, UDP, and VLAN Data." "Booleans in Compound Expressions."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
CLIENT.UDP.DNS. [IS_AAAAREC | IS_ANYREC | IS_AREC | IS_CNAMEREC | IS_MXREC | IS_NSREC | IS_PTRREC | IS_SOAREC | IS_SRVREC]	"Expressions for TCP, UDP, and VLAN Data." "Booleans in Compound Expressions."
CLIENT.UDP.[DSTPORT | SRCPORT]	"Expressions for TCP, UDP, and VLAN Data." "Compound Operations for Numbers."
CLIENT.VLAN	Operates on the VLAN through which the current packet entered the NetScaler. See the other CLIENT.VLAN prefixes in this table.
CLIENT.VLAN.ID	"Expressions for TCP, UDP, and VLAN Data." "Compound Operations for Numbers."
HTTP.REQ	Operates on HTTP requests. See the other HTTP.REQ prefixes in this table.
HTTP.REQ.BODY(integer)	"Expression Prefixes for Text in HTTP Requests and Responses." "Basic Operations on Text."
HTTP.REQ.CACHE_CONTROL	"Prefixes for Cache-Control Headers." "Operations for Cache-Control Headers."
HTTP.REQ.CONTENT_LENGTH	"Expressions for Numeric HTTP Payload Data Other Than Dates." "Compound Operations for Numbers."
HTTP.REQ.COOKIE	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text."
HTTP.REQ.DATE	"Format of Dates and Times in an

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
	Expression]." "[Expressions for HTTP Request and Response Dates]." "[Default Syntax Expressions: Evaluating Text]." "Compound Operations for Numbers." "[Operations for HTTP Headers]."
HTTP.REQ.HEADER("header_name")	"[Expression Prefixes for Text in HTTP Requests and Responses]." "[Prefixes for HTTP Headers]." "[Operations for HTTP Headers]."
HTTP.REQ.FULL_HEADER("header_name")	"[Prefixes for HTTP Headers]." "[Operations for HTTP Headers]."
HTTP.REQ.HOSTNAME	"[Expression Prefixes for Text in HTTP Requests and Responses]."
HTTP.REQ.HOSTNAME.[DOMAIN | Server]	"[Expression Prefixes for Text in HTTP Requests and Responses]." "[Basic Operations on Text]."
HTTP.REQ.HOSTNAME.EQ("hostname")	"[Expression Prefixes for Text in HTTP Requests and Responses]." "Booleans in Compound Expressions." "[Basic Operations on Expression Prefixes]."
HTTP.REQ.HOSTNAME.PORT	"[Expression Prefixes for Text in HTTP Requests and Responses]." "Compound Operations for Numbers."
HTTP.REQ.IS_VALID	Returns TRUE if the HTTP request is properly formed. See "Booleans in Compound Expressions."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
HTTP.REQ.METHOD	"Expression Prefixes for Text in HTTP Requests and Responses." "Basic Operations on Text." "Complex Operations on Text."
HTTP.REQ.TRACKING	Returns the HTTP body tracking mechanism. See the descriptions of other HTTP.REQ.TRACKING prefixes in this table.
HTTP.REQ.TRACKING.EQ("tracking_mechanism")	Returns TRUE or FALSE. See "Booleans in Compound Expressions."
HTTP.REQ.URL	Obtains the HTTP URL object from the request and sets the text mode to URLENCODED by default. See "Expression Prefixes for Text in HTTP Requests and Responses."
HTTP.REQ.URL.[CVPN_ENCODE | HOSTNAME | HOSTNAME.DOMAIN | SERVER | PATH | PATH_AND_QUERY | PROTOCOL | QUERY | SUFFIX | VERSION]	"Expression Prefixes for Text in HTTP Requests and Responses." "Basic Operations on Text." "Complex Operations on Text."
HTTP.REQ.URL.HOSTNAME.EQ("hostname")	"Expression Prefixes for Text in HTTP Requests and Responses." "Booleans in Compound Expressions."
HTTP.REQ.URL.HOSTNAME.PORT	"Expression Prefixes for Text in HTTP Requests and Responses." "Compound Operations for Numbers."
HTTP.REQ.URL.PATH.IGNORE_ EMPTY_ELEMENTS	Ignores spaces in the data. See the table "HTTP Expression Prefixes that Return Text."
HTTP.REQ.URL.QUERY.IGNORE _EMPTY_ELEMENTS	Ignores spaces in the data. See the table "HTTP Expression Prefixes that Return Text."
HTTP.REQ.USER.IS_MEMBER_OF	"HTTP Expression Prefixes that Return Text."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operation Descriptions
HTTP.REQ.USER.NAME	"Expression Prefixes for Text in HTTP Requests and Responses."
HTTP.REQ.VERSION	"Expression Prefixes for Text in HTTP Requests and Responses."
HTTP.REQ.VERSION.[MAJOR | MINOR]	Operates on the major or minor HTTP version string. See "Expression Prefixes for Text in HTTP Requests and Responses" and "Compound Operations for Numbers."
HTTP.RES	Operates on HTTP responses.
HTTP.RES.BODY(integer)	"Expression Prefixes for Text in HTTP Requests and Responses." "Basic Operations on Text." "Complex Operations on Text."
HTTP.RES.CACHE_CONTROL	"Prefixes for Cache-Control Headers." "Operations for Cache-Control Headers."
HTTP.RES.CONTENT_LENGTH	"Expression Prefixes for Text in HTTP Requests and Responses." "Operations for HTTP Headers." "Compound Operations for Numbers."
HTTP.RES.DATE	"Format of Dates and Times in an Expression." "Expressions for HTTP Request and Response Dates." "Expression Prefixes for Text in HTTP Requests and Responses." "Compound Operations for Numbers." "Operations for HTTP Headers."
HTTP.RES.HEADER("header_name")	"Expression Prefixes for Text in HTTP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
	Requests and Responses]." "[Prefixes for HTTP Headers]." "[Operations for HTTP Headers]."
HTTP.REQ.FULL_HEADER("header_name")	"[Prefixes for HTTP Headers]." "[Operations for HTTP Headers]."
HTTP.REQ.TXID	"[Prefixes for HTTP Headers]." "[Operations for HTTP Headers]."
HTTP.RES.IS_VALID	Returns TRUE if the HTTP response is properly formed. See "Booleans in Compound Expressions."
HTTP.RES.SET_COOKIE	"[Prefixes for HTTP Headers]." "[Operations for HTTP Headers]." "[Default Syntax Expressions: Evaluating Text]."
HTTP.RES.SET_COOKIE.COOKIE("name")	"[Prefixes for HTTP Headers]." "[Operations for HTTP Headers]." "[Default Syntax Expressions: Evaluating Text]."
HTTP.RES.SET_COOKIE.COOKI E.[DOMAIN | PATH | PORT]	"[Prefixes for HTTP Headers]." "[Operations for HTTP Headers]." "[Default Syntax Expressions: Evaluating Text]."
HTTP.RES.SET_COOKIE.COOKIE.EXPIRES	Obtains the Expires field of the cookie as a date string. The value of the Expires attribute can be operated upon as a time object. If multiple Expires fields are present, this expression operates on the first one. If the Expires attribute is absent, a string of length zero is returned.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Also see: **Links to Relevant Information, with Applicable Notes and Operator Descriptions**
	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text." "Compound Operations for Numbers."
HTTP.RES.SET_COOKIE.COOKIE.PATH.IGNORE_EMPTY_ELEMENTS	Ignores spaces in the data. For an example, see the table "Expression Prefixes for Text in HTTP Requests and Responses."
HTTP.RES.SET_COOKIE.COOKIE.PORT.IGNORE_EMPTY_ELEMENTS	Ignores spaces in the data. For an example, see the table "HTTP Expression Prefixes that Return Text."
HTTP.RES.SET_COOKIE.COOKIE.VERSION	"Prefixes for HTTP Headers." "Compound Operations for Numbers."
HTTP.RES.SET_COOKIE.COOKIE("name",integer)[.PORT | PATH | DOMAIN | VERSION | EXPIRES]	"Prefixes for HTTP Headers." "Default Syntax Expressions: Evaluating Text."
HTTP.RES.SET_COOKIE.COOKIE.EXPIRES	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text." "Compound Operations for Numbers."
HTTP.RES.SET_COOKIE.EXISTS("name")	"Prefixes for HTTP Headers." "Booleans in Compound Expressions."
HTTP.RES.SET_COOKIE2	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
HTTP.RES.SET_COOKIE2.COOKIE("name")	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text."
HTTP.RES.SET_COOKIE2.COOKIE.[DOMAIN | PATH | PORT]	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text."
HTTP.RES.SET_COOKIE2.COOKIE.EXPIRES	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text." "Compound Operations for Numbers."
HTTP.RES.SET_COOKIE2.COOKIE.PATH.IGNORE_EMPTY_ELEMENTS	Ignores spaces in the data. For an example, see the table "HTTP Expression Prefixes that Return Text."
HTTP.RES.SET_COOKIE2.COOKIE.PORT.IGNORE_EMPTY_ELEMENTS	Ignores spaces in the data. For an example, see the table "HTTP Expression Prefixes that Return Text." See also "Default Syntax Expressions: Evaluating Text" and "Compound Operations for Numbers."
HTTP.RES.SET_COOKIE2.COOKIE("name",integer)[.PORT | PATH | DOMAIN | VERSION | EXPIRES]	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text."
HTTP.RES.SET_COOKIE2.COOKIE.DOMAIN	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
HTTP.RES.SET_COOKIE2.COOKIE.EXPIRES	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text." "Compound Operations for Numbers."
HTTP.RES.SET_COOKIE2.COOKIE.VERSION	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Default Syntax Expressions: Evaluating Text." "Compound Operations for Numbers."
HTTP.RES.SET_COOKIE2.EXISTS("name")	"Prefixes for HTTP Headers." "Operations for HTTP Headers." "Booleans in Compound Expressions."
HTTP.RES.STATUS	"Expression Prefixes for Text in HTTP Requests and Responses." "Compound Operations for Numbers."
HTTP.RES.STATUS_MSG	"Expression Prefixes for Text in HTTP Requests and Responses."
HTTP.RES.TRACKING	Returns the HTTP body tracking mechanism. See the descriptions of other HTTP.REQ.TRACKING prefixes in this table.
HTTP.RES.TRACKING.EQ("tracking_method")	Returns TRUE or FALSE. See "Booleans in Compound Expressions."
HTTP.RES.TXID	" Prefixes for HTTP Headers." "Operations for HTTP Headers."
HTTP.RES.VERSION	"Expression Prefixes for Text in HTTP Requests and Responses."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
HTTP.RES.VERSION.[MAJOR | MINOR]	Applicable to the major or minor HTTP version. See "Expression Prefixes for Text in HTTP Requests and Responses" and "Compound Operations for Numbers."
SERVER	Designates an expression that refers to the server. This is the starting point for access into parameters such as Ether and SSL. See the other SERVER prefixes in this table.
SERVER.ETHER	Operates on the ethernet protocol data associated with the current packet. See the other SERVER prefixes in this table.
SERVER.ETHER.DSTMAC	"Prefixes for MAC Addresses." "Prefixes for MAC Addresses."
SERVER.INTERFACE	Designates an expression that refers to the ID of the network interface that received the current packet of data. See the other SERVER.INTERFACE prefixes in this table.
SERVER.INTERFACE.ID.EQ("id")	Returns Boolean TRUE if the interface's ID matches the ID that is passed as the argument. For example: SERVER.INTERFACE.ID.EQ("LA/1") See "Booleans in Compound Expressions."
SERVER.INTERFACE.[RXTHROUGHPUT | RXTXTHROUGHPUT | TXTHROUGHPUT]	"Expressions for Numeric Client and Server Data." "Compound Operations for Numbers."
SERVER.IP	Operates on the IP protocol data associated with the current packet. See the other SERVER.IP prefixes in this table.
SERVER.IP.[DST | SRC]	"Prefixes for IPV4 Addresses and IP Subnets."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
SERVER.IPV6	Operates on IPv6 protocol data. See the other SERVER.IPV6 prefixes in this table.
SERVER.IPV6.DST	"Expression Prefixes for IPv6 Addresses." "Operations for IPV6 Prefixes."
SERVER.IPV6.SRC	"Expression Prefixes for IPv6 Addresses." "Operations for IPV6 Prefixes."
SERVER.TCP	Operates on TCP protocol data. See the other CLIENT.TCP prefixes in this table.
SERVER.TCP.[DSTPORT | MSS | SRCPORT]	"Expressions for TCP, UDP, and VLAN Data." "Compound Operations for Numbers."
SERVER.VLAN	Operates on the VLAN through which the current packet entered the NetScaler. See the other SERVER.VLAN prefixes in this table.
SERVER.VLAN.ID	"Expressions for TCP, UDP, and VLAN Data." "Compound Operations for Numbers."
SYS	Designates an expression that refers to the NetScaler itself, not to the client or server.. See the other SYS prefixes in this table.
SYS.EVAL_CLASSIC_EXPR(cla ssic_expression)	"Classic Expressions in Default Syntax Expressions." "Booleans in Compound Expressions."
SYS.HTTP_CALLOUT(http_callout)	"HTTP Callouts."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
SYS.CHECK_LIMIT	"Rate Limiting."
SYS.TIME	"Expressions for the NetScaler System Time."
	"Compound Operations for Numbers."
SYS.TIME.[BETWEEN(time1, time2) | EQ(time) | GE(time) | GT(time) | LE(time) | LT(time) | WITHIN(time1, time2)]	"Expressions for the NetScaler System Time." "Booleans in Compound Expressions." "Compound Operations for Numbers."
SYS.TIME.[DAY | HOURS | MINUTES | MONTH | RELATIVE_BOOT | RELATIVE_NOW SECONDS | WEEKDAY | YEAR]	"Expressions for the NetScaler System Time." "Compound Operations for Numbers."
SYS.RANDOM	Returns a random number between 0 and 1, inclusive of 0 but exclusive of 1.
VPN.BASEURL.[CVPN_DECODE | CVPN_ENCODE | HOSTNAME | HOSTNAME.DOMAIN | HOSTNAME.SERVER | PATH | PATH_AND_QUERY | PROTOCOL | QUERY | SUFFIX]	"Expression Prefixes for VPNs and Clientless VPNs."
VPN.BASEURL.HOSTNAME.EQ("hostname")	"Expression Prefixes for VPNs and Clientless VPNs." "Booleans in Compound Expressions."
VPN.BASEURL.HOSTNAME.PORT	"Expression Prefixes for VPNs and Clientless VPNs." "Compound Operations for Numbers."
VPN.BASEURL.PATH.IGNORE_EMPTY_ELEMENTS	Ignores spaces in the data. For an example, see the table "HTTP Expression Prefixes that Return Text."
VPN.BASEURL.QUERY.IGNORE_EMPTY_ELEMENTS	Ignores spaces in the data. For an example, see the table "HTTP Expression Prefixes that Return Text."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
VPN.CLIENTLESS_BASEURL	"Expression Prefixes for VPNs and Clientless VPNs."
VPN.CLIENTLESS_BASEURL. [CVPN_DECODE | CVPN_ENCODE | HOSTNAME | HOSTNAME.DOMAIN | HOSTNAME.SERVER | PATH | PATH_AND_QUERY | PROTOCOL | QUERY | SUFFIX]	"Expression Prefixes for VPNs and Clientless VPNs."
VPN.CLIENTLESS_BASEURL.HOSTNAME.EQ("hostname")	"Expression Prefixes for VPNs and Clientless VPNs." "Booleans in Compound Expressions."
VPN.CLIENTLESS_BASEURL.HOSTNAME.PORT	"Expression Prefixes for VPNs and Clientless VPNs." "Compound Operations for Numbers."
VPN.CLIENTLESS_BASEURL.PATH.IGNORE_EMPTY_ELEMENTS	Ignores spaces in the data. For an example, see the table "HTTP Expression Prefixes that Return Text."
VPN.CLIENTLESS_BASEURL.QUERY.IGNORE_EMPTY_ELEMENTS	Ignores spaces in the data. For an example, see the table "HTTP Expression Prefixes that Return Text."
VPN.CLIENTLESS_HOSTURL	"Expression Prefixes for VPNs and Clientless VPNs."
VPN.CLIENTLESS_HOSTURL.[CVPN_DECODE | CVPN_ENCODE | HOSTNAME | HOSTNAME.DOMAIN | HOSTNAME.SERVER | PATH | PATH_AND_QUERY | PROTOCOL | QUERY | SUFFIX]	"Expression Prefixes for VPNs and Clientless VPNs."
VPN.CLIENTLESS_HOSTURL.HOSTNAME.EQ("hostname")	"Expression Prefixes for VPNs and Clientless VPNs." "Booleans in Compound Expressions."
VPN.CLIENTLESS_HOSTURL.HOSTNAME.PORT	"Expression Prefixes for VPNs and Clientless VPNs." "Compound Operations for Numbers."
VPN.CLIENTLESS_HOSTURL.PATH.IGNORE_EMPTY_ELEMENTS	Ignores spaces in the data. For an example,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Prefix	Links to Relevant Information, with Applicable Notes and Operator Descriptions
VPN.CLIENTLESS_HOSTURL.QUERY.IGNORE_EMPTY_ELEMENTS	see the table "HTTP Expression Prefixes that Return Text." Ignores spaces in the data. For an example, see the table "HTTP Expression Prefixes that Return Text."
VPN.HOST	" Expression Prefixes for VPNs and Clientless VPNs."
VPN.HOST.[DOMAIN | Server]	"Expression Prefixes for VPNs and Clientless VPNs."
VPN.HOST.EQ("hostname")	"Expression Prefixes for VPNs and Clientless VPNs." "Booleans in Compound Expressions."
VPN.HOST.PORT	"Expression Prefixes for VPNs and Clientless VPNs." "Default Syntax Expressions: Evaluating Text." "Compound Operations for Numbers."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions Reference-Classic Expressions

May 25, 2015

The subtopics listed in the table of contents on the left side of your screen contain tables listing the NetScaler classic expressions.

In the table of operators, the result type of each operator is shown at the beginning of the description. In the other tables, the level of each expression is shown at the beginning of the description. For named expressions, each expression is shown as a whole.

This document includes the following details:

- Operators
- General Expressions
- Client Security Expressions
- Network-Based Expressions
- Date/Time Expressions
- File System Expressions
- Built-In Named Expressions (General)
- Built-In Named Expressions (Anti-Virus)
- Built-In Named Expressions (Personal Firewall)
- Built-In Named Expressions (Client Security)

Operators

Expression Element	Definition
==	Boolean. Returns TRUE if the current expression equals the argument. For text operations, the items being compared must exactly match one another. For numeric operations, the items must evaluate to the same number.
!=	Boolean. Returns TRUE if the current expression does not equal the argument. For text operations, the items being compared must not exactly match one another. For numeric operations, the items must not evaluate to the same number.
CONTAINS	Boolean. Returns TRUE if the current expression contains the string that is designated in the argument.
NOTCONTAINS	Boolean. Returns TRUE if the current expression does not contain the string that is designated in the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Element	Definition
CONTENTS	Text. Returns the contents of the current expression.
EXISTS	Boolean. Returns TRUE if the item designated by the current expression exists.
NOTEXISTS	Boolean. Returns TRUE if the item designated by the current expression does not exist.
>	Boolean. Returns TRUE if the current expression evaluates to a number that is greater than the argument.
<	Boolean. Returns TRUE if the current expression evaluates to a number that is less than the argument.
>=	Boolean. Returns TRUE if the current expression evaluates to a number that is greater than or equal to the argument.
<=	Boolean. Returns TRUE if the current expression evaluates to a number that is less than or equal to the argument.

General Expressions

Expression Element	Definition
REQ	Flow Type. Operates on incoming (or request) packets.
REQ.HTTP	Protocol Operates on HTTP requests.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Element	Definition
REQ.HTTP.METHOD	Qualifier Designates the HTTP method.
REQ.HTTP.URL	Qualifier Designates the URL.
REQ.HTTP.URLTOKENS	Qualifier Designates the URL token.
REQ.HTTP.VERSION	Qualifier Designates the HTTP version.
REQ.HTTP.HEADER	Qualifier Designates the HTTP header.
REQ.HTTP.URLLEN	Qualifier Designates the number of characters in the URL.
REQ.HTTP.URLQUERY	Qualifier Designates the query portion of the URL.
REQ.HTTP.URLQUERYLEN	Qualifier Designates the length of the query portion of the URL.
REQ.SSL	Protocol Operates on SSL requests.
REQ.SSL.CLIENT.CERT	Qualifier Designates the entire client certificate.
REQ.SSL.CLIENT.CERT.SUBJECT	Qualifier Designates the client certificate subject.
REQ.SSL.CLIENT.CERT.ISSUER	Qualifier

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Element	Definition
	Designates the issuer of the client certificate.
REQ.SSL.CLIENT.CERT.SIGALGO	Qualifier Designates the validation algorithm used by the client certificate.
REQ.SSL.CLIENT.CERT.VERSION	Qualifier Designates the client certificate version.
REQ.SSL.CLIENT.CERT.VALIDFROM	Qualifier Designates the date before which the client certificate is not valid.
REQ.SSL.CLIENT.CERT.VALIDTO	Qualifier Designates the date after which the client certificate is not valid.
REQ.SSL.CLIENT.CERT.SERIALNUMBER	Qualifier Designates the serial number of the client certificate.
REQ.SSL.CLIENT.CIPHER.TYPE	Qualifier Designates the encryption protocol used by the client.
REQ.SSL.CLIENT.CIPHER.BITS	Qualifier Designates the number of bits used by the client's SSL key.
REQ.SSL.CLIENT.SSL.VERSION	Qualifier Designates the SSL version that the client is using.
REQ.TCP	Protocol Operates on incoming TCP packets.
REQ.TCP.SOURCEPORT	Qualifier Designates the source port of the incoming packet.
REQ.TCP.DESTPORT	Qualifier Designates the destination port of the incoming packet.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Element	Definition
REQ.IP	Protocol Operates on incoming IP packets.
REQ.IP.SOURCEIP	Qualifier Designates the source IP of the incoming packet.
REQ.IP.DESTIP	Qualifier Designates the destination IP of the incoming packet.
RES	Flow Type Operates on outgoing (or response) packets.
RES.HTTP	Protocol Operates on HTTP responses.
RES.HTTP.VERSION	Qualifier Designates the HTTP version.
RES.HTTP.HEADER	Qualifier Designates the HTTP header.
RES.HTTP.STATUSCODE	Qualifier Designates the status code of the HTTP response.
RES.TCP	Protocol Operates on incoming TCP packets.
RES.TCP.SOURCEPORT	Qualifier Designates the source port of the outgoing packet.
RES.TCP.DESTPORT	Qualifier Designates the destination port of the outgoing packet.
RES.IP	Protocol

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Element	Definition
	Operates on outgoing IP packets.
RES.IP.SOURCEIP	Qualifier Designates the source IP of the outgoing packet. This can be in IPv4 or IPv6 format. For example: add expr exp3 “sourceip == 10.102.32.123 –netmask 255.255.255.0 && destip == 2001::23/120”.
RES.IP.DESTIP	Qualifier Designates the destination IP of the outgoing packet.

Client Security Expressions

Updated: 2013-10-21

The expressions to configure client settings on the Access Gateway with the following software:

- Antivirus
- Personal firewall
- Antispam
- Internet Security

For example usage, see http://support.citrix.com/article/CTX112599.

Actual Expression	Definition
CLIENT.APPLICATION.AV(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated anti-virus program and version.
CLIENT.APPLICATION.AV(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated anti-virus program and version.
CLIENT.APPLICATION.PF(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated personal firewall program and version.
CLIENT.APPLICATION.PF(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated personal firewall program and version.
CLIENT.APPLICATION.IS(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated internet security program and version.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Actual Expression	Definition
CLIENT.APPLICATION.IS(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated internet security program and version.
CLIENT.APPLICATION.AS(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated anti-spam program and version.
CLIENT.APPLICATION.AS(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated anti-spam program and version.

Network-Based Expressions

Expression	Definition
REQ	Flow Type. Operates on incoming, or request, packets.
REQ.VLANID	Qualifier. Operates on the virtual LAN (VLAN) ID.
REQ.INTERFACE.ID	Qualifier. Operates on the ID of the designated NetScaler interface.
REQ.INTERFACE.RXTHROUGHPUT	Qualifier. Operates on the raw received packet throughput of the designated NetScaler interface.
REQ.INTERFACE.TXTHROUGHPUT	Qualifier. Operates on the raw transmitted packet throughput of the designated NetScaler interface.
REQ.INTERFACE.RXTXTHROUGHPUT	Qualifier. Operates on the raw received and transmitted packet throughput of the designated NetScaler interface.
REQ.ETHER.SOURCEMAC	Qualifier. Operates on the source MAC address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Definition
REQ.ETHER.DESTMAC	Qualifier. Operates on the destination MAC address.
RES	Flow Type. Operates on outgoing (or response) packets.
RES.VLANID	Qualifier. Operates on the virtual LAN (VLAN) ID.
RES.INTERFACE.ID	Qualifier. Operates on the ID of the designated NetScaler interface.
RES.INTERFACE.RXTHROUGHPUT	Qualifier. Operates on the raw received packet throughput of the designated NetScaler interface.
RES.INTERFACE.TXTHROUGHPUT	Qualifier. Operates on the raw transmitted packet throughput of the designated NetScaler interface.
RES.INTERFACE.RXTXTHROUGHPUT	Qualifier. Operates on the raw received and transmitted packet throughput of the designated NetScaler interface.
RES.ETHER.SOURCEMAC	Qualifier. Operates on the source MAC address.
RES.ETHER.DESTMAC	Qualifier. Operates on the destination MAC address.

Date/Time Expressions

Expression	Definition
TIME	Qualifier.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Definition
	Operates on the date and time of day, GMT.
DATE	Qualifier. Operates on the date, GMT.
DAYOFWEEK	Operates on the specified day in the week, GMT.

File System Expressions

Updated: 2013-09-30

You can specify file system expressions in authorization policies for users and groups who access file sharing through the NetScaler Gateway file transfer utility (the VPN portal). These expressions work with the NetScaler Gateway file transfer authorization feature to control user access to file servers, folders, and files. For example, you can use these expressions in authorization policies to control access based on file type and size.

Expression	Definition
FS.COMMAND	Qualifier. Operates on a file system command. The user can issue multiple commands on a file transfer portal. (For example, ls to list files or mkdir to create a directory). This expression returns the current action that the user is taking. Possible values: Neighbor, login, ls, get, put, rename, mkdir, rmdir, del, logout, any. Following is an example: Add authorization policy pol1 “fs.command eq login && (fs.user eq administrator || fs.serverip eq 10.102.88.221 –netmask 255.255.255.252)” allow
FS.USER	Returns the user who is logged on to the file system.
FS.SERVER	Returns the host name of the target server. In the following example, the string win2k3-88-22 is the server name: fs.server eq win2k3-88-221
FS.SERVERIP	Returns the IP address of the target server.
FS.SERVICE	Returns a shared root directory on the file server. If a particular folder is exposed as shared, a user can directly log on to the specified first level folder. This first level folder is called a service. For example, in the path \\hostname\SERVICEX\ETC, SERVICEX is the service. As another example, if a user accesses the file \\hostname\service1\dir1\file1.doc, FS.SERVICE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Definition
	will return service1. Following is an example: fs.service notcontains New
FS.DOMAIN	Returns the domain name of the target server.
FS.PATH	Returns the complete path of the file being accessed. For example, if a user accesses the file \\hostname\service1\dir1\file1.doc, FS.PATH will return \service\dir1\file1.doc. Following is an example: fs.path notcontains SSL
FS.FILE	Returns the name of the file being accessed. For example, if a user accesses the file \\hostname\service1\dir1\file1.doc, FS.FILE will return file1.doc.
FS.DIR	Returns the directory being accessed. For example, if a user accesses the file \\hostname\service1\dir1\file1.doc, FS.DIR will return \service\dir1.
FS.FILE.ACCESSTIME	Returns the time at which the file was last accessed. This is one of several options that provide you with granular control over actions that the user performs. (See the following entries in this table.)
FS.FILE.CREATETIME	Returns the time at which the file was created.
FS.FILE.MODIFYTIME	Returns the time at which the file was edited.
FS.FILE.WRITETIME	Returns the time of the most recent change in the status of the file.
FS.FILE.SIZE	Returns the file size.
FS.DIR.ACCESSTIME	Returns the time at which the directory was last accessed.
FS.DIR.CREATETIME	Returns the time at which the directory was created.
FS.DIR.MODIFYTIME	Returns the time at which the directory was last modified.
FS.DIR.WRITETIME	Returns the time at which the directory status last changed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: File system expressions do not support regular expressions.

Built-In Named Expressions (General)

Expression	Definition
ns_all_apps_ncomp	Tests for connections with destination ports between 0 and 65535. In other words, tests for all applications.
ns_cachecontrol_nocache	Tests for connections with an HTTP Cache-Control header that contains the value “no-cache”.
ns_cachecontrol_nostore	Tests for connections with an HTTP Cache-Control header that contains the value “no-store”.
ns_cmpclient	Tests the client to determine if it accepts compressed content.
ns_content_type	Tests for connections with an HTTP Content-Type header that contains “text”.
ns_css	Tests for connections with an HTTP Content-Type header that contains “text/css”.
ns_ext_asp	Tests for HTTP connections to any URL that contains the string .asp—in other words, any connection to an active server page (ASP).
ns_ext_cfm	Tests for HTTP connections to any URL that contains the string .cfm
ns_ext_cgi	Tests for HTTP connections to any URL that contains the string .cgi—in other words, any connection to a common gateway interface (CGI) script.
ns_ext_ex	Tests for HTTP connections to any URL that contains the string .ex
ns_ext_exe	Tests for HTTP connections to any URL that contains the string .exe—in other words, any connection to a executable file.
ns_ext_htx	Tests for HTTP connections to any URL that contains the string .htx
ns_ext_not_gif	Tests for HTTP connections to any URL that does not contain the string .gif—in other words, any connection to a URL that is not a GIF image.
ns_ext_not_jpeg	Tests for HTTP connections to any URL that does not contain the string .jpeg—in other words, any connection to a URL that is not a JPEG image.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Definition
ns_ext_shtml	Tests for HTTP connections to any URL that contains the string .shtml—in other words, any connection to a server-parsed HTML page.
ns_false	Always returns a value of FALSE.
ns_farclient	Client is in a different geographical region from the NetScaler, as determined by the geographical region in the client's IP address. The following regions are predefined: 192.0.0.0 – 193.255.255.255: Multi-regional 194.0.0.0 – 195.255.255.255: European Union 196.0.0.0 – 197.255.255.255: Other1 198.0.0.0 – 199.255.255.255: North America 200.0.0.0 – 201.255.255.255: Central and South America 202.0.0.0 – 203.255.255.255: Pacific Rim 204.0.0.0 – 205.255.255.255: Other2 206.0.0.0 – 207.255.255.255: Other3
ns_header_cookie	Tests for HTTP connections that contain a Cookie header
ns_header_pragma	Tests for HTTP connections that contain a Pragma: no-cache header.
ns_mozilla_47	Tests for HTTP connections whose User-Agent header contains the string Mozilla/4.7—in other words, any connection from a client using the Mozilla 4.7 Web browser.
ns_msexcel	Tests for HTTP connections whose Content-Type header contains the string application/vnd.msexcel—in other words, any connection transmitting a Microsoft Excel spreadsheet.
ns_msie	Tests for HTTP connections whose User-Agent header contains the string MSIE—in other words, any connection from a client using any version of the Internet Explorer Web browser.
ns_msppt	Tests for HTTP connections whose Content-Type header contains the string application/vnd.ms-powerpoint—in other words, any connection transmitting a Microsoft PowerPoint file.
ns_msword	Tests for HTTP connections whose Content-Type header contains the string

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Definition
	application/vnd.msword—in other words, any connection transmitting a Microsoft Word file.
ns_non_get	Tests for HTTP connections that use any HTTP method except for GET.
ns_slowclient	Returns TRUE if the average round trip time between the client and the NetScaler is more than 80 milliseconds.
ns_true	Returns TRUE for all traffic.
ns_url_path_bin	Tests the URL path to see if it points to the /bin/ directory.
ns_url_path_cgibin	Tests the URL path to see if it points to the CGI-BIN directory.
ns_url_path_exec	Tests the URL path to see if it points to the /exec/ directory.
ns_url_tokens	Tests for the presence of URL tokens.
ns_xmldata	Tests for the presence of XML data.

Built-In Named Expressions (Anti-Virus)

Expression	Definition
McAfee Virus Scan 11	Tests to determine whether the client is running the latest version of McAfee VirusScan.
McAfee Antivirus	Tests to determine whether the client is running any version of McAfee Antivirus.
Symantec AntiVirus 10 (with Updated Definition File)	Tests to determine whether the client is running the most current version of Symantec AntiVirus.
Symantec AntiVirus 6.0	Tests to determine whether the client is running Symantec AntiVirus 6.0.
Symantec AntiVirus 7.5	Tests to determine whether the client is running Symantec AntiVirus 7.5.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression	Definition
TrendMicro OfficeScan 7.3	Tests to determine whether the client is running Trend Microsystems' OfficeScan, version 7.3.
TrendMicro AntiVirus 11.25	Tests to determine whether the client is running Trend Microsystems' AntiVirus, version 11.25.
Sophos Antivirus 4	Tests to determine whether the client is running Sophos Antivirus, version 4.
Sophos Antivirus 5	Tests to determine whether the client is running Sophos Antivirus, version 5.
Sophos Antivirus 6	Tests to determine whether the client is running Sophos Antivirus, version 6.

Built-In Named Expressions (Personal Firewall)

Expression	Definition
TrendMicro OfficeScan 7.3	Tests to determine whether the client is running Trend Microsystems' OfficeScan, version 7.3.
Sygate Personal Firewall 5.6	Tests to determine whether the client is running the Sygate Personal Firewall, version 5.6.
ZoneAlarm Personal Firewall 6.5	Tests to determine whether the client is running the ZoneAlarm Personal Firewall, version 6.5.

Built-In Named Expressions (Client Security)

Expression	Definition
Norton Internet Security	Tests to determine whether the client is running any version of Norton Internet Security.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Summary Examples of Default Syntax Expressions and Policies

Jul 10, 2013

The following table provides examples of default syntax expressions that you can use as the basis for your own default syntax expressions.

Table 1. Examples of Default Syntax Expressions

Expression Type	Sample Expressions
Look at the method used in the HTTP request.	http.req.method.eq(post) http.req.method.eq(get)
Check the Cache-Control or Pragma header value in an HTTP request (req) or response (res).	http.req.header("Cache-Control").contains("no-store") http.req.header("Cache-Control").contains("no-cache") http.req.header("Pragma").contains("no-cache") http.res.header("Cache-Control").contains("private") http.res.header("Cache-Control").contains("public") http.res.header("Cache-Control").contains("must-revalidate") http.res.header("Cache-Control").contains("proxy-revalidate") http.res.header("Cache-Control").contains("max-age")
Check for the presence of a header in a request (req) or response (res).	http.req.header("myHeader").exists http.res.header("myHeader").exists
Look for a particular file type in an HTTP request based on the file extension.	http.req.url.contains(".html") http.req.url.contains(".cgi") http.req.url.contains(".asp")

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Type	Sample Expressions
	http.req.url.contains(".exe") http.req.url.contains(".cfm") http.req.url.contains(".ex") http.req.url.contains(".shtml") http.req.url.contains(".htx") http.req.url.contains("/cgi-bin/") http.req.url.contains("/exec/") http.req.url.contains("/bin/")
Look for anything that is other than a particular file type in an HTTP request.	http.req.url.contains(".gif").not http.req.url.contains(".jpeg").not
Check the type of file that is being sent in an HTTP response based on the Content-Type header.	http.res.header("Content-Type").contains("text") http.res.header("Content-Type").contains("application/msword") http.res.header("Content-Type").contains("vnd.ms-excel") http.res.header("Content-Type").contains("application/vnd.ms-powerpoint") http.res.header("Content-Type").contains("text/css") http.res.header("Content-Type").contains("text/xml") http.res.header("Content-Type").contains("image/")
Check whether this response contains an expiration header.	http.res.header("Expires").exists
Check for a Set-Cookie header in a response.	http.res.header("Set-Cookie").exists
Check the agent that sent the response.	http.res.header("User-Agent").contains("Mozilla/4.7") http.res.header("User-Agent").contains("MSIE")
Check if the first 1024 bytes of the body of a request starts with	http.req.body(1024).contains("some text")

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expression Type	Sample Expressions
the string “some text”.	

The following table shows examples of policy configurations and bindings for commonly used functions.

Table 2. Examples of Default Syntax Expressions and Policies

Purpose	Example
Use the rewrite feature to replace occurrences of http:// with https:// in the body of an HTTP response.	add rewrite action httpRewriteAction replace_all http.res.body(50000) "\"https://\"" -pattern http:// add rewrite policy demo_rep34312 "http.res.body(50000).contains(\"http://\")" httpRewriteAction
Replace all occurrences of “abcd” with “1234” in the first 1000 bytes of the HTTP body.	add rewrite action abcdTo1234Action replace_all "http.req.body(1000)" "\"1234\"" -pattern abcd add rewrite policy abcdTo1234Policy "http.req.body(1000).contains(\"abcd\")" abcdTo1234Action bind rewrite global abcdTo1234Policy 100 END -type REQ_OVERRIDE
Downgrade the HTTP version to 1.0 to prevent the server from chunking HTTP responses.	add rewrite action downgradeTo1.0Action replace http.req.version.minor "\"0\"" add rewrite policy downgradeTo1.0Policy "http.req.version.minor.eq(1)" downgradeTo1.0Action bind lb vserver myLBVserver -policyName downgradeTo1.0Policy -priority 100 -gotoPriorityExpression NEXT -type REQUEST
Remove references to the HTTP or HTTPS protocol in all responses, so that if the user's connection is HTTP, the link is opened by using HTTP, and if the user's connection is HTTPS, the link is opened by using HTTPS.	add rewrite action remove_http_https replace_all "http.res.body(1000000).set_text_mode(ignorecase)" "\"//\"" -pattern "re~https?://|HTTPS?://~" add rewrite policy remove_http_https true remove_http_https bind lb vserver test_vsvr -policyName remove_http_https -priority 20 -gotoPriorityExpression NEXT -type RESPONSE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Purpose	Example
Rewrite instances of http:// to https:// in all URLs. This policy uses the responder functionality.	add responder action httpToHttpsAction redirect "\"https://\" + http.req.hostname + http.req.url" -bypassSafetyCheck YES add responder policy httpToHttpsPolicy "!CLIENT.SSL.IS_SSL" httpToHttpsAction bind responder global httpToHttpsPolicy 1 END -type OVERRIDE
Modify a URL to redirect from URL A to URL B. In this example, “file5.html” is appended to the path. This policy uses the responder functionality.	add responder action appendFile5Action redirect "\"http://\" + http.req.hostname + http.req.url + \"/file5.html\"" -bypassSafetyCheck YES add responder policy appendFile5Policy "http.req.url.eq(\"/testsite\")" appendFile5Action bind responder global appendFile5Policy 1 END -type OVERRIDE
Redirect an external URL to an internal URL.	add rewrite action act_external_to_internal REPLACE 'http.req.hostname.server' '"www.my.host.com"' add rewrite policy pol_external_to_internal 'http.req.hostname.server.eq("www.external.host.com")' act_external_to_internal bind rewrite global pol_external_to_internal 100 END -type REQ_OVERRIDE
Redirect requests to www.example.com that have a query string to www.Webn.example.com. The value n is derived from a server parameter in the query string, for example, server=5.	add rewrite action act_redirect_query REPLACE q#http.req.header("Host").before_str(".example.com")' '"Web" + http.req.url.query.value("server")# add rewrite policy pol_redirect_query q#http.req.header("Host").eq("www.example.com") && http.req.url.contains("?")' act_redirect_query#
Limit the number of requests per second from a URL.	add ns limitSelector ip_limit_selector http.req.url "client.ip.src" add ns limitIdentifier ip_limit_identifier -threshold 4 -timeSlice 3600 -mode request_rate -limitType smooth -selectorName ip_limit_selector add responder action my_Web_site_redirect_action redirect "\"http://www.mycompany.com/\""

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Purpose	Example
	add responder policy ip_limit_responder_policy "http.req.url.contains(\"myasp.asp\") && sys.check_limit(\"ip_limit_identifier\")" my_Web_site_redirect_action bind responder global ip_limit_responder_policy 100 END -type default
Check the client IP address but pass the request without modifying the request.	add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' NOREWRITE bind rewrite global check_client_ip_policy 100 END
Remove old headers from a request and insert an NS-Client header.	add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS' del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC' add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END
Remove old headers from a request, insert an NS-Client header, and then modify the “insert header” action so that the value of the inserted header contains the client IP values from the old headers and the NetScaler appliance's connection IP	add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip

© 1999-2017 Citrix Systems, Inc. All rights reserved.

address. **Purpose** Note that this example repeats the previous example, with the exception of the final set rewrite action.	**Example** add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS' del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC' add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END set rewrite action insert_ns_client_header -stringBuilderExpr 'HTTP.REQ.HEADER("x-forwarded-for").VALUE(0) + " " + HTTP.REQ.HEADER("client-ip").VALUE(0) + " " + CLIENT.IP.SRC' -bypassSafetyCheck YES

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tutorial Examples of Default Syntax Policies for Rewrite

Oct 16, 2017

With the rewrite feature, you can modify any part of an HTTP header, and, for responses, you can modify the HTTP body. You can use this feature to accomplish a number of useful tasks, such as removing unnecessary HTTP headers, masking internal URLs, redirecting Web pages, and redirecting queries or keywords.

In the following examples, you first create a rewrite action and a rewrite policy. Then you bind the policy globally.

This document includes the following details:

- Redirecting an External URL to an Internal URL
- Redirecting a Query
- Rewriting HTTP to HTTPS
- Removing Unwanted Headers
- Reducing Web Server Redirects
- Masking the Server Header

For more information about Rewrite syntax descriptions, see Rewrite Command Reference page.

Redirecting an External URL to an Internal URL

Updated: 2013-10-29

This example describes how to create a rewrite action and rewrite policy that redirects an external URL to an internal URL. You create an action, called act_external_to_internal, that performs the rewrite. Then you create a policy called pol_external_to_internal.

To redirect an external URL to an internal URL by using the command line interface

- To create the rewrite action, at the command prompt, type:

  add rewrite action act_external_to_internal REPLACE 'http.req.hostname.server' '"host_name_of_internal_Web_server"

- To create the rewrite policy, at the NetScaler command prompt, type:

  add rewrite policy pol_external_to_internal 'http.req.hostname.server.eq("host_name_of_external_Web_server")' act_external_to_internal

- Bind the policy globally.

To redirect an external URL to an internal URL by using the configuration utility

1. Navigate to AppExpert > Rewrite > Actions.
2. In the details pane, click Add.
3. In the Create Rewrite Action dialog box, enter the name act_external_to_internal.
4. To replace the HTTP server host name with the internal server name, choose Replace from the Type list box.
5. In the Header Name field, type Host.
6. In the String expression for replacement text field, type the internal host name of your Web server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

7. Click Create and then click Close.
8. In the navigation pane, click Policies.
9. In the details pane, click Add.
10. In the Name field, type pol_external_to_internal. This policy will detect connections to the Web server.
11. In the Action drop-down menu, choose the action act_external_to_internal.
12. In the Expression editor, construct the following expression:

    HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")
13. Bind your new policy globally.

Redirecting a Query

This example describes how to create a rewrite action and rewrite policy that redirects a query to the proper URL. The example assumes that the request contains a Host header set to **www.example.com** and a GET method with the **string /query.cgi?server=5**. The redirect extracts the domain name from the host header and the number from the query string, and redirects the user's query to the server **Web5.example.com**, where the rest of the user's query is processed.

Note: Although the following commands appears on multiple lines, you should enter them on a single line without line breaks.

To redirect a query to the appropriate URL using the command line

- To create a rewrite action named act_redirect_query that replaces the HTTP server host name with the internal server name, type:

  add rewrite action act_redirect_query REPLACE q#http.req.header("Host").before_str(".example.com")' '"Web" + http.req.url.query.value("server")#

- To create a rewrite policy named pol_redirect_query, type the following commands at the NetScaler command prompt.. This policy detects connections, to the Web server, that contain a query string. Do not apply this policy to connections that do not contain a query string:

  add rewrite policy pol_redirect_query q#http.req.header("Host").eq("www.example.com") && http.req.url.contains("?")' act_redirect_query#

- Bind your new policy globally.

Because this rewrite policy is highly specific and should be run before any other rewrite policies, it is advisable to assign it a high priority. If you assign it a priority of 1, it will be evaluated first.

Rewriting HTTP to HTTPS

Updated: 2014-09-17

This example describes how to rewrite Web server responses to find all URLs that begin with the string "http" and replace that string with "https." You can use this to avoid having to update Web pages after moving a server from HTTP to HTTPS.

To redirect HTTP URLs to HTTPS by using the command line interface

- To create a rewrite action named act_replace_http_with_https that replaces all instances of the string "http" with the string "https," enter the following command:
  add rewrite action act_replace_http_with_https replace_all 'http.res.body(100)' '"https"' -pattern http

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- To create a rewrite policy named pol_replace_http_with_https that detects connections to the Web server, enter the following command:
  add rewrite policy pol_replace_http_with_https TRUE act_replace_http_with_https NOREWRITE
- Bind your new policy globally.

To troubleshoot this rewrite operation, see "Case Study: Rewrite Policy for Converting HTTP Links to HTTPS not Working."

Removing Unwanted Headers

Updated: 2013-09-02

This example explains how to use a Rewrite policy to remove unwanted headers. Specifically, the example shows how to remove the following headers:

- **Accept Encoding header.** Removing the Accept Encoding header from HTTP responses prevents compression of the response.
- **Content Location header.** Removing the Content Location header from HTTP responses prevents your server from providing a hacker with information that might allow a security breach.

To delete headers from HTTP responses, you create a rewrite action and a rewrite policy, and you bind the policy globally.

To create the appropriate Rewrite action by using the command line interface

At the command prompt, type one of the following commands to either remove the Accept Encoding header and prevent response compression or remove the Content Location header:

- add rewrite action "act_remove-ae" delete_http_header "Accept-Encoding"
- add rewrite action "act_remove-cl" delete_http_header "Content-Location"

To create the appropriate Rewrite policy by using the command line interface

At the command prompt, type one of the following commands to remove either the Accept Encoding header or the Content Location header:

- add rewrite policy "pol_remove-ae" true "act_remove-ae"
- add rewrite policy "pol_remove-cl" true "act_remove-cl"

To bind the policy globally by using the command line interface

At the command prompt, type one of the following commands, as appropriate, to globally bind the policy that you have created:

- bind rewrite global pol_remove_ae 100
- bind rewrite global pol_remove_cl 200

Reducing Web Server Redirects

This example explains how to use a Rewrite policy to modify connections to your home page and other URLs that end with a forward slash (/) to the default index page for your server, preventing redirects and reducing load on your server.

To modify directory-level HTTP requests to include the default home page by using

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the command line

- To create a Rewrite action named action-default-homepage that modifies URLs that end in a forward slash to include the default home page index.html, type:

  add rewrite action "action-default-homepage" replace q#http.req.url.path "/" "/index.html"#

- To create a Rewrite policy named policy-default-homepage that detects connections to your home page and applies your new action, type:

  add rewrite policy "policy-default-homepage" q#http.req.url.path.EQ("/") "action-default-homepage"#

- Globally bind your new policy to put it into effect.

Masking the Server Header

This example explains how to use a Rewrite policy to mask the information in the Server header in HTTP responses from your Web server. That header contains information that hackers can use to compromise your Web site. While masking the header will not prevent a skilled hacker from finding out information about your server, it will make hacking your Web server more difficult and encourage hackers to choose less well protected targets.

To mask the Server header in responses from the command line

1. To create a Rewrite action named act_mask-server that replaces the contents of the Server header with an uninformative string, type:

   add rewrite action "act_mask-server" replace "http.RES.HEADER(\"Server\")" "\"Web Server 1.0\""

2. To create a Rewrite policy named pol_mask-server that detects all connections, type:

   add rewrite policy "pol_mask-server" true "act_mask-server"

3. Globally bind your new policy to put it into effect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tutorial Examples of Classic Policies

May 26, 2015

The following examples describe useful examples of classic policy configuration for certain NetScaler features, such as NetScaler Gateway, application firewall, and SSL.

This document includes the following details:

- NetScaler Gateway Policy to Check for a Valid Client Certificate
- Application Firewall Policy to Protect a Shopping Cart Application
- Application Firewall Policy to Protect Scripted Web Pages
- DNS Policy to Drop Packets from Specific IPs
- SSL Policy to Require Valid Client Certificates

NetScaler Gateway Policy to Check for a Valid Client Certificate

Updated: 2014-09-25

The following policies enable the NetScaler to ensure that a client presents a valid certificate before establishing a connection to a company's SSL VPN.

To check for a valid client certificate by using the command line interface

- Add an action to perform client certificate authentication.

  add ssl action act1 -clientAuth DOCLIENTAUTH

- Create an SSL policy to evaluate the client requests.

  add ssl policy pol1 -rule "REQ.HTTP.METHOD == GET" -action act1

- Add a rewrite action to insert the certificate issuer details into the HTTP header of the requests being sent to web server.

  add rewrite action act2 insert_http_header "CertDN" CLIENT.SSL.CLIENT_CERT.SUBJECT

- Create a rewrite policy to insert the certificate issuer details, if the client certificate exists.

  add rewrite policy pol2 "CLIENT.SSL.CLIENT_CERT.EXISTS" act2

  Bind these new policies to the NetScaler VIP to put them into effect.

Application Firewall Policy to Protect a Shopping Cart Application

Updated: 2013-09-02

Shopping cart applications handle sensitive customer information, for example, credit card numbers and expiration dates, and they access back-end database servers. Many shopping cart applications also use legacy CGI scripts, which can contain security flaws that were unknown at the time they were written, but are now known to hackers and identity thieves.

A shopping cart application is particularly vulnerable to the following attacks:

- **Cookie tampering.** If a shopping cart application uses cookies, and does not perform the appropriate checks on the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cookies that users return to the application, an attacker could modify a cookie and gain access to the shopping cart application under another user's credentials. Once logged on as that user, the attacker could obtain sensitive private information about the legitimate user or place orders using the legitimate user's account.

- **SQL injection.** A shopping cart application normally accesses a back-end database server. Unless the application performs the appropriate safety checks on the data users return in the form fields of its Web forms before it passes that information on to the SQL database, an attacker can use a Web form to inject unauthorized SQL commands into the database server. Attackers normally use this type of attack to obtain sensitive private information from the database or modify information in the database.

The following configuration will protect a shopping cart application against these and other attacks.

To protect a shopping cart application by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles, and then click Add.
2. In the Create Application Firewall Profile dialog box, in the Profile Name field, enter shopping_cart.
3. In the Profile Type drop-down list, select Web Application.
4. In the Configure Select Advanced defaults.
5. Click Create and then click Close.
6. In the details view, double-click the new profile.
7. In the Configure Web Application Profile dialog box, configure your new profile as described below:
   1. Click the Checks tab, double-click the Start URL check, and in the Modify Start URL Check dialog box, click the General tab and disable blocking, and enable learning, logging, statistics, and URL closure. Click OK and then click Close.

      Note that if you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

      set appfw profile shopping_cart -startURLAction LEARN LOG STATS -startURLClosure ON

   2. For the Cookie Consistency check and Form Field Consistency checks, disable blocking, and enable learning, logging, statistics, using a similar method to the Modify Start URL Check configuration.

      If you are using the command line, you configure these settings by typing the following commands:

      set appfw profile shopping_cart -cookieConsistencyAction LEARN LOG STATS

      set appfw profile shopping_cart -fieldConsistencyAction LEARN LOG STATS

   3. For the SQL Injection check, disable blocking, and enable learning, logging, statistics, and transformation of special characters in the Modify SQL Injection Check dialog box, General tab, Check Actions section.

      If you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

      set appfw profile shopping_cart -SQLInjectionAction LEARN LOG STATS -SQLInjectionTransformSpecialChars ON

   4. For the Credit Card check, disable blocking; enable logging, statistics, and masking of credit card numbers; and enable protection for those credit cards you accept as forms of payment.

      - If you are using the configuration utility, you configure blocking, logging, statistics, and masking (or x-out) in the Modify Credit Card Check dialog box, General tab, Check Actions section. You configure protection for specific credit cards in the Settings tab of the same dialog box.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- If you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

  set appfw profile shopping_cart -creditCardAction LOG STATS -creditCardXOut ON -creditCard <name> [<name>...]

  For <name> you substitute the name of the credit card you want to protect. For Visa, you substitute VISA. For Master Card, you substitute MasterCard. For American Express, you substitute Amex. For Discover, you substitute Discover. For Diners Club, you substitute DinersClub. For JCB, you substitute JCB.

8. Create a policy named shopping_cart that detects connections to your shopping cart application and applies the shopping_cart profile to those connections.

   To detect connections to the shopping cart, you examine the URL of incoming connections. If you host your shopping cart application on a separate host (a wise measure for security and other reasons), you can simply look for the presence of that host in the URL. If you host your shopping cart in a directory on a host that handles other traffic, as well, you must determine that the connection is going to the appropriate directory and/or HTML page.

   The process for detecting either of these is the same; you create a policy based on the following expression, and substitute the proper host or URL for <string>.

   REQ.HTTP.HEADER URL CONTAINS <string>

   - If you are using the configuration utility, you navigate to the application firewall Policies page, click the Add... button to add a new policy, and follow the policy creation process described in “To create a policy with classic expressions using the configuration utility” beginning on page 201 and following.
   - If you are using the command line, you type the following command at the prompt and press Enter:
     add appfw policy shopping_cart "REQ.HTTP.HEADER URL CONTAINS <string>" shopping_cart

9. Globally bind your new policy to put it into effect.

Because you want to ensure that this policy will match all connections to the shopping cart, and not be preempted by another more general policy, you should assign a high priority to it. If you assign one (1) as the priority, no other policy can preempt this one.

Application Firewall Policy to Protect Scripted Web Pages

Updated: 2013-11-14

Web pages with embedded scripts, especially legacy JavaScripts, often violate the “same origin rule,” which does not allow scripts to access or modify content on any server but the server where they are located. This security vulnerability is called cross-site scripting. The application firewall Cross-Site Scripting rule normally filters out requests that contain cross-site scripting.

Unfortunately, this can cause Web pages with older JavaScripts to stop functioning, even when your system administrator has checked those scripts and knows that they are safe. The example below explains how to configure the application firewall to allow cross-site scripting in Web pages from trusted sources without disabling this important filter for the rest of your Web sites.

To protect Web pages with cross-site scripting by using the command line interface

- At the command line, to create an advanced profile, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add appfw profile pr_xssokay -defaults advanced

- To configure the profile, type:

  set appfw profile pr_xssokay -startURLAction NONE -startURLClosure OFF -cookieConsistencyAction LEARN LOG STATS -fieldConsistencyAction LEARN LOG STATS -crossSiteScriptingAction LEARN LOG STATS$"

- Create a policy that detects connections to your scripted Web pages and applies the pr_xssokay profile, type:

  add appfw policy pol_xssokay "REQ.HTTP.HEADER URL CONTAINS ^\.pl\?$ || REQ.HTTP.HEADER URL CONTAINS ^\.js$" pr_xssokay

- Globally bind the policy.

To protect Web pages with cross-site scripting by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the details view, click Add.
3. In the Create Application Firewall Profile dialog box, create a Web Application profile with advanced defaults and name it pr_xssokay. Click Create and then click Close.
4. In the details view, click the profile, click Open, and in the Configure Web Application Profile dialog box, configure the pr_xssokay profile as shown below.

   Start URL Check: Clear all actions.

   - Cookie Consistency Check: Disable blocking.
   - Form Field Consistency Check: Disable blocking.
   - Cross-Site Scripting Check: Disable blocking.

   This should prevent blocking of legitimate requests involving Web pages with cross-site scripting that you know are nonetheless safe.

5. Click Policies, and then click Add.
6. In the Create Application Firewall Policy dialog box, create a policy that detects connections to your scripted Web pages and applies the pr_xssokay profile:

   - Policy name: pol_xssokay
   - Associated profile: pr_xssokay

   Policy expression: "REQ.HTTP.HEADER URL CONTAINS ^\.pl\?$ || REQ.HTTP.HEADER URL CONTAINS ^\.js$"

7. Globally bind your new policy to put it into effect.

DNS Policy to Drop Packets from Specific IPs

Updated: 2013-09-02

The following example describes how to create a DNS action and DNS policy that detects connections from unwanted IPs or networks, such as those used in a DDOS attack, and drops all packets from those locations. The example shows networks within the IANA reserved IP block 192.168.0.0/16. A hostile network will normally be on publicly routable IPs.

To drop packets from specific IPs by using the command line interface

- To create a DNS policy named pol_ddos_drop that detects connections from hostile networks and drops those packets,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

type:

add dns policy pol_ddos_drop 'client.ip.src.in_subnet(192.168.253.128/25) || client.ip.src.in_subnet(192.168.254.32/27)' -drop YES'

For the example networks in the 192.168.0.0/16 range, you substitute the IP and netmask in ###.###.###.###/## format of each network you want to block. You can include as many networks as you want, separating each CLIENT.IP.SRC.IN_SUBNET(###.###.###.###./##) command with the OR operator.

- Globally bind your new policy to put it into effect.

SSL Policy to Require Valid Client Certificates

Updated: 2013-09-02

The following example shows an SSL policy that checks the user's client certificate validity before initiating an SSL connection with a client.

To block connections from users with expired client certificates

- Log on to the command line interface.
  If you are using the GUI, navigate to the SSL Policies page, then in the Data area, click the Actions tab.
- Create an SSL action named act_current_client_cert that requires that users have a current client certificate to establish an SSL connection with the NetScaler.

  add ssl action act_current_client_cert-clientAuth DOCLIENTAUTH -clientCert ENABLED -certHeader "clientCertificateHeader" -clientCertNotBefore ENABLED -certNotBeforeHeader "Mon, 01 Jan 2007 00:00:00 GMT"
- Create an SSL policy named pol_current_client_cert that detects connections to the Web server that contain a query string.

  add ssl policy pol_current_ client_cert 'REQ.SSL.CLIENT.CERT.VALIDFROM >= "Mon, 01 Jan 2007 00:00:00 GMT"' act_block_ssl
- Bind your new policy globally.

  Because this SSL policy should apply to any user's SSL connection unless a more specific SSL policy applies, you may want to assign it a low priority. If you assign it a priority of one thousand (1000), that should ensure that other SSL policies are evaluated first, meaning that this policy will apply only to connections that do not match more specific policy criteria.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Migration of Apache mod_rewrite Rules to the Default Syntax

May 26, 2015

The Apache HTTP Server provides an engine known as mod_rewrite for rewriting HTTP request URLs. If you migrate the mod_rewrite rules from Apache to the NetScaler, you boost back-end server performance. In addition, because the NetScaler typically load balances multiple (sometimes thousands of) Web servers, after migrating the rules to the NetScaler you will have a single point of control for these rules.

Following are examples of mod_rewrite functions, and translations of these functions into Rewrite and Responder policies on the NetScaler.

This document includes the following details:

- Converting URL Variations into Canonical URLs
- Converting Host Name Variations to Canonical Host Names
- Moving a Document Root
- Moving Home Directories to a New Web Server
- Working with Structured Home Directories
- Redirecting Invalid URLs to Other Web Servers
- Rewriting a URL Based on Time
- Redirecting to a New File Name (Invisible to the User)
- Redirecting to New File Name (User-Visible URL)
- Accommodating Browser Dependent Content
- Blocking Access by Robots
- Blocking Access to Inline Images
- Creating Extensionless Links
- Redirecting a Working URI to a New Format
- Ensuring That a Secure Server Is Used for Selected Pages

Converting URL Variations into Canonical URLs

On some Web servers you can have multiple URLs for a resource. Although the canonical URLs should be used and distributed, other URLs can exist as shortcuts or internal URLs. You can make sure that users see the canonical URL regardless of the URL used to make an initial request.

In the following examples, the URL /~user is converted to /u/user.

Apache mod_rewrite solution for converting a URL

```
RewriteRule   ^/~([^/]+)/?(.*)    /u/$1/$2[R]
```

NetScaler solution for converting a URL

```
add responder action act1 redirect '"/u/"+HTTP.REQ.URL.AFTER_STR("/~")' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.STARTSWITH("/~") && HTTP.REQ.URL.LENGTH.GT(2)' act1
bind responder global pol1 100
```

Converting Host Name Variations to Canonical Host Names

You can enforce the use of a particular host name for reaching a site. For example, you can enforce the use of www.example.com instead of example.com.

Apache mod_rewrite solution for enforcing a particular host name for sites running on a port other than 80

```
RewriteCond %{HTTP_HOST}   !^www.example.com
RewriteCond %{HTTP_HOST}   !^$
RewriteCond %{SERVER_PORT} !^80$
RewriteRule ^/(.*)        http://www.example.com:%{SERVER_PORT}/$1 [L,R]
```

Apache mod_rewrite solution for enforcing a particular host name for sites running on port 80

```
RewriteCond %{HTTP_HOST}   !^www.example.com
RewriteCond %{HTTP_HOST}   !^$
RewriteRule ^/(.*)        http://www.example.com/$1 [L,R]
```

NetScaler solution for enforcing a particular host name for sites running on a port other than 80

```
add responder action act1 redirect '"http://www.example.com:"+CLIENT.TCP.DSTPORT+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HOSTNAME.CONTAINS("www.example.com")&&!HTTP.REQ.HOSTNAME.EQ("")&&!HTTP.REQ.HOSTNAME.PORT.EQ(80)&&HTTP.REQ.HOSTNAME.CONT
bind responder global pol1 100 END
```

NetScaler solution for enforcing a particular host name for sites running on port 80

```
add responder action act1 redirect '"http://www.example.com"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1  '!HTTP.REQ.HOSTNAME.CONTAINS("www.example.com")&&!HTTP.REQ.HOSTNAME.EQ("")&&HTTP.REQ.HOSTNAME.PORT.EQ(80)&&HTTP.REQ.HOSTNAME.CONT
bind responder global  pol1 100 END
```

Moving a Document Root

Usually the document root of a Web server is based on the URL “/”. However, the document root can be any directory. You can redirect traffic to the document root if it changes from the top-level “/” directory to another directory.

In the following examples, you change the document root from / to /e/www. The first two examples simply replace one string with another. The third example is more universal because, along with replacing the root directory, it preserves the rest of the URL (the path and query string), for example, redirecting /example/file.html to /e/www/example/file.html.

Apache mod_rewrite solution for moving the document root

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
RewriteEngine on
RewriteRule   ^/$  /e/www/  [R]
```

NetScaler solution for moving the document root

```
add responder action act1 redirect '"/e/www/"' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.EQ("/")' act1
bind responder global pol1 100
```

NetScaler solution for moving the document root and appending path information to the request

```
add responder action act1 redirect '"/e/www"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.URL.STARTSWITH("/e/www/")' act1
bind responder global pol1 100 END
```

Moving Home Directories to a New Web Server

You may want to redirect requests that are sent to home directories on a Web server to a different Web server. For example, if a new Web server is replacing an old one over time, as you migrate home directories to the new location you need to redirect requests for the migrated home directories to the new Web server.

In the following examples, the host name for the new Web server is newserver.

Apache mod_rewrite solution for redirecting to another Web server

```
RewriteRule   ^/(.+)  http://newserver/$1     [R,L]
```

NetScaler solution for redirecting to another Web server (method 1)

```
add responder  action act1 redirect '"http://newserver"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.REGEX_MATCH(re#^/(.+)#)'   act1
bind responder global pol1 100 END
```

NetScaler solution for redirecting to another Web server (method 2)

```
add responder  action act1 redirect '"http://newserver"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.LENGTH.GT(1)' act1
bind responder global pol1 100 END
```

Working with Structured Home Directories

Typically, a site with thousands of users has a structured home directory layout. For example, each home directory may reside under a subdirectory that is named using the first character of the user name. For example, the home directory for jsmith (/~jsmith/anypath) might be /home/j/smith/.www/anypath, and the home directory for rvalveti (/~rvalveti/anypath) might be /home/r/rvalveti/.www/anypath.

The following examples redirect requests to the home directory.

Apache mod_rewrite solution for structured home directories

```
RewriteRule   ^/~(([a-z])[a-z0-9]+)(.*)  /home/$2/$1/.www$3
```

NetScaler solution for structured home directories

NetScaler solution for structured home directories

```
add rewrite action act1 replace 'HTTP.REQ.URL'  '"/home/"+ HTTP.REQ.URL.AFTER_STR("~").PREFIX(1)+"/"+ HTTP.REQ.URL.AFTER_STR("~").BEFORE_STR("/")+"/.www"+HTTP.REQ.URL.SKIP(
add rewrite policy pol1  'HTTP.REQ.URL.PATH.STARTSWITH("/~")' act1
bind rewrite global pol1 100
```

Redirecting Invalid URLs to Other Web Servers

If a URL is not valid, it should be redirected to another Web server. For example, you should redirect to another Web server if a file that is named in a URL does not exist on the server that is named in the URL.

On Apache, you can perform this check using mod_rewrite. On the NetScaler, an HTTP callout can check for a file on a server by running a script on the server. In the following NetScaler examples, a script named file_check.cgi processes the URL and uses this information to check for the presence of the target file on the server. The script returns TRUE or FALSE, and the NetScaler uses the value that the script returns to validate the policy.

In addition to performing the redirection, the NetScaler can add custom headers or, as in the second NetScaler example, it can add text in the response body.

Apache mod_rewrite solution for redirection if a URL is wrong

```
RewriteCond   /your/docroot/%{REQUEST_FILENAME} !-f
RewriteRule   ^(.+)      http://webserverB.com/$1 [R]
```

NetScaler solution for redirection if a URL is wrong (method 1)

```
add HTTPCallout Call
set policy httpCallout Call -IPAddress 10.102.59.101 -port 80 -hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cg
add responder action act1 redirect '"http://webserverB.com"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HEADER("Name").EXISTS  &&  !SYS.HTTP_CALLOUT(call)' act1
bind responder global pol1 100
```

NetScaler solution for redirection if a URL is wrong (method 2)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add HTTPCallout Call
set policy httpCallout Call -IPAddress 10.102.59.101 -port 80 -hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cg
add responder  action act1 respondwith  '"HTTP/1.1 302 Moved Temporarily\r\nLocation: http://webserverB.com"+HTTP.REQ.URL+"\r\n\r\nHTTPCallout Used"' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HEADER("Name").EXISTS  &&  !SYS.HTTP_CALLOUT(call)' act1
bind responder global pol1 100
```

Rewriting a URL Based on Time

You can rewrite a URL based on the time. The following examples change a request for example.html to example.day.html or example.night.html, depending on the time of day.

Apache mod_rewrite solution for rewriting a URL based on the time

```
RewriteCond   %{TIME_HOUR}%{TIME_MIN} >0700
RewriteCond   %{TIME_HOUR}%{TIME_MIN} <1900
RewriteRule   ^example\.html$ example.day.html [L]
RewriteRule   ^example\.html$ example.night.html
```

NetScaler solution for rewriting a URL based on the time

```
add rewrite action act1 insert_before 'HTTP.REQ.URL.PATH.SUFFIX(\'.\',0)' '"day."'
add rewrite action act2  insert_before 'HTTP.REQ.URL.PATH.SUFFIX(\'.\',0)' '"night."'
add rewrite  policy pol1 'SYS.TIME.WITHIN(LOCAL 07h 00m,LOCAL 18h 59m)' act1
add rewrite policy pol2 'true'  act2
bind rewrite global pol1 101
bind rewrite global pol2 102
```

Redirecting to a New File Name (Invisible to the User)

If you rename a Web page, you can continue to support the old URL for backward compatibility while preventing users from recognizing that the page was renamed.

In the first two of the following examples, the base directory is /~quux/. The third example accommodates any base directory and the presence of query strings in the URL.

Apache mod_rewrite solution for managing a file name change in a fixed location

```
RewriteEngine  on
RewriteBase    /~quux/
RewriteRule    ^foo\.html$  bar.html
```

NetScaler solution for managing a file name change in a fixed location

```
add rewrite action act1 replace 'HTTP.REQ.URL.AFTER_STR("/~quux").SUBSTR("foo.html")' '"bar.html"'
add rewrite policy pol1 'HTTP.REQ.URL.ENDSWITH("/~quux/foo.html")' act1
bind rewrite global pol1 100
```

NetScaler solution for managing a file name change regardless of the base directory or query strings in the URL

```
add rewrite action act1 replace 'HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' '"bar.html"'
Add rewrite policy pol1 'HTTP.REQ.URL.PATH.CONTAINS("foo.html")' act1
Bind rewrite global pol1 100
```

Redirecting to New File Name (User-Visible URL)

If you rename a Web page, you may want to continue to support the old URL for backward compatibility and allow users to see that the page was renamed by changing the URL that is displayed in the browser.

In the first two of the following examples, redirection occurs when the base directory is /~quux/. The third example accommodates any base directory and the presence of query strings in the URL.

Apache mod_rewrite solution for changing the file name and the URL displayed in the browser

```
RewriteEngine on
RewriteBase    /~quux/
RewriteRule    ^old\.html$ new.html  [R]
```

NetScaler solution for changing the file name and the URL displayed in the browser

```
add responder action act1 redirect 'HTTP.REQ.URL.BEFORE_STR("foo.html")+"new.html"' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.ENDSWITH("/~quux/old.html")' act1
bind responder global pol1 100
```

NetScaler solution for changing the file name and the URL displayed in the browser regardless of the base directory or query strings in the URL

```
add responder action act1 redirect 'HTTP.REQ.URL.PATH.BEFORE_STR("old.html")+"new.html"+HTTP.REQ.URL.AFTER_STR("old.html")' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.PATH.CONTAINS("old.html")' act1
bind responder global pol1 100
```

Accommodating Browser Dependent Content

To accommodate browser-specific limitations—at least for important top-level pages—it is sometimes necessary to set restrictions on the browser type and version. For example, you might want to set a maximum version for the latest Netscape variants, a minimum version for Lynx browsers, and an average feature version for all others.

The following examples act on the HTTP header "User-Agent", such that if this header begins with "Mozilla/3", the page MyPage.html is rewritten to MyPage.NS.html. If the browser is "Lynx" or "Mozilla" version 1 or 2, the URL becomes MyPage.20.html. All other browsers receive page MyPage.32.html.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Apache mod_rewrite solution for browser-specific settings

```
RewriteCond %{HTTP_USER_AGENT}  ^Mozilla/3.*
RewriteRule ^MyPage\.html$ MyPage.NS.html [L]
RewriteCond %{HTTP_USER_AGENT}  ^Lynx/.* [OR]
RewriteCond %{HTTP_USER_AGENT}  ^Mozilla/[12].*
RewriteRule ^MyPage\.html$ MyPage.20.html [L]
RewriteRule ^fMyPage\.html$ MyPage.32.html [L]
```

NetScaler solution for browser-specific settings

```
add patset pat1
bind patset pat1 Mozilla/1
bind Patset pat1 Mozilla/2
bind patset pat1 Lynx
bind Patset pat1 Mozilla/3
add rewrite action act1 insert_before 'HTTP.REQ.URL.SUFFIX' '"NS."'
add rewrite action act2 insert_before 'HTTP.REQ.URL.SUFFIX' '"20."'
add rewrite action act3 insert_before 'HTTP.REQ.URL.SUFFIX' '"32."'
add rewrite policy pol1 'HTTP.REQ.HEADER("User-Agent").STARTSWITH_INDEX("pat1").EQ(4)' act1
add rewrite policy pol2 'HTTP.REQ.HEADER("User-Agent").STARTSWITH_INDEX("pat1").BETWEEN(1,3)' act2
add rewrite policy pol3 '!HTTP.REQ.HEADER("User-Agent").STARTSWITH_ANY("pat1")' act3
bind rewrite global pol1 101 END
bind rewrite global pol2 102 END
bind rewrite global pol3 103 END
```

Blocking Access by Robots

You can block a robot from retrieving pages from a specific directory or a set of directories to ease up the traffic to and from these directories. You can restrict access based on the specific location or you can block requests based on information in User-Agent HTTP headers.

In the following examples, the Web location to be blocked is /~quux/foo/arc/, the IP addresses to be blocked are 123.45.67.8 and 123.45.67.9, and the robot's name is NameOfBadRobot.

Apache mod_rewrite solution for blocking a path and a User-Agent header

```
RewriteCond %{HTTP_USER_AGENT}   ^NameOfBadRobot.*
RewriteCond %{REMOTE_ADDR}       ^123\.45\.67\.[8-9]$
RewriteRule ^/~quux/foo/arc/.+   -   [F]
```

NetScaler solution for blocking a path and a User-Agent header

```
add responder action act1 respondwith '"HTTP/1.1 403 Forbidden\r\n\r\n"'
add responder policy pol1 'HTTP.REQ.HEADER("User_Agent").STARTSWITH("NameOfBadRobot")&&CLIENT.IP.SRC.EQ(123.45.67.8)&&CLIENT.IP.SRC.EQ(123.45.67.9) && HTTP.REQ.URL.STAF
bind responder global pol1 100
```

Blocking Access to Inline Images

If you find people frequently going to your server to copy inline graphics for their own use (and generating unnecessary traffic), you may want to restrict the browser's ability to send an HTTP Referer header.

In the following example, the graphics are located in http://www.quux-corp.de/~quux/.

Apache mod_rewrite solution for blocking access to an inline image

```
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://www.quux-corp.de/~quux/.*$
RewriteRule .*\.gif$ - [F]
```

NetScaler solution for blocking access to an inline image

```
add patset pat1
bind patset pat1 .gif
bind patset pat1 .jpeg
add responder action act1 respondwith '"HTTP/1.1 403 Forbidden\r\n\r\n"'
add responder policy pol1 '!HTTP.REQ.HEADER("Referer").EQ("") && !HTTP.REQ.HEADER("Referer").STARTSWITH("http://www.quux-corp.de/~quux/")&&HTTP.REQ.URL.ENDSWITH_ANY("pat1")'
bind responder global pol1 100
```

Creating Extensionless Links

To prevent users from knowing application or script details on the server side, you can hide file extensions from users. To do this, you may want to support extensionless links. You can achieve this behavior by using rewrite rules to add an extension to all requests, or to selectively add extensions to requests.

The first two of the following examples show adding an extension to all request URLs. In the last example, one of two file extensions is added. Note that in the last example, the mod_rewrite module can easily find the file extension because this module resides on the Web server. In contrast, the NetScaler must invoke an HTTP callout to check the extension of the requested file on the Web server. Based on the callout response, the NetScaler adds the .html or .php extension to the request URL.

Note: In the second NetScaler example, an HTTP callout is used to query a script named file_check.cgi hosted on the server. This script checks whether the argument that is provided in the callout is a valid file name.

Apache mod_rewrite solution for adding a .php extension to all requests

```
RewriteRule ^/?([a-z]+)$ $1.php [L]
```

NetScaler policy for adding a .php extension to all requests

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add rewrite action act1 insert_after 'HTTP.REQ.URL' '".php"'
add rewrite policy pol1 'HTTP.REQ.URL.PATH.REGEX_MATCH(re#^/([a-z]+)$#)' act1
bind rewrite global pol1 100
```

Apache mod_rewrite solution for adding either .html or .php extensions to requests

```
RewriteCond %{REQUEST_FILENAME}.php -f
RewriteRule ^/?([a-zA-Z0-9]+)$ $1.php [L]
RewriteCond %{REQUEST_FILENAME}.html –f
RewriteRule ^/?([a-zA-Z0-9]+)$ $1.html [L]
```

NetScaler policy for adding either .html or .php extensions to requests

```
add HTTPCallout Call_html
add HTTPCallout Call_php
set policy httpCallout Call_html -IPAddress 10.102.59.101 -port 80 -hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_che
set policy httpCallout Call_php -IPAddress 10.102.59.101 -port 80 -hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_che
add patset pat1
bind patset pat1 .html
bind patset pat1 .php
bind patset pat1 .asp
bind patset pat1 .cgi
add rewrite  action act1 insert_after 'HTTP.REQ.URL.PATH'  '".html"'
add rewrite  action act2 insert_after "HTTP.REQ.URL.PATH"  '".php"'
add rewrite policy pol1 '!HTTP.REQ.URL.CONTAINS_ANY("pat1") && SYS.HTTP_CALLOUT(Call_html)' act1
add rewrite policy pol2 '!HTTP.REQ.URL.CONTAINS_ANY("pat1") && SYS.HTTP_CALLOUT(Call_php)' act2
bind rewrite global pol1 100 END
bind rewrite global pol2 101 END
```

Redirecting a Working URI to a New Format

Suppose that you have a set of working URLs that resemble the following:

/index.php?id=nnnn

To change these URLs to /nnnn and make sure that search engines update their indexes to the new URI format, you need to do the following:

- Redirect the old URIs to the new ones so that search engines update their indexes.
- Rewrite the new URI back to the old one so that the index.php script runs correctly.

To accomplish this, you can insert marker code into the query string (making sure that the marker code is not seen by visitors), and then removing the marker code for the index.php script.

The following examples redirect from an old link to a new format only if a marker is not present in the query string. The link that uses the new format is re-written back to the old format, and a marker is added to the query string.

Apache mod_rewrite solution

```
RewriteCond %{QUERY_STRING} !marker
RewriteCond %{QUERY_STRING} id=([-a-zA-Z0-9_+]+)
RewriteRule ^/?index\.php$ %1? [R,L]
RewriteRule ^/?([-a-zA-Z0-9_+]+)$  index.php?marker&id=$1 [L]
NetScaler solution
add responder action act_redirect redirect 'HTTP.REQ.URL.PATH.BEFORE_STR("index.php")+HTTP.REQ.URL.QUERY.VALUE("id")' -bypassSafetyCheck yes
add responder policy pol_redirect '!HTTP.REQ.URL.QUERY.CONTAINS("marker")&& HTTP.REQ.URL.QUERY.VALUE("id").REGEX_MATCH(re/[-a-zA-Z0-9_+]+/) && HTTP.REQ.URL.PATH.CONTAIN
bind responder global pol_redirect 100 END
add rewrite action act1 replace 'HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' '"index.phpmarker&id="+HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' -bypassSafetyCheck yes
add rewrite policy pol1 '!HTTP.REQ.URL.QUERY.CONTAINS("marker")'  act1
bind rewrite global pol1 100 END
```

Ensuring That a Secure Server Is Used for Selected Pages

To make sure that only secure servers are used for selected Web pages, you can use the following Apache mod_rewrite code or NetScaler Responder policies.

Apache mod_rewrite solution

```
RewriteCond %{SERVER_PORT} !^443$
RewriteRule ^/?(page1|page2|page3|page4|page5)$  https://www.example.com/%1 [R,L]
```

NetScaler solution using regular expressions

```
add responder action res_redirect redirect  '"https://www.example.com"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol_redirect '!CLIENT.TCP.DSTPORT.EQ(443)&&HTTP.REQ.URL.REGEX_MATCH(re/page[1-5]/)'  res_redirect
bind responder global pol_redirect 100 END
```

NetScaler solution using pattern sets

```
add patset pat1
bind patset pat1 page1
bind patset pat1 page2
bind patset pat1 page3
bind patset pat1 page4
bind patset pat1 page5
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add responder action res_redirect redirect  '"https://www.example.com"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol_redirect '!CLIENT.TCP.DSTPORT.EQ(443)&&HTTP.REQ.URL.CONTAINS_ANY("pat1")'  res_redirect
bind responder global pol_redirect 100 END
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rate Limiting

Aug 12, 2014

The rate limiting feature enables you to define the maximum load for a given network entity or virtual entity on the Citrix NetScaler appliance. The feature enables you to configure the appliance to monitor the rate of traffic associated with the entity and take preventive action, in real time, based on the traffic rate. This feature is particularly useful when the network is under attack from a hostile client that is sending the appliance a flood of requests. You can mitigate the risks that affect the availability of resources to clients, and you can improve the reliability of the network and the resources that the appliance manages.

You can monitor and control the rate of traffic that is associated with virtual and user-defined entities, including virtual servers, URLs, domains, and combinations of URLs and domains. You can throttle the rate of traffic if it is too high, base information caching on the traffic rate, and redirect traffic to a given load balancing virtual server if the traffic rate exceeds a predefined limit. You can apply rate-based monitoring to HTTP, TCP, and DNS requests.

To monitor the rate of traffic for a given scenario, you configure a *rate limit identifier*. A rate limit identifier specifies numeric thresholds such as the maximum number of requests or connections (of a particular type) that are permitted in a specified time period called a *time slice*.

Optionally, you can configure filters, known as *stream selectors*, and associate them with rate limit identifiers when you configure the identifiers. After you configure the optional stream selector and the limit identifier, you must invoke the limit identifier from a default syntax policy. You can invoke identifiers from any feature in which the identifier may be useful, including rewrite, responder, DNS, and integrated caching.

You can globally enable and disable SNMP traps for rate limit identifiers. Each trap contains cumulative data for the rate limit identifier's configured data collection interval (time slice), unless you specified multiple traps to be generated per time slice. For more information about configuring SNMP traps and managers, see "SNMP."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Stream Selector

Jul 30, 2014

A traffic stream selector is an optional filter for identifying an entity for which you want to throttle access. The selector is applied to a request or a response and selects data points (keys) that can be analyzed by a rate stream identifier. These data points can be based on almost any characteristic of the traffic, including IP addresses, subnets, domain names, TCP or UDP identifiers, and particular strings or extensions in URLs.

A stream selector consists of individual default syntax expressions called selectlets. Each selectlet is a non-compound default syntax expression. A traffic stream selector can contain up to five non-compound expressions called selectlets. Each selectlet is considered to be in an AND relationship with the other expressions. Following are some examples of selectlets:

```
http.req.url
http.res.body(1000>after_str(\"car_model\").before_str(\"made_in\")"
"client.ip.src.subnet(24)"
```

The order in which you specify parameters is significant. For example, if you configure an IP address and a domain (in that order) in one selector, and then specify the domain and the IP address (in the reverse order) in another selector, the NetScaler considers these values to be unique. This can lead to the same transaction being counted twice. Also, if multiple policies invoke the same selector, the NetScaler, again, can count the same transaction more than once.

Note: If you modify an expression in a stream selector, you may get an error if any policy that invokes it is bound to a new policy label or bind point. For example, suppose that you create a stream selector named myStreamSelector1, invoke it from myLimitID1, and invoke the identifier from a DNS policy named dnsRateLimit1. If you change the expression in myStreamSelector1, you might receive an error when binding dnsRateLimit1 to a new bind point. The workaround is to modify these expressions before creating the policies that invoke them.

To configure a traffic stream selector by using the command line interface

At the command prompt, type:

add stream selector <name> <rule> ...

Example

```
> add stream selector myStreamSel HTTP.REQ.URL CLIENT.IP.SRC
```

To configure a stream selector by using the configuration utility

Navigate to AppExpert > Rate Limiting > Selectors, click Add and specify the relevant details.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Traffic Rate Limit Identifier

Oct 16, 2017

A rate limit identifier returns a Boolean TRUE if the amount of traffic exceeds a numeric limit within a particular time interval. The rate limit identifier definition can optionally include a stream selector. When you include a limit identifier in the compound default syntax expression in a policy rule, if you do not specify a stream selector, the limit identifier is applied to all the requests or responses that are identified by the compound expression.

Note: The maximum length for storing string results of selectors (for example, HTTP.REQ.URL) is 60 characters. If the string (for example, URL) is 1000 characters long, of which 50 characters are enough to uniquely identify a string, use an expression to extract only the required 50 characters.

To configure a traffic limit identifier from the command line interface

At the command prompt, type:

add ns limitIdentifier <limitIdentifier> -threshold <positive_integer> -timeSlice <positive_integer> -mode <mode> -limitType (BURSTY | SMOOTH) -selectorName <string> -maxBandwidth <positive_integer> -trapsInTimeSlice <positive_integer>

Argument description

limitIdentifier. Name for a rate limit identifier. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Reserved words must not be used. This is a mandatory argument. Maximum Length: 31

threshold. A maximum number of requests that are allowed in the given timeslice when requests (mode is set as REQUEST_RATE) are tracked per timeslice. When connections (mode is set as CONNECTION) are tracked, it is the total number of connections that would be let through. Default value: 1 Minimum value: 1 Maximum Value: 4294967295

timeSlice. Time interval, in milliseconds, specified in multiples of 10, during which requests are tracked to check if they cross the threshold. This argument is needed only when the mode is set to REQUEST_RATE. Default value: 1000 Minimum value: 10 Maximum Value: 4294967295

mode. Defines the type of traffic to be tracked. * REQUEST_RATE - Tracks requests/timeslice. * CONNECTION - Tracks active transactions.

limitType. Smooth or bursty request type.

selectorName. Name of the rate limit selector. If this argument is NULL, rate limiting will be applied on all traffic received by the virtual server or the NetScaler (depending on whether the limit identifier is bound to a virtual server or globally) without any **filtering. Maximum Length: 31**

maxBandwidth. Maximum bandwidth permitted, in kbps. Minimum value: 0 Maximum value: 4294967287

Example

Configuring traffic rate limit identifier in BURSTY mode:

```
> add ns limitIdentifier 100_request_limit -threshold 100 -timeSlice 1000 -mode REQUEST_RATE -limitType BURSTY -selectorName limit_100_requests_selector -trapsInTimeSlice 30
```

Configuring traffic rate limit identifier in SMOOTH mode:

```
> add ns limitidentifier limit_req -mode request_rate -limitType smooth -timeslice 1000 -Threshold 2000 -trapsInTimeSlice 200
```

To configure a traffic limit identifier by using the configuration utility

Navigate to AppExpert > Rate Limiting > Limit Identifiers, click Add and specify the relevant details.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring and Binding a Traffic Rate Policy

Aug 30, 2013

You implement rate-based application behavior by configuring a policy in an appropriate NetScaler feature. The feature must support default syntax policies. The policy expression must contain the following expression prefix to enable the feature to analyze the traffic rate:

sys.check_limit(<limit_identifier>)

Where limit_identifier is the name of a limit identifier.

The policy expression must be a compound expression that contains at least two components:

- An expression that identifies traffic to which the rate limit identifier is applied. For example: http.req.url.contains("my_aspx.aspx").
- An expression that identifies a rate limit identifier, for example, sys.check_limit("my_limit_identifier"). This must be the last expression in the policy expression.

To configure a rate-based policy by using the command line interface

At the command prompt, type the following command to configure a rate-based policy and verify the configuration:

add cache | dns | rewrite | responder policy <policy_name> -rule expression && sys.check_limit("<LimitIdentifierName>") [<feature-specific information>]

Following is a complete example of a rate-based policy rule. Note that this example assumes that you have configured the responder action, send_direct_url, that is associated with the policy. Note that the sys.check_limit parameter must be the last element of the policy expression:

```
add responder policy responder_threshold_policy "http.req.url.contains(\"myindex.html\") && sys.check_limit(\"my_limit_identifier\")" send_direct_url
```

For information about binding a policy globally or to a virtual server, see "Binding Default Syntax Policies."

To configure a rate-based policy by using the configuration utility

1. In the navigation pane, expand the feature in which you want to configure a policy (for example, Integrated Caching, Rewrite, or Responder), and then click Policies.
2. In the details pane, click Add. In Name, enter a unique name for the policy.
3. Under Expression, enter the policy rule, and make sure that you include the sys.check_limit parameter as the final component of the expression. For example:

   http.req.url.contains("my_aspx.aspx") && sys.check_limit("my_limit_identifier")
4. Enter feature-specific information about the policy.

   For example, you may be required to associate the policy with an action or a profile. For more information, see the feature-specific documentation.
5. Click Create, and then click Close.
6. Click Save.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing the Traffic Rate

Aug 30, 2013

If traffic through one or more virtual servers matches a rate-based policy, you can view the rate of this traffic. The rate statistics are maintained in the limit identifier that you named in the rule for the rate-based policy. If more than one policy uses the same limit identifier, you can view the traffic rate as defined by hits to all of the policies that use the particular limit identifier.

To view the traffic rate by using the command line interface

At the command prompt, type the following command to view the traffic rate:

show ns limitSessions <limitIdentifier>

Example

```
sh limitsession myLimitSession
```

To view the traffic rate by using the configuration utility

1. Navigate to AppExpert > Rate Limiting > Limit Identifiers.
2. Select a limit identifier whose traffic rate you want to view.
3. Click the Show Sessions button. If traffic through one or more virtual servers has matched a rate limiting policy that uses this limit identifier (and the hits are within the configured time slice for this identifier), the Session Details dialog box appears. Otherwise, you receive a "No session exists" message.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Testing a Rate-Based Policy

Oct 29, 2013

To test a rate-based policy, you can send traffic to any virtual server to which a rate-based policy is bound.

Task overview: Testing a rate-based policy

1. Configure a stream selector (optional) and a rate limit identifier (required). For example:

   ```
   add stream selector sel_subnet Q.URL "CLIENT.IP.SRC.SUBNET(24)"
   add ns limitIdentifier k_subnet -Threshold 4 -timeSlice 3600 -mode REQUEST_RATE  -limittype smooth  -selectorName sel_subnet  -trapsInTimeSlice 8
   ```
2. Configure the action that you want to associate with the policy that uses the rate limit identifier. For example:

   ```
   add responder action resp_redirect redirect "\"http://response_site.com/\""
   ```
3. Configure a policy that uses the sys.check_limit expression prefix to call the rate limit identifier. For example, the policy can apply a rate limit identifier to all requests arriving from a particular subnet, as follows:

   ```
   add responder policy resp_subnet "SYS.CHECK_LIMIT(\"k_subnet\")" resp_redirect
   ```
4. Bind the policy globally or to a virtual server. For example:

   ```
   bind responder global resp_subnet 6 END -type DEFAULT
   ```
5. In a browser address bar, send a test HTTP query to a virtual server. For example:

   ```
   http://<IP of a vserver>/testsite/test.txt
   ```
6. At the NetScaler command prompt, type:

   ```
   show ns limitSessions <limitIdentifier>
   ```

   Example

   ```
   > sh limitsession k_subnet
   1)     Time Remaining:      98 secs  Hits: 2                    Action Taken: 0
         Total Hash:    1718618  Hash String: /test.txt
         IPs gathered:
               1) 10.217.253.0
         Active Transactions: 0
    Done
   >
   ```
7. Repeat the query and check the limit identifier statistics again to verify that the statistics are being updated correctly.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Examples of Rate-Based Policies

May 11, 2017

The following table shows examples of rate-based policies.

Table 1. Examples of Rate-Based Policies

Purpose	Example
Limit the number of requests per second from a URL	add stream selector ipStreamSelector http.req.url "client.ip.src" add ns limitIdentifier ipLimitIdentifier -threshold 4 -timeSlice 1000 -mode request_rate -limitType smooth -selectorName ipStreamSelector add responder action myWebSiteRedirectAction redirect "\"http://www.mycompany.com/\"" add responder policy ipLimitResponderPolicy "http.req.url.contains(\"myasp.asp\") && sys.check_limit(\"ipLimitIdentifier\")" myWebSiteRedirectAction bind responder global ipLimitResponderPolicy 100 END -type default
Cache a response if the request URL rate exceeds 5 per 20000 milliseconds	add stream selector cacheStreamSelector http.req.url add ns limitidentifier cacheRateLimitIdentifier -threshold 5 -timeSlice 2000 -selectorName cacheStreamSelector add cache policy cacheRateLimitPolicy -rule "http.req.method.eq(get) && sys.check_limit(\"cacheRateLimitIdentifier\")" -action cache bind cache global cacheRateLimitPolicy -priority 10
Drop a connection on the basis of cookies received in requests from www.yourcompany.com if the requests exceed the rate limit	add stream selector reqCookieStreamSelector "http.req.cookie .value(\"mycookie\")" "client.ip.src.subnet(24)" add ns limitIdentifier myLimitIdentifier -Threshold 2 -timeSlice 3000 -selectorName reqCookieStreamSelector add responder action sendRedirectUrl redirect '\"http://www.mycompany.com\" + http.req.url' -bypassSafetyCheck YES add responder policy rateLimitCookiePolicy "http.req.url.contains(\"www.yourcompany.com\") && sys.check_limit(\"myLimitIdentifier\")" sendRedirectUrl
Drop a DNS packet if the requests from a particular client IP address and DNS domain exceed the rate limit	add stream selector dropDNSStreamSelector client.udp.dns.domain client.ip.src add ns limitIdentifier dropDNSRateIdentifier -timeslice 20000 -mode request_rate -selectorName dropDNSStreamSelector -maxBandwidth 1 -trapsintimeslice 20

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Purpose	Example
	add dns policy dnsDropOnClientRatePolicy "sys.check_limit (\"dropDNSRateIdentifier\")" -drop yes
Limit the number of HTTP requests that arrive from the same host (with a subnet mask of 32) and that have the same destination IP address.	add stream selector ipv6_sel "CLIENT.IPv6.src.subnet(32)" CLIENT.IPv6.dst Q.URL add ns limitIdentifier ipv6_id -imeSlice 20000 -selectorName ipv6_sel add lb vserver ipv6_vip HTTP 3ffe::209 80 -persistenceType NONE -cltTimeout 180 add responder action redirect_page redirect "\"http://redirectpage.com/\"" add responder policy ipv6_resp_pol "SYS.CHECK_LIMIT(\"ipv6_id\")" redirect_page bind responder global ipv6_resp_pol 5 END -type DEFAULT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Sample Use Cases for Rate-Based Policies

Dec 10, 2013

The following scenarios describe two uses of rate-based policies in global server load balancing (GSLB):

- The first scenario describes the use of a rate-based policy that sends traffic to a new data center if the rate of DNS requests exceed 1000 per second.
- In the second scenario, if more than five DNS requests arrive for a local DNS (LDNS) client within a particular period, the additional requests are dropped.

Redirecting Traffic on the Basis of Traffic Rate

In this scenario, you configure a proximity-based load balancing method, and a rate-limiting policy that identifies DNS requests for a particular region. In the rate-limiting policy, you specify a threshold of 1000 DNS requests per second. A DNS policy applies the rate limiting policy to DNS requests for the region "Europe.GB.17.London.UK-East.ISP-UK." In the DNS policy, DNS requests that exceed the rate limiting threshold, starting with request 1001 and continuing to the end of the one-second interval, are to be forwarded to the IP addresses that are associated with the region "North America.US.TX.Dallas.US-East.ISP-US."

The following configuration demonstrates this scenario:

```
add stream selector DNSSelector1 client.udp.dns.domain
add ns limitIdentifier DNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName DNSSelector1
add dns policy DNSLimitPolicy1 "client.ip.src.matches_location(\"Europe.GB.17.London.*.*\") &&
sys.check_limit(\"DNSLimitIdentifier1\")" -preferredLocation "North America.US.TX.Dallas.*.*"
bind dns global DNSLimitPolicy1 5
```

Dropping DNS Requests on the Basis of Traffic Rate

In the following example of global server load balancing, you configure a rate limiting policy that permits a maximum of five DNS requests in a particular interval, per domain, to be directed to an LDNS client for resolution. Any requests that exceed this rate are dropped. This type of policy can help protect the NetScaler from resource exploitation. For example, in this scenario, if the time to live (TTL) for a connection is five seconds, this policy prevents the LDNS from requerying a domain. Instead, it uses data that is cached on the NetScaler.

```
add stream selector LDNSSelector1 client.udp.dns.domain client.ip.src
add ns limitIdentifier LDNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName LDNSSelector1
add dns policy LDNSPolicy1 "client.udp.dns.domain.contains(\".\") && sys.check_limit(\"LDNSLimitIdentifier1\")" -drop YES
bind dns global LDNSPolicy1 6
show gslb vserver gvip
gvip - HTTP     State: UP
Last state change was at Mon Sep  8 11:50:48 2008 (+711 ms)
Time since last state change: 1 days, 02:55:08.830
Configured Method: STATICPROXIMITY
BackupMethod: ROUNDROBIN
No. of Bound Services :  3 (Total)      3 (Active)
Persistence: NONE      Persistence ID: 100
Disable Primary Vserver on Down: DISABLED      Site Persistence: NONE
Backup Session Timeout: 0
Empty Down Response: DISABLED
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Multi IP Response: DISABLED Dynamic Weights: DISABLED
Cname Flag: DISABLED
Effective State Considered: NONE
1)     site11_svc(10.100.00.00: 80)- HTTP State: UP    Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: Europe.GB.17.London.UK-East.ISP-UK
2)     site12_svc(10.101.00.100: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.TX.Dallas.US-East.ISP-US
3)     site13_svc(10.102.00.200: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.NJ.Salem.US-Mid.ISP-US
1)     www.gslbindia.com        TTL: 5 secn
Cookie Timeout: 0 min   Site domain TTL: 3600 sec
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rate Limiting for Traffic Domains

Jun 19, 2014

You can configure rate limiting for traffic domains. The following expression in the NetScaler expressions language for identifies traffic associated with traffic domains.

- client.traffic_domain.id

You can configure rate limiting for traffic associated with a particular traffic domain, a set of traffic domains, or all traffic domains.

For configuring rate limiting for traffic domains, you perform the following steps on a NetScaler appliance by using the configuration utility or the NetScaler command line:

1. Configure a stream selector that uses the client.traffic_domain.id expression for identifying the traffic, associated with traffic domains, to be rate limited.
2. Configure a rate limit identifier that specifies parameters such as maximum threshold for traffic to be rate limited. You also associate a stream selector to the rate limiter in this step.
3. Configure an action that you want to associate with the policy that uses the rate limit identifier.
4. Configure a policy that uses the sys.check_limit expression prefix to call the rate limit identifier, and associate the action with this policy.
5. Bind the policy globally.

Consider an example in which two traffic domains, with IDs 10 and 20, are configured on NetScaler ADC NS1. On traffic domain 10, LB1-TD-1 is configured to load balance servers S1 and S2; LB2-TD1 is configured to load balance servers S3 and S4.

On traffic domain 20, LB1-TD-2 is configured to load balance servers S5 and S6; LB2-TD2 is configured to load balance servers S7 and S8.

The following table lists some examples of rate limiting policies for traffic domains in the example setup.

Purpose	CLI commands
Limit the number of requests to 10 per second for each of the traffic domains.	add stream selector tdratelimit-1 CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier limitidf-1 -threshold 10 -selectorName tdratelimit-1 -trapsInTimeSlice 0 add responder policy ratelimit-pol "sys.check_limit(\"limitidf-1\")" DROP bind responder global ratelimit-pol 1
Limit the number of requests to 5 per client per second for each of the traffic domains.	add stream selector tdandclientip CLIENT.IP.SRC,CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td_limitidf -threshold 5 -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy tdratelimit-pol "sys.check_limit(\"td_limitidf\")" DROP bind responder global tdratelimit-pol 2
Limit the number of requests sent for a particular traffic domain (for example traffic domain 10) to 30 requests every 3 seconds.	add stream selector tdratelimit CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td10_limitidf -threshold 30 -timeSlice 3000 -selectorName tdratelimit -trapsInTimeSlice 5 add responder policy td10ratelimit "client.traffic_domain.id==10 && sys.check_limit(\"td10_limitidf\")" DROP bind responder global td10ratelimit 3

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Purpose	CLI commands
Limit the number of connections to 5 per client per second for a particular traffic domain (for example traffic domain 20).	add stream selector tdandclientip CLIENT.IP.SRC CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td20_limitidf -threshold 5 -mode CONNECTION -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy td20_ratelimit "client.traffic_domain.id==20 && sys.check_limit(\"td20_limitidf\")" DROP bind responder global td20_ratelimit 4

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Responder

Mar 20, 2012

Today's complex Web configurations often require different responses to HTTP requests that appear, on the surface, to be similar. When users request a Web site's home page, you may want to provide a different home page depending on where each user is located, which browser the user is using, or which language(s) the browser accepts and the order of preference. You might want to break the connection immediately if the request is coming from an IP range that has been generating DDoS attacks or initiating hacking attempts.

With the Responder feature, responses can be based on who sends the request, where it is sent from, and other criteria with security and system management implications. The feature is simple and quick to use. By avoiding the invocation of more complex features, it reduces CPU cycles and time spent in handling requests that do not require complex processing.

For handling sensitive data such as financial information, if you want to ensure that the client uses a secure connection to browse a site, you can redirect the request to secure connection by using https:// instead of http://.

To use the Responder feature, do the following;

- Enable the Responder feature on the NetScaler.
- Configure responder actions. The action can be to generate a custom response, redirect a request to a different Web page, or reset a connection.
- Configure responder policies. The policy determines the requests (traffic) on which an action has to be taken.
- Bind each policy to a bind point put it into effect. A bind point refers to an entity at which NetScaler examines the traffic to see if it matches a policy. For example, a bind point can be a load balancing virtual server.

You can specify a default action for requests that do not match any policy, and you can bypass the safety check for actions that would otherwise generate error messages.

The Rewrite feature of NetScaler helps in rewriting some information in the requests or responses handled by NetScaler. The following section shows some differences between the two features.

Comparison between Rewrite and Responder options

The main difference between the rewrite feature and the responder feature is as follows:

Responder cannot be used for response or server-based expressions. Responder can be used only for the following scenarios depending on client parameters:

- Redirecting a http request to new Web sites or Web pages
- Responding with some custom response
- Dropping or resetting a connection at request level

In case of a responder policy, the NetScaler examines the request from the client, takes action according to the applicable policies, sends the response to the client, and closes the connection with the client.

In case of a rewrite policy, the NetScaler examines the request from the client or response from the server, takes action according to the applicable policies, and forwards the traffic to the client or the server.

In general, it is recommended to use responder if you want the NetScaler to reset or drop a connection based on a client or request-based parameter. Use responder to redirect traffic, or respond with custom messages. Use rewrite for manipulating data on HTTP requests and responses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling the Responder Feature

Oct 29, 2013

To use the Responder feature, you must first enable it.

To enable the responder feature by using the command line interface

At the command prompt, type the following commands to enable the responder feature and verify the configuration:

- enable ns feature<feature>
- show ns feature

Example

```
enable ns feature Responder
 Done
> show ns feature

      Feature                  Acronym          Status
      -------                  -------          ------
 1)   Web Logging              WL               ON
 2)   Surge Protection         SP               ON
 .
 .
 .
 22)  Responder                RESPONDER        ON
 23)  HTML Injection           HTMLInjection    ON
 24)  NetScaler Push           push             OFF
 Done
>
```

To enable the responder feature by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Change advanced features.
3. In the Configure Advanced Features dialog box, select the Responder check box, and then click OK.
4. In the Enable/Disable Feature(s)? dialog box, click YES. A message appears in the status bar, stating that the feature has been enabled.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Responder Action

Aug 30, 2013

After enabling the responder feature, you must configure one or more actions for handling requests. The responder supports the following types of actions:

Respond with

Sends the response defined by the Target expression without forwarding the request to a web server. (The NetScaler appliance substitutes for and acts as a web server.) Use this type of action to manually define a simple HTML-based response. Normally the text for a Respond with action consists of a web server error code and brief HTML page.

Respond with SQL OK

Sends the designated SQL OK response defined by the Target expression. Use this type of action to send an SQL OK response to an SQL query.

Respond with SQL Error

Sends the designated SQL Error response defined by the Target expression. Use this type of action to send an SQL Error response to an SQL query.

Respond with HTML page

Sends the designated HTML page as the response. You can choose from a drop-down list of HTML pages that were previously uploaded, or upload a new HTML page. Use this type of action to send an imported HTML page as the response.

Redirect

Redirects the request to a different web page or web server. A Redirect action can redirect requests originally sent to a "dummy" web site that exists in DNS, but for which there is no actual web server, to an actual web site. It can also redirect search requests to an appropriate URL. Normally, the redirection target for a Redirect action consists of a complete URL.

To configure a responder action by using the command line interface

At the command prompt, type the following commands to configure a responder action and verify the configuration:

- add responder action <name> <type> <target> [-bypassSafetyCheck (YES | NO)]
- show responder action

Displays the current settings for the specified responder action. If no action name is provided, displays a list of all responder actions currently configured on the NetScaler appliance, with abbreviated settings.

Arguments

Name. Name of the responder action. Maximum Length: 127

type. Type of responder action. It can be: (respondwith).

target. An expression specifying what to respond with

htmlpage. Option specifying to respondwith htmlpage

bypassSafetyCheck. The safety check to allow unsafe expressions. NOTE: This attribute is deprecated.

hits. The number of times the action has been taken.

referenceCount. The number of references to the action.

undefHits. The number of times the action resulted in UNDEF.

comment. Any type of information about this responder action.

builtin. Flag to determine whether responder action is built-in or not

Example

To create a responder action that displays a “Not Found” error page for URLs that do not exist:

add responder action act404Error respondWith '"HTTP/1.1 404 Not Found\r\n\r\n"+ "HTTP.REQ.URL.HTTP_URL_SAFE" + "does not exist on the web server."'

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done
> show responder action

1) Name: act404Error
Operation: respondwith
Target: "HTTP/1.1 404 Not Found

"+ "HTTP.REQ.URL.HTTP_URL_SAFE" + "does not exist on the web server."
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
```

To create a responder action that displays a “Not Found” error page for URLs that do not exist:

```
add responder action act404Error respondWith '"HTTP/1.1 404 Not Found\r\n\r\n"+ "HTTP.REQ.URL.HTTP_URL_SAFE" + "does not exist on the web server."'
Done
> show responder action

1) Name: act404Error
Operation: respondwith
Target: "HTTP/1.1 404 Not Found

"+ "HTTP.REQ.URL.HTTP_URL_SAFE" + "does not exist on the web server."
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
```

To modify an existing responder action by using the command line interface

At the command prompt, type the following command to modify an existing responder action and verify the configuration:

- set responder action <name> -target <string> [-bypassSafetyCheck (YES | NO)]
- show responder action

Example

```
set responder action act404Error  -target '"HTTP/1.1 404 Not Found\r\n\r\n"+ "HTTP.REQ.URL.HTTP_URL_SAFE" + "does not exist on the web server."'
 Done
> show responder action

1)     Name: act404Error
       Operation: respondwith
       Target: "HTTP/1.1 404 Not Found

"+ "HTTP.REQ.URL.HTTP_URL_SAFE" + "does not exist on the web server."
       BypassSafetyCheck : NO
       Hits: 0
       Undef Hits: 0
       Action Reference Count: 0
 Done
```

To remove a responder action by using the command line interface

At the command prompt, type the following command to remove a responder action and verify the configuration:

- rm responder action <name>
- show responder action

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

```
rm responder action act404Error
 Done

> show responder action
 Done
```

To configure a responder action by using the configuration utility

1. Navigate to AppExpert > Responder > Actions.
2. In the details pane, do one of the following:
   - To create a new action, click Add.
   - To modify an existing action, select the action, and then click Open.
3. Click Create or OK, depending on whether you are creating a new action or modifying an existing action.
4. Click Close. A message appears in the status bar, stating that the feature has been enabled.
5. To delete a responder action, select the action, and then click Remove. A message appears in the status bar, stating that the feature has been disabled.

To add an expression by using the Add Expression dialog box

1. In the Create Responder Action or Configure Responder Action dialog box, click Add.
2. In the Add Expression dialog box, in the first list box choose the first term for your expression.

   HTTP

   The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.

   SYS

   The protected web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

   CLIENT

   The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

   ANALYTICS

   The analytics data associated with the request. Choose this if you want to examine request metadata.

   SIP

   A SIP request. Choose this if you want to examine some aspect of a SIP request.

   When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.
3. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
4. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

Configuring the Global HTTP Action

You can configure the global HTTP action to invoke a responder action when an HTTP request times out. To configure this feature, you must first create the responder action that you want to invoke. Then, you configure the global HTTP timeout action to respond to a timeout with that responder action.

To configure the global HTTP action by using the command line interface

At the command prompt, type the following command:

- set ns httpProfile -reqTimeoutAction <responder action name>
- save ns config

For <responder action name>, substitute the name of the responder action.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Responder Policy

Oct 29, 2013

After you configure a responder action, you must next configure a responder policy to select the requests to which the NetScaler appliance should respond. A responder policy is based on a rule, which consists of one or more expressions. The rule is associated with an action, which is performed if a request matches the rule.

Note: For creating and managing responder policies, the configuration utility provides assistance that is not available at the NetScaler command prompt.

To configure a responder policy by using the command line interface

At the command prompt, type the following command to add a new responder policy and verify the configuration:

- add responder policy <name> <expression> <action> [<undefaction>]-appFlowaction<actionName>
- show responder policy <name>

Example

```
> add responder policy policyThree "CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)" RESET
 Done
> show responder policy policyThree

      Name: policyThree
      Rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
      Responder Action: RESET
      UndefAction: Use Global
      Hits: 0
      Undef Hits: 0
 Done
```

To modify an existing responder policy by using the command line interface

At the command prompt, type the following command to modify an existing responder policy and verify the configuration:

- set responder policy <name> [-rule <expression>] [-action <string>] [-undefAction <string>]
- show responder policy <name>

To remove a responder policy by using the command line interface

At the command prompt, type the following command to remove a responder policy and verify the configuration:

- rm responder policy <name>
- show responder policy

Example

```
>rm responder policy pol404Error
 Done

> show responder policy
 Done
```

To configure a responder policy by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to AppExpert > Responder > Policies.
2. In the details pane, do one of the following:
   - To create a new policy, click Add.
   - To modify an existing policy, select the policy, and then click Open.
3. Click Create or OK, depending on whether you are creating a new policy or modifying an existing policy.
4. Click Close. A message appears in the status bar, stating that the feature has been configured.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding a Responder Policy

Oct 29, 2013

To put a policy into effect, you must bind it either globally, so that it applies to all traffic that flows through the NetScaler, or to a specific virtual server, so that the policy applies only to requests whose destination IP address is the VIP of that virtual server.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the NetScaler operating system, policy priorities work in reverse order—the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000. The responder feature implements only the first policy that a request matches, not any additional policies that it might also match, so policy priority is important for getting the results you intend.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you globally bind it. You can then add additional policies at any time without having to reassign the priority of an existing policy.

For additional information about binding policies on the NetScaler, see "Policies and Expressions."
Note: Responder policies cannot be bound to TCP-based virtual servers.

To globally bind a responder policy by using the command line interface

At the command prompt, type the following command to globally bind a responder policy and verify the configuration:

- bind responder global <policyName> <priority> [<gotoPriorityExpression [-type <type>] [-invoke (<labelType> <labelName>)]
- show responder global

Example

```
> bind responder global poliError 100
 Done
> show responder global
1)     Global bindpoint: REQ_DEFAULT
       Number of bound policies: 1

 Done
```

To bind responder policy to a specific virtual server by using the command line interface

At the command prompt, type the following command to bind responder policy to a specific virtual server and verify the configuration:
bind lb vserver <name> -policyname <policy_name> -priority <priority>
Example

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
1)    vs-loadbal (10.102.29.20:80) - HTTP     Type: ADDRESS
      State: OUT OF SERVICE
      Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
      Time since last state change: 2 days, 00:58:03.260
      Effective State: DOWN
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
      No. of Bound Services :  0 (Total)       0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Vserver IP and Port insertion: OFF
      Push: DISABLED  Push VServer:
      Push Multi Clients: NO
      Push Label Rule: none
2)    vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
      State: DOWN
      Last state change was at Wed Aug 19 10:03:46 2009 (+213 ms)
      Time since last state change: 2 days, 00:00:04.260
      Effective State: DOWN
      Client Idle Timeout: 9000 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      No. of Bound Services :  0 (Total)       0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Connection Failover: DISABLED
 Done
```

To globally bind a responder policy by using the configuration utility

1. Navigate to AppExpert > Responder > Policies.
2. On the Responder Policies page, select a responder policy, and then click Policy Manager.
3. In the Responder Policy Manager dialog box Bind Points menu, select Default Global.
4. Click Insert Policy to insert a new row and display a drop-down list of all unbound responder policies.
5. Click one of the policies on the list. That policy is inserted into the list of globally bound responder policies.
6. Click Apply Changes.
7. Click Close. A message appears in the status bar, stating that the configuration has been successfully completed.

To bind a responder policy to a specific virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. On the Load Balancing Virtual Servers page, select the virtual server to which you want to bind the responder policy, and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, select the Policies tab, which displays a list of all policies configured on your NetScaler appliance.
4. Select the check box next to the name of the policy you want to bind to this virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Click OK. A message appears in the status bar, stating that the configuration has been successfully completed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting the Default Action for a Responder Policy

Aug 30, 2013

The NetScaler appliance generates an undefined event (UNDEF event) when a request does not match a responder policy, and then carries out the default action assigned to undefined events. By default, that action is to forward the request to the next feature without changing it. This default behavior is normally what you want; it ensures that requests that do not require special handling by a specific responder action are sent to your Web servers and clients receive access to the content that they requested.

If the Web site(s) your NetScaler appliance protects receive a significant number of invalid or malicious requests, however, you may want to change the default action to either reset the client connection or drop the request. In this type of configuration, you would write one or more responder policies that would match any legitimate requests, and simply redirect those requests to their original destinations. Your NetScaler appliance would then block any other requests as specified by the default action you configured.

You can assign any one of the following actions to an undefined event:

NOOP

The NOOP action aborts responder processing but does not alter the packet flow. This means that the appliance continues to process requests that do not match any responder policy, and eventually forwards them to the requested URL unless another feature intervenes and blocks or redirects the request. This action is appropriate for normal requests to your Web servers and is the default setting.

RESET

If the undefined action is set to RESET, the appliance resets the client connection, informing the client that it must re-establish its session with the Web server. This action is appropriate for repeat requests for Web pages that do not exist, or for connections that might be attempts to hack or probe your protected Web site(s).

DROP

If the undefined action is set to DROP, the appliance silently drops the request without responding to the client in any way. This action is appropriate for requests that appear to be part of a DDoS attack or other sustained attack on your servers.

Note: UNDEF events are triggered only for client requests. No UNDEF events are triggered for responses.

To set the undefined action by using the command line interface

At the command prompt, type the following command to set the undefined action and verify the configuration:

- set responder param -undefAction (RESET | DROP | NOOP)
- show responder param

Example

```
>set responder param -undefAction RESET
 Done
> show responder param
      Action Name: RESET
 Done
>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To set the undefined action by using the configuration utility

1. Navigate to AppExpert > Responder, and then under Settings, click the Change Responder Settings link.
2. In the Set Responder Params dialog box, under Global Undefined-Result Action, select NOOP, RESET, or DROP.
3. Click OK. A message appears in the status bar, stating that the Responder Parameters have been configured.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Responder Action and Policy Examples

Oct 03, 2017

Responder actions and policies are powerful and complex, but you can get started with relatively simple applications. For typical examples, refer to the topics given below.

Example: Blocking Access from Specified IPs

The following procedures block access to your protected Web site(s) by clients originating from the CIDR 222.222.0.0/16. The responder sends an error message stating that the client is not authorized to access the URL requested.

To block access by using the command line interface

At the command prompt, type the following commands to block access:

- add responder action act_unauthorized respondwith '"HTTP/1.1 403 Forbidden\r\n\r\n" + "Client: " + CLIENT.IP.SRC + " is not authorized to access URL:" + "HTTP.REQ.URL.HTTP_URL_SAFE"'
- add responder policy pol_un "CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)" act_unauthorized
- bind responder global pol_un 10

To block access by using the configuration utility

1. In the navigation pane, expand Responder, and then click Actions.
2. In the details pane, click Add.
3. In the Create Responder Action dialog box, do the following:
   1. In the Name text box, type act_unauthorized.
   2. Under Type, select Respond with.
   3. In the Target text area, type the following string: "HTTP/1.1 200 OK\r\n\r\n" + "Client: " + CLIENT.IP.SRC + " is not authorized to access URL:" + HTTP.REQ.URL.HTTP_URL_SAFE
   4. Click Create, and then click Close.

   The responder action you configured, named act_unauthorized, now appears in the Responder Actions page.
4. In the navigation pane, click Policies.
5. In the details pane, click Add.
6. In the Create Responder Policy dialog box, do the following:
   1. In the Name text box, type pol_unauthorized.
   2. Under Action, select act_unauthorized.
   3. In the Expression window, type the following rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
   4. Click Create, then click Close.

   The responder policy you configured, named pol_unauthorized, now appears in the Responder Policies page.
7. Globally bind your new policy, pol_unauthorized, as described in "Binding a Responder Policy."

Example: Redirecting a Client to a new URL

The following procedures redirect clients who access your protected Web site(s) from within the CIDR 222.222.0.0/16 to a specified URL.

To redirect clients by using the command line interface

At the command prompt, type the following commands to redirect clients and verify the configuration:

- add responder action act_redirect redirect '"http://www.example.com/404.html"'
- show responder action act_redirect

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- add responder policy pol_redirect "CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)" act_redirect
- show responder policy pol_redirect
- bind responder global pol_redirect 10

Example

```
> add responder action act_redirect redirect '" http ://www.example.com/404.html "'
> add responder policy pol_redirect "CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)" act_redirect
```

To redirect clients by using the configuration utility

1. Navigate to AppExpert > Responder > Actions.
2. In the details pane, click Add.
3. In the Create Responder Action dialog box, do the following:
   1. In the Name text box, type act_redirect.
   2. Under Type, select Redirect.
   3. In the Target text area, type the following string: "http://www.example.com/404.html"
   4. Click Create, then click Close.

   The responder action you configured, named act_redirect, now appears in the Responder Actions page.
4. In the navigation pane, click Policies.
5. In the details pane, click Add.
6. In the Create Responder Policy dialog box, do the following:
   1. In the Name text box, type pol_redirect.
   2. Under Action, select act_redirect.
   3. In the Expression window, type the following rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
   4. Click Create, then click Close.

   The responder policy you configured, named pol_redirect, now appears in the Responder Policies page.
7. Globally bind your new policy, pol_redirect, as described in "Binding a Responder Policy."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Diameter Support for Responder

Apr 09, 2014

The Responder feature now supports the Diameter protocol. You can configure Responder to respond to Diameter requests as it does HTTP and TCP requests. For example, you could configure Responder to respond to requests from a specific Diameter origin with a redirect to a web page enhanced for mobile devices. A number of NetScaler expressions have been added that support examination of the Diameter header and the attribute-value pairs (AVPs). These expressions support lookup of specific AVPs by index, ID or name, examine the information in each AVP, and send an appropriate response.

To configure Responder to respond to a Diameter request

To configure the Responder feature to send a response to a diameter request, at the command prompt, type the following commands:

- add responder action <actname> RESPONDWITH "DIAMETER.NEW_REDIRECT(\"aaa://host.example.com\")"
  For <actname>, substitute a name for your new action. The name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For aaa://host.example.com, substitute the URL of the diameter host to which you want to redirect connections.
- add responder policy <polname> "diameter.req.avp(264).value.eq(\"host1.example.net\")" <actname>
  For <polname>, substitute a name for your new policy. As with <actname>, the name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For host1.example.net, substitute the name of the originating host of the requests that you want to redirect. For <actname>, substitute the name of the action that you just created.
- bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST
  For <vservername>, substitute the name of the load balancing virtual server to which you want to bind the policy. For <polname>, substitute the name of the policy you just created. For <priority>, substitute a priority for the policy.

Example

To create a Responder action and policy to respond to Diameter requests that originate from "host1.example.net" with a redirect to "host.example.com", you could add the following action and policy, and bind the policy as shown.

```
> add responder action act_resp-dm-redirect RESPONDWITH "DIAMETER.NEW_REDIRECT(\"aaa://host.example.com\")"
> add responder pol_resp-dm-redirect "diameter.req.avp(264).value.eq(\"host1.example.net\")" act_resp-dm-redirect
> bind lb vserver vs1 -policyName pol_resp-dm-redirect -priority 10 -type REQUEST

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

RADIUS Support for Responder

Oct 16, 2014

The NetScaler expressions language contains expressions that can extract information from and manipulate RADIUS requests. These expressions enable you to use the Responder feature to respond to RADIUS requests. Your responder policies and actions can use any expression that is appropriate or relevant to a RADIUS request. The available expressions enable you to identify the RADIUS message type, extract any attribute-value pair (AVP) from the connection, and send different responses on the basis of that information. You can also create policy labels that invoke all responder policies for RADIUS connections.

You can use RADIUS expressions to construct simple responses that do not require communication with the RADIUS server to which the request was sent. When a responder policy matches a connection, the NetScaler ADC constructs and sends the appropriate RADIUS response without contacting the RADIUS authentication server. For example, if the source IP address of a RADIUS request is from a subnet that is specified in the responder policy, the NetScaler ADC can reply to that request with an access-reject message, or can simply drop the request.

You can also create policy labels to route specific types of RADIUS requests through a series of policies that are appropriate to those requests.

Note: The current RADIUS expressions do not work with RADIUS IPv6 attributes.
The NetScaler documentation for expressions that support RADIUS assumes familiarity with the basic structure and purpose of RADIUS communications. If you need more information about RADIUS, see your RADIUS server documentation or search online for an introduction to the RADIUS protocol.

Configuring Responder Policies for RADIUS

The following procedure uses the NetScaler command line to configure a responder action and policy, and bind the policy to a RADIUS-specific global bind point.

To configure a Responder action and policy, and bind the policy

At the command prompt, type the following commands:

- add responder action <actName> <actType>
- add responder policy <polName> <rule> <actName>
- bind responder policy <polName> <priority> <nextExpr> -type <bindPoint> where <bindPoint> represents one of the RADIUS-specific global bind points.

RADIUS Expressions for Responder

In a responder configuration, you can use the following NetScaler expressions to refer to various portions of a RADIUS request.

Identifying the Type of Connection

RADIUS.IS_CLIENT

Returns TRUE if the connection is a RADIUS client (request) message.

RADIUS.IS_SERVER

Returns TRUE if the connection is a RADIUS server (response) message.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Expressions

RADIUS.REQ.CODE

Returns the number that corresponds to the RADIUS request type. A derivative of the num_at class. For example, a RADIUS access request would return 1 (one). A RADIUS accounting request would return 4.

RADIUS.REQ.LENGTH

Returns the length of the RADIUS request, including the header. A derivative of the num_at class.

RADIUS.REQ.IDENTIFIER

Returns the RADIUS request identifier, a number assigned to each request that allows the request to be matched to the corresponding response. A derivative of the num_at class.

RADIUS.REQ.AVP(<AVP Code No>).VALUE

Returns the value of first occurrence of this AVP as a string of type text_t.

RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)

Returns the specified instance of the AVP as a string of type RAVP_t. A specific RADIUS AVP can occur multiple times in a RADIUS message. INSTANCE (0) returns the first instance, INSTANCE (1) returns second instance, and so on, up to sixteen instances.

RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)

Returns the value of specified instance of the AVP as a string of type text_t.

RADIUS.REQ.AVP(<AVP code no>).COUNT

Returns the number of instances of a specific AVP in a RADIUS connection, as an integer.

RADIUS.REQ.AVP(<AVP code no>).EXISTS

Returns TRUE if the specified type of AVP exists in the message, or FALSE if it does not.

Response Expressions

RADIUS response expressions are identical to RADIUS request expressions, except that RES replaces REQ.

Typecasts of AVP Values

The ADC supports expressions to typecast RADIUS AVP values to the text, integer, unsigned integer, long, unsigned long, ipv4 address, ipv6 address, ipv6 prefix and time data types. The syntax is the same as for other NetScaler typecast expressions.

Example

The ADC supports expressions to typecast RADIUS AVP values to the text, integer, unsigned integer, long, unsigned long, ipv4 address, ipv6 address, ipv6 prefix and time data types. The syntax is the same as for other NetScaler typecast expressions.

RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at

AVP Type Expressions

The NetScaler ADC supports expressions to extract RADIUS AVP values by using the assigned integer codes described in RFC2865 and RFC2866. You can also use text aliases to accomplish the same task. Some examples follow.

RADIUS.REQ.AVP (1).VALUE or RADIUS.REQ.USERNAME.value

Extracts the RADIUS user-name value.

RADIUS.REQ.AVP (4). VALUE or RADIUS.REQ. ACCT_SESSION_ID.value

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Extracts the Acct-Session-ID AVP (code 44) from the message.

RADIUS.REQ.AVP (26). VALUE or RADIUS.REQ.VENDOR_SPECIFIC.VALUE

Extracts the vendor-specific value.

The values of most commonly-used RADIUS AVPs can be extracted in the same manner.

RADIUS Bind Points

Four global bind points are available for policies that contain RADIUS expressions.

RADIUS_REQ_OVERRIDE

Priority/override request policy queue.

RADIUS_REQ_DEFAULT

Standard request policy queue.

RADIUS_RES_OVERRIDE

Priority/override response policy queue.

RADIUS_RES_DEFAULT

Standard response policy queue.

RADIUS Responder-Specific Expressions

RADIUS_RESPONDWITH

Respond with the specified RADIUS response. The response is created with NetScaler expressions, both RADIUS expressions and any others that are applicable.

RADIUS.NEW_ANSWER

Sends a new RADIUS answer to the user.

RADIUS.NEW_ACCESSREJECT

Rejects the RADIUS request.

RADIUS.NEW_AVP

Adds the specified new AVP to the response.

Use Cases

Following are use cases for RADIUS with responder.

Blocking RADIUS Requests from a Specific Network

To configure the responder feature to block authentication requests from a specific network, begin by creating a responder action that rejects requests. Use the action in a policy that selects requests from the networks that you want to block. Bind the responder policy to a RADIUS-specific global bind point, specifying:

- The priority
- END as the nextExpr value, to ensure that policy evaluation stops when this policy is matched
- RADIUS_REQ_OVERRIDE as the queue to which you assign the policy, so that it is evaluated before policies assigned to the default queue

To configure Responder to block logons from a specific network

- add responder action <actName> <actType>
- add responder policy <polName> <rule> <actName>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- bind responder global <polName> <priority> <nextExpr> -type <bindPoint>

Example

add responder action rspActRadiusReject respondwith radius.new_accessreject
add responder policy rspPolRadiusReject client.ip.src.in_subnet(10.224.85.0/24) rspActRadiusReject
bind responder global rspPolRadiusReject 1 END -type RADIUS_REQ_OVERRIDE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

DNS Support for the Responder Feature

Jun 17, 2015

You can configure the responder feature to respond to DNS requests as it does to HTTP and TCP requests. For example, you could configure it to send DNS responses over UDP and ensure that the DNS requests from the client are sent over TCP. A number of NetScaler expressions support examination of the DNS header in the request. These expressions examine specific header fields and send an appropriate response.

DNS Expressions

In a responder configuration, you can use the following NetScaler expressions to refer to various portions of a DNS request:

Expressions	Descriptions
DNS.NEW_RESPONSE	Creates a new empty DNS response based on the request.
DNS.NEW_RESPONSE <AA, TC, rcode>	Creates a new DNS response based on the specified parameters.

DNS Bind Points

The following global bind points are available for policies that contain DNS expressions.

Bind Points	Descriptions
DNS_REQ_OVERRIDE	Priority/override request policy queue.
DNS_REQ_DEFAULT	Standard request policy queue.

In addition to the default bind points, you can create policy labels of type DNS and bind DNS policies to them.

Configuring Responder Policies for DNS

The following procedure uses the NetScaler command line to configure a responder action and policy and bind the policy to a responder-specific global bind point.

To configure Responder to respond to a DNS request

At the command prompt, type the following commands:

1. add responder action <actName> <actType>
   For <actname>, substitute a name for your new action. The name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For <actType>, substitute a responder action type, *respondWith*.

2. add responder policy <polName> <rule> <actName>
   For <polname>, substitute a name for your new policy. For <actname>, the name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For <actname>, substitute the name of the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

action that you just created.

3. bind responder policy <polName> <priority> <nextExpr> -type <bindPoint>
   For <bindPoint>, specify one of the responder-specific global bind points. For <polName>, substitute the name of the policy that you just created. For <priority>, specify the priority of the policy.

Example

Enforce all DNS request over TCP

To enforce all the DNS requests over TCP, create a responder action that will set the TC bit and rcode as NOERROR.

```
add responder action resp_act_set_tc_bit respondwith DNS.NEW_RESPONSE(true, true, NOERROR)
add responder policy enforce_tcp  dns.REQ.TRANSPORT.EQ(udp)  resp_act_set_tc_bit
bind lb vserver dns_udp –policyName  enforce_tcp   -type request –priority 100
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Jul 22, 2013

If the responder feature does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for Troubleshooting

Updated: 2013-07-22

For best results, use the following resources to troubleshoot an integrated cache issue on a NetScaler appliance:

- The ns.conf file
- The relevant trace files from the client and the NetScaler appliance

In addition to the above resources, the following tools expedite troubleshooting:

- The iehttpheaders or a similar utility
- The Wireshark application customized for the NetScaler trace files

Troubleshooting Responder Issues

Updated: 2013-07-29

- **Issue**
  The Responder feature is configured, but the responder action is not working.

  Resolution
  - Verify that the feature is enabled.
  - Check the hit counters of any of the policies to see if the counters are getting incremented.
  - Verify that the policies and actions are configured correctly.
  - Verify that the actions and policies are bound appropriately.
  - Record the packet traces on the client and the NetScaler appliance, and analyze the them to get some pointer to the issue.
  - Record the iehttpHeaters packet traces on the client and verify the HTTP requests and responses to get some pointer to the issue.
- **Issue**
  You need to create a maintenance page.

  Resolution
  1. Configure the services and virtual Server.
  2. Configure a backup virtual server with a service bound to it. This ensures that the status of the Web site is always displayed as UP.
  3. Configure the primary virtual server to use the backup virtual server as a backup.
  4. Create a responder action with an appropriate target. Following is an example for your reference:
     add responder action sorry_page respondwith q{"HTTP/1.0 200 OK" +"\r\n\r\n" + "<html><body>Sorry, this page is not available</body></html>" + "\r\n"} .
  5. Create a responder policy and bind the action to it.
  6. Bind the responder policy to the backup virtual Server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rewrite

Aug 30, 2013

Rewrite refers to the rewriting of some information in the requests or responses handled by the NetScaler appliance. Rewriting can help in providing access to the requested content without exposing unnecessary details about the Web site's actual configuration. A few situations in which the rewrite feature is useful are described below:

- To improve security, the NetScaler can rewrite all the http:// links to https:// in the response body.
- In the SSL offload deployment, the insecure links in the response have to be converted into secure links. Using the rewrite option, you can rewrite all the http:// links to https:// for making sure that the outgoing responses from NetScaler to the client have the secured links.
- If a Web site has to show an error page, you can show a custom error page instead of the default 404 Error page. For example, if you show the home page or site map of the Web site instead of an error page, the visitor remains on the site instead of moving away from the Web site.
- If you want to launch a new Web site, but use the old URL, you can use the Rewrite option.
- When a topic in a site has a complicated URL, you can rewrite it with a simple, easy-to-remember URL (also referred to as 'cool URL').
- You can append the default page name to the URL of a Web site. For example, if the default page of a company's Web site is 'http://www.abc.com/index.php', when the user types 'abc.com' in the address bar of the browser, you can rewrite the URL to 'abc.com/index.php'.

When you enable the rewrite feature, NetScaler can modify the headers and body of HTTP requests and responses.

To rewrite HTTP requests and responses, you can use protocol-aware NetScaler policy expressions in the rewrite policies you configure. The virtual servers that manage the HTTP requests and responses must be of type HTTP or SSL. In HTTP traffic, you can take the following actions:

- Modify the URL of a request
- Add, modify or delete headers
- Add, replace, or delete any specific string within the body or headers.

To rewrite TCP payloads, consider the payload as a raw stream of bytes. Each of the virtual servers that managing the TCP connections must be of type TCP or SSL_TCP. The term TCP rewrite is used to refer to the rewrite of TCP payloads that are not HTTP data. In TCP traffic, you can add, modify, or delete any part of the TCP payload.

For examples to use the rewrite feature, see Rewrite Action and Policy Examples.

Comparison between Rewrite and Responder options

The main difference between the rewrite feature and the responder feature is as follows:

Responder cannot be used for response or server-based expressions. Responder can be used only for the following scenarios depending on client parameters:

- Redirecting a http request to new Web sites or Web pages
- Responding with some custom response
- Dropping or resetting a connection at request level

In case of a responder policy, the NetScaler examines the request from the client, takes action according to the applicable policies, sends the response to the client, and closes the connection with the client.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

In case of a rewrite policy, the NetScaler examines the request from the client or response from the server, takes action according to the applicable policies, and forwards the traffic to the client or the server.

In general, it is recommended to use responder if you want the NetScaler to reset or drop a connection based on a client or request-based parameter. Use responder to redirect traffic, or respond with custom messages. Use rewrite for manipulating data on HTTP requests and responses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How Rewrite Works

Aug 30, 2013

A rewrite policy consists of a rule and action. The rule determines the traffic on which rewrite is applied and the action determines the action to be taken by the NetScaler. You can define multiple rewrite policies. For each policy, specify the bind point and priority.

A bind point refers to a point in the traffic flow at which the NetScaler examines the traffic to verify whether any rewrite policy can be applied to it. You can bind a policy to a specific load balancing or content switching virtual server, or make the policy global if you want the policy to be applied to the entire traffic handled by the NetScaler. These policies are referred to as global policies.

In addition to the user-defined policies, the NetScaler has some default policies. You cannot modify or delete a default policy.

For evaluating the policies, NetScaler follows the order mentioned below:

- Global policies
- Policies bound to specific virtual servers
- Default policies

Note: NetScaler can apply a rewrite policy only when it is bound to a point.
NetScaler implements the rewrite feature in the following steps:

- The NetScaler appliance checks for global policies and then checks for policies at individual bind points.
- If multiple policies are bound to a bind point, the NetScaler evaluates the policies in the order of their priority. The policy with the highest priority is evaluated first. After evaluating each policy, if the policy is evaluated to TRUE (the traffic matches the rule), it adds the action associated with the policy to a list of actions to be performed. A match occurs when the characteristics specified in the policy rule match the characteristics of the request or response being evaluated.
- For any policy, in addition to the action, you can specify the policy that should be evaluated after the current policy is evaluated. This policy is referred to as the 'Go to Expression'. For any policy, if a Go to Expression (gotoPriorityExpr) is specified, the NetScaler evaluates the Go to Expression policy; it ignores policy with the next highest priority.
  You can specify the priority of the policy to indicate the Go to Expression policy; you cannot use the name of the policy.
  If you want the NetScaler to stop evaluating other policies after evaluating a particular policy, you can set the Go to Expression to 'END'.

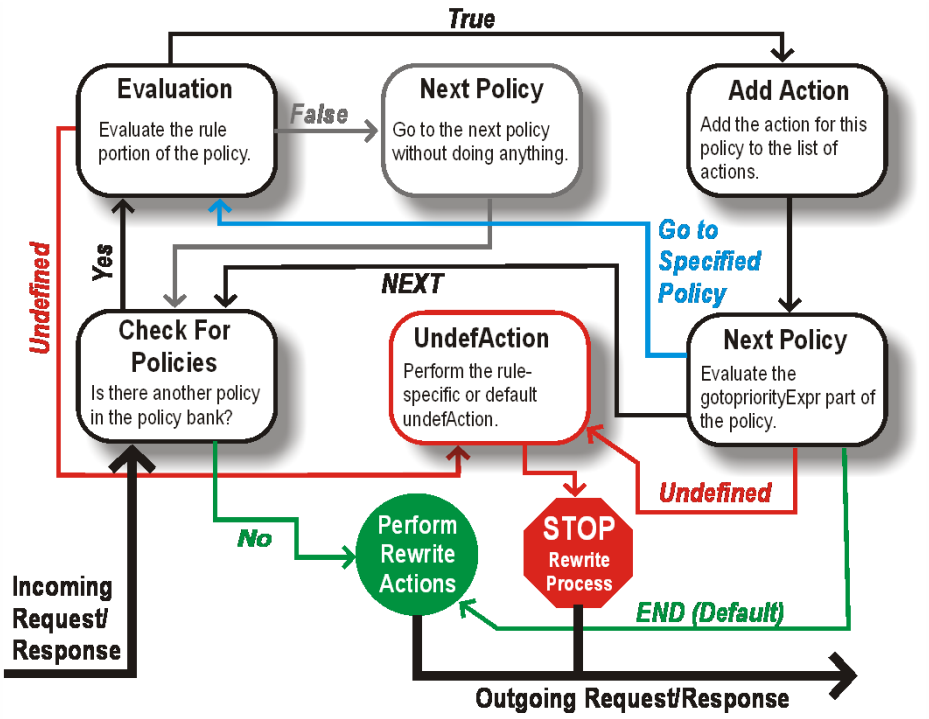
- After all the policies are evaluated or when a policy has the Go to Expression set as END, the NetScaler starts performing the actions according to the list of actions.

For more information about configuring rewrite policies, see "Configuring a Rewrite Policy" and about binding rewrite policies, see "Binding a Rewrite Policy."

The following figure illustrates how NetScaler processes a request or response when the rewrite feature is used.

Figure 1. The Rewrite Process

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Policy Evaluation

The policy with the highest priority is evaluated first. NetScaler does not stop the evaluation of rewrite policies when it finds a match; it evaluates all the rewrite policies configured on the NetScaler.

- If a policy evaluates to TRUE, the NetScaler follows the procedure below:
  - If the policy has the Go to Expression set to END, the NetScaler stops evaluating all the other policies and starts performing the rewrite.
  - The gotoPriorityExpression can be set to 'NEXT', 'END', some integer or 'INVOCATION_LIST'. The value determines the policy with the next priority. The following table shows the action taken by NetScaler for each value of the expression.

Value of the expression	Action
NEXT	Policy with the next priority gets evaluated.
END	Evaluation of policies stops.
<an integer>	Policy with specified priority gets evaluated.
INVOCATION_LIST	Goto NEXT or END is applied based on the result of the invocation list.

- If a policy evaluates to FALSE, the NetScaler continues the evaluation in the order of priority.
- If a policy evaluates to UNDEFINED (cannot be evaluated on the received traffic due to an error), the NetScaler performs the action assigned to the UNDEFINED condition (referred to as undefAction) and stops further evaluation of polices.

The NetScaler starts the actual rewriting only after the evaluation is complete. It refers to the list of actions identified by

© 1999-2017 Citrix Systems, Inc. All rights reserved.

policies that are evaluated to TRUE, and starts the rewriting. After implementing all the actions in the list, the NetScaler forwards the traffic as required.

Note: Ensure that the policies do not specify conflicting or overlapping actions on the same part of the HTTP header or body, or TCP payload. When such a conflict occurs, the NetScaler encounters an undefined situation and aborts the rewrite.

Rewrite Actions

On the NetScaler appliance, specify the actions to be taken such as adding, replacing, or deleting text within the body, or adding, modifying or deleting headers, or any changes in the TCP payload as rewrite actions. For more information about rewrite actions, see "Configuring a Rewrite Action."

The following table describes the steps the NetScaler can take when a policy evaluates to TRUE.

Action	Result
Insert	The rewrite action specified for the policy is carried out.
NOREWRITE	The request or response is not rewritten. NetScaler forwards the traffic without rewriting any part of the message.
RESET	The connection is aborted at the TCP level.
DROP	The message is dropped.

Note: For any policy, you can configure the undefaction (action to be taken when the policy evaluates to UNDEFINED) as NOREWRITE, RESET, or DROP.

To use the Rewrite feature, take the following steps:

- Enable the feature on the NetScaler.
- Define rewrite actions.
- Define rewrite policies.
- Bind the policies to a bind point to bring a policy into effect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling the Rewrite Feature

Aug 30, 2013

Enable the rewrite feature on the NetScaler appliance if you want to rewrite the HTTP or TCP requests or responses. If the feature is enabled, NetScaler takes rewrite action according to the specified policies. For more information, see "How Rewrite Works."

To enable the rewrite feature by using the command line interface

At the command prompt, type the following commands to enable the rewrite feature and verify the configuration:

- enable ns feature REWRITE
- show ns feature

Example

```
> enable ns feature REWRITE
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging            WL               OFF
 2)   Surge Protection       SP               ON
 .
 .
 .
 19)  Rewrite                REWRITE          ON
 .
 .
 24)  NetScaler Push         push             OFF
 Done
```

To enable the rewrite feature by using the configuration utility

1. In the navigation pane, click System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure basic features.
3. In the Configure Basic Features dialog box, select the Rewrite check box, and then click OK.
4. In the Enable/Disable Feature(s) dialog box, click Yes. A message appears in the status bar, stating that the selected feature was enabled.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Rewrite Action

Nov 24, 2014

After enabling the rewrite feature, you need to configure one or more actions unless a built-in rewrite action is sufficient. All of the built-in actions have names beginning with the string ns_cvpn, followed by a string of letters and underscore characters. Built-in actions perform useful and complex tasks such as decoding parts of a clientless VPN request or response or modifying JavaScript or XML data. The built-in actions can be viewed, enabled, and disabled, but cannot be modified or deleted.

Target expressions in actions for TCP rewrite must begin with one of the following expression prefixes:

- **CLIENT.TCP.PAYLOAD.** For rewriting TCP payloads in client requests. For example, CLIENT.TCP.PAYLOAD(10000).AFTER_STR("string1").
- **SERVER.TCP.PAYLOAD.** For rewriting TCP payloads in server responses. For example, SERVER.TCP.PAYLOAD(1000).B64DECODE.BETWEEN("string1","string2").

You can use all types of existing string manipulation functions with these prefixes to identify the strings that you want to rewrite. To configure a rewrite action, you assign it a name, specify an action type, and add one or more arguments specifying additional data. The following table describes the action types and the arguments you use with them.
Note: Action types that can be used only for HTTP rewrite are identified in the **Rewrite Action Type** column.

Table 1. Rewrite Action Types and Their Arguments

Rewrite Action Type	Argument 1	Argument 2
INSERT_HTTP_HEADER: Inserts the HTTP header you specify into the HTTP request or response. This is the default choice. This action type can be used only with HTTP requests and responses.	The HTTP header you want to insert. For example, if you want to insert the client IP from which a request is sent, type Client-IP.	A string expression that describes the contents of the header you want to insert. For example, if you want to insert the Client IP from which a request is sent, type CLIENT.IP.SRC.
INSERT_BEFORE: Inserts a new string before the designated string.	A string expression that describes the string before which you want to insert a new string. For example, if you want to find the hostname www.example.com and insert a string before the example.com portion, type the following: HTTP.REQ.HOSTNAME.BEFORE_STR ("example.com")	A string expression that describes the new string you want to insert. For example, if you want to insert the new string en. before the string example in the hostname, type en followed by a period.
INSERT_AFTER: Inserts a new string after the designated string.	A string expression that describes the string after which you want to insert a	A string expression that describes the new string

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rewrite Action Type	Argument 1	Argument 2
	new string. For example, if you want to find the hostname www.example.com, and insert a string after the www. portion, type the following: HTTP.REQ.HOSTNAME.AFTER_STR ("www.")	you want to insert. For example, if you want to insert the new string en. after the string www. in the hostname, type en followed by a period.
REPLACE: Replaces the designated string with a different string.	A string expression that describes the string you want to replace with a new string. For example, if you want to replace the entire hostname in the Host header, type HTTP.REQ.HOSTNAME.SERVER.	A string expression that describes the new string you want to insert. For example, if you want to replace the current host header with the string web01.example.net, type web01.example.net.
DELETE: Deletes the designated string.	A string expression that describes the string you want to delete. For example, if you want to find and delete the string .en in the hostname of HTTP response headers, type the following: HTTP.RES.HEADER("Host").SUBSTR("en.")	
DELETE_HTTP_HEADER: Deletes the designated HTTP header, including all header contents. This action type can be used only with HTTP requests and responses.	The name of the HTTP header you want to delete. For example, if you want to delete the cache-control header from HTTP responses, type HTTP.RES.HEADER ("Cache-Control").	
CORRUPT_HTTP_HEADER: Replaces the name of the given HTTP header with a corrupted name so that it will not be recognized by the receiver. This action type can be used only with HTTP requests and responses.	The name of the HTTP header that you want to corrupt. If the specified header occurs more than once in a request, all the occurrences are corrupted. For example, if you want to corrupt the Host header in an HTTP request, you can use the following rewrite action command: add rewrite action corrupt_header_act	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rewrite Action Type	Argument 1	Argument 2
	CORRUPT_HTTP_HEADER Host.	
REPLACE_HTTP_RES: Replace the http response with the value specified in the target field. This action type can be used only with HTTP requests and responses.	A string expression that describes the string you want to replace the HTTP response with. For example, type HTTP 200 OK You are not authorized to view this page to replace the entire HTTP response with this warning.	
REPLACE_ALL: Will replace all occurrences of a pattern in the target text reference with the value specified in the string builder expression.	The part of either the HTTP request or response where you want to carry out the replacement.	A string expression that describes the new string you want to insert.
DELETE_ALL: Delete every occurrence of the pattern specified in the target text reference.	The part of either the HTTP request or response where you want the deletion to occur.	A string pattern after which the deletion should occur.
INSERT_AFTER_ALL: Inserts the value specified by string builder expression after each occurrence of a specified pattern in the target text reference.	The part of either the HTTP request or response where you want the insertion to occur.	A string expression that describes the new string you want to insert.
INSERT_BEFORE_ALL: Inserts the value you specify before each occurrence of the pattern you specify.	The part of either the HTTP request or response that you want to delete.	A string expression that describes the new string you want to insert.
CLIENTLESS_VPN_ENCODE: Encodes the URL you specify in clientless VPN format.	The URL you want to encode.	
CLIENTLESS_VPN_ENCODE_ALL: Encodes all of the URLs you specify in clientless VPN format.	A pattern that matches the URLs you want to encode.	
CLIENTLESS_VPN_DECODE: Decodes the URL you specify from clientless VPN format and returns it as unencoded text.	The URL you want to decode.	
CLIENTLESS_VPN_DECODE_ALL: Decodes all of the URLs you specify from clientless VPN format and returns them as unencoded text.	A pattern that matches all of the URLs you want to decode.	

To create a new rewrite action by using the command line interface

At the command prompt, type the following commands to create a new rewrite action and verify the configuration:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- add rewrite action <name> <type> <target> [<stringBuilderExpr>] [(-pattern <expression> | -patset <string>)] [-bypassSafetyCheck (YES|NO)]
- show rewrite action <name>

Example 1: Inserting an HTTP Header With the Client IP

```
> add rewrite action insertact INSERT_HTTP_HEADER "client-IP" CLIENT.IP.SRC
Done

> show rewrite action insertact

      Name: insertact
      Operation: insert_http_header   Target:Client-IP
      Value:CLIENT.IP.SRC
      BypassSafetyCheck : NO
      Hits: 0
      Undef Hits: 0
      Action Reference Count: 0
 Done
```

Example 2: Replacing Strings in a TCP Payload (TCP Rewrite)

```
> add rewrite action client_tcp_payload_replace_all REPLACE_ALL
  'client.tcp.payload(1000)' '"new-string"' -search text("old-string")
 Done
> show rewrite action client_tcp_payload_replace_all

      Name: client_tcp_payload_replace_all
      Operation: replace_all
      Target:client.tcp.payload(1000)
      Value:"new-string"
      Search: text("old-string")
      BypassSafetyCheck : NO
      Hits: 0
      Undef Hits: 0
      Action Reference Count: 0
 Done
>
```

To modify an existing rewrite action by using the command line interface

At the command prompt, type the following commands to modify an existing rewrite action and verify the configuration:

- set rewrite action <name> [-target <string>] [-stringBuilderExpr <string>] [(-pattern <expression> | -patset <string>)] [-bypassSafetyCheck (YES|NO)]
- show rewrite action <name>

Example

```
> set rewrite action insertact -target "Client-IP"
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> show rewrite action insertact

    Name: insertact
    Operation: insert_http_header   Target:Client-IP
    Value:CLIENT.IP.SRC
    BypassSafetyCheck : NO
    Hits: 0
    Undef Hits: 0
    Action Reference Count: 0
 Done
```

To remove a rewrite action by using the command line interface

At the command prompt, type the following commands to remove a rewrite action :
rm rewrite action <name>

Example

```
> rm rewrite action insertact
Done
```

To configure a rewrite action by using the configuration utility

1. Navigate to AppExpert > Rewrite > Actions.
2. In the details pane, do one of the following:
   - To create a new action, click Add.
   - To modify an existing action, select the action, and then click Open.
3. Click Create or OK. A message appears in the status bar, stating that the Action has been configured successfully.
4. Repeat steps 2 through 4 to create or modify as many rewrite actions as you wish.
5. Click Close.

To add an expression by using the Add Expression dialog box

1. In the Create Rewrite Action or Configure Rewrite Action dialog box, under the text area for the type argument you want to enter, click Add.
2. In the Add Expression dialog box, in the first list box choose the first term for your expression.

   HTTP

   The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.

   SYS

   The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

   CLIENT

   The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

   When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.
3. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
4. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

For more information about the PI expressions language and creating expressions for responder policies, see "Policies and Expressions."

If you want to test the effect of a rewrite action when used on sample HTTP data, you can use the Rewrite Expression Evaluator.
Note: The Rewrite Expression Evaluator is only available in the configuration utility. There is no NetScaler command line version.

To evaluate a rewrite action by using the Rewrite Action Evaluator dialog box

1. In the Rewrite Actions details pane, select the rewrite action that you want to evaluate, and then click Evaluate.
2. In the Rewrite Expression Evaluator dialog box, specify values for the following parameters. (An asterisk indicates a required parameter.)
   - Rewrite Action*—If the rewrite action you want to evaluate is not already selected, select it from the drop-down list. After you select a Rewrite action, the Details section displays the details of the selected Rewrite action.
   - New*—Select New to open the Create Rewrite Action dialog box and create a new rewrite action.
   - Modify*—Select Modify to open the Configure Rewrite Action dialog box and modify the selected rewrite action.
   - Flow Type*—Specifies whether to test the selected rewrite action with HTTP Request data or HTTP Response data. The default is Request. If you want to test with Response data, select Response.
   - HTTP Request/Response Data*—Provides a space for you to provide the HTTP data that the Rewrite Action Evaluator will use for testing. You can paste the data directly into the window, or click Sample to insert some sample HTTP headers.
   - Show end-of-line—Specifies whether to show UNIX-style end-of-line characters (\n) at the end of each line of sample HTTP data.
   - Sample—Inserts sample HTTP data into the HTTP Request/Response Data window. You can choose either GET or POST data.
   - Browse—Opens a local browse window so that you can choose a file containing sample HTTP data from a local or network location.
   - Clear—Clears the current sample HTTP data from the HTTP Request/Response Data window.
3. Click Evaluate. The Rewrite Action Evaluator evaluates the effect of the Rewrite action on the sample data that you chose, and displays the results as modified by the selected Rewrite action in the Results window. Additions and deletions are highlighted as indicated in the legend in the lower left-hand corner of the dialog box.
4. Continue evaluating Rewrite actions until you have determined that all of your actions have the effect that you wanted.
   - You can modify the selected rewrite action and test the modified version by clicking Modify to open the Configure Rewrite Action dialog box, making and saving your changes, and then clicking Evaluate again.
   - You can evaluate a different rewrite action using the same request or response data by selecting it from the Rewrite Action drop-down list, and then clicking Evaluate again.
5. Click Close to close the Rewrite Expression Evaluator and return to the Rewrite Actions pane.

To delete a rewrite action, select the rewrite action you want to delete, then click Remove and, when prompted, confirm your choice by clicking OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Rewrite Policy

Aug 30, 2013

After you create any needed rewrite action(s), you must create at least one rewrite policy to select the requests that you want the NetScaler appliance to rewrite.

A rewrite policy consists of a rule, which itself consists of one or more expressions, and an associated action that is performed if a request or response matches the rule. Policy rules for evaluating HTTP requests and responses can be based on almost any part of a request or response.

Even though you cannot use TCP rewrite actions to rewrite data other than the TCP payload, you can base the policy rules for TCP rewrite policies on the information in the transport layer and the layers below the transport layer.

If a configured rule matches a request or response, the corresponding policy is triggered and the action associated with it is carried out.

Note: You can use either the command line interface or the configuration utility to create and configure rewrite policies. Users who are not thoroughly familiar with the command line interface and the NetScaler Policy expression language will usually find using the configuration utility much easier.

To add a new rewrite policy by using the command line interface

At the command prompt, type the following commands to add a new rewrite policy and verify the configuration:

- add rewrite policy <name> <expression> <action> [<undefaction>]
- show rewrite policy <name>

Example 1: Rewriting HTTP Content

```
> add rewrite policy policyNew "HTTP.RES.IS_VALID" insertact NOREWRITE
 Done
> show rewrite policy policyNew
      Name: policyNew
      Rule: HTTP.RES.IS_VALID
      RewriteAction: insertact
      UndefAction: NOREWRITE
      Hits: 0
      Undef Hits: 0

 Done
```

Example 2: Rewriting a TCP Payload (TCP Rewrite)

```
> add rewrite policy client_tcp_payload_policy CLIENT.IP.SRC.EQ(172.168.12.232) client_tcp_payload_replace_all
 Done
> show rewrite policy client_tcp_payload_policy
      Name: client_tcp_payload_policy
      Rule: CLIENT.IP.SRC.EQ(172.168.12.232)
      RewriteAction: client_tcp_payload_replace_all
      UndefAction: Use Global
      LogAction: Use Global
      Hits: 0
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Undef Hits: 0

 Done
>
```

To modify an existing rewrite policy by using the command line interface

At the command prompt, type the following commands to modify an existing rewrite policy and verify the configuration:

- set rewrite policy <name> -rule <expression> -action <action> [<undefaction>]
- show rewrite policy <name>

Example

```
> set rewrite policy policyNew -rule "HTTP.RES.IS_VALID" -action insertaction
 Done

> show rewrite policy policyNew
    Name: policyNew
    Rule: HTTP.RES.IS_VALID
    RewriteAction: insertaction
    UndefAction: NOREWRITE
    Hits: 0
    Undef Hits: 0

 Done
```

To remove a rewrite policy by using the command line interface

At the command prompt, type the following command to remove a rewrite policy:
rm rewrite policy <name>

Example

```
> rm rewrite policy policyNew
Done
```

To configure a rewrite policy by using the configuration utility

1. Navigate to AppExpert > Rewrite > Policies.
2. In the details pane, do one of the following:
   - To create a new policy, click Add.
   - To modify an existing policy, select the policy, and then click Open.
3. Click Create or OK. A message appears in the status bar, stating that the Policy has been configured successfully.
4. Repeat steps 2 through 4 to create or modify as many rewrite actions as you wish.
5. Click Close. To delete a rewrite policy, select the rewrite policy you want to delete, then click Remove and, when prompted, confirm your choice by clicking OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding a Rewrite Policy

Oct 29, 2013

After creating a rewrite policy, you must bind it to put it into effect. You can bind your policy to Global if you want to apply it to all traffic that passes through your NetScaler, or you can bind your policy to a specific virtual server or bind point to direct only that virtual server or bind point's incoming traffic to that policy. If an incoming request matches a rewrite policy, the action associated with that policy is carried out.

Rewrite policies for evaluating HTTP requests and responses can be bound to virtual servers of type HTTP or SSL, or they can be bound to the REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, and RES_DEFAULT bind points. Rewrite policies for TCP rewrite can be bound only to virtual servers of type TCP or SSL_TCP, or to the OTHERTCP_REQ_OVERRIDE, OTHERTCP_REQ_DEFAULT, OTHERTCP_RES_OVERRIDE, and OTHERTCP_RES_DEFAULT bind points.
Note: The term OTHERTCP is used in the context of the NetScaler appliance to refer to all TCP or SSL_TCP requests and responses that you want to treat as a raw stream of bytes regardless of the protocols that the TCP packets encapsulate.
When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the NetScaler operating system, policy priorities work in reverse order - the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is applied first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000.

Unlike most other features in the NetScaler operating system, the rewrite feature continues to evaluate and implement policies after a request matches a policy. However, the effect of a particular action policy on a request or response will often be different depending on whether it is performed before or after another action. Priority is important to get the results you intended.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you bind it. If you do this, you can add additional policies at any time without having to reassign the priority of an existing policy.

When binding a rewrite policy, you also have the option of assigning a goto expression (gotoPriorityExpression) to the policy. A goto expression can be any positive integer that matches the priority assigned to a different policy that has a higher priority than the policy that contains the goto expression. If you assign a goto expression to a policy, and a request or response matches the policy, the NetScaler will immediately go to the policy whose priority matches the goto expression. It will skip over any policies with priority numbers that are lower than that of the current policy, but higher than the priority number of the goto expression, and not evaluate those policies.

For more information about binding policies on the NetScaler, see "Binding a Rewrite Policy."

To globally bind a rewrite policy by using the command line interface

At the command prompt, type the following commands to globally bind a rewrite policy and verify the configuration:

- bind rewrite global <policyName> <priority> [<gotoPriorityExpression>] [-type <type>] [-invoke (<labelType> <labelName>)]
- show rewrite global

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>bind rewrite global policyNew 10
 Done

> show rewrite global
1)     Global bindpoint: RES_DEFAULT
       Number of bound policies: 1

2)     Global bindpoint: REQ_OVERRIDE
       Number of bound policies: 1

 Done
```

To bind rewrite policy to a specific virtual server by using the command line interface

At the command prompt, type the following commands to bind rewrite policy to a specific virtual server and verify the configuration:

- bind lb vserver <name>@ (<serviceName>@ [-weight <positive_integer>]) | <serviceGroupName>@ | (-policyName <string>@ [-priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type (REQUEST | RESPONSE)] [-invoke (<labelType> <labelName>)])
- show lb vserver <name>

Example

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 50
 Done
>
> show lb vserver lbvip
       lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
       State: DOWN
       Last state change was at Wed Jul 15 05:54:24 2009 (+226 ms)
       Time since last state change: 28 days, 01:57:26.350
       Effective State: DOWN
       Client Idle Timeout: 180 sec
       Down state flush: ENABLED
       Disable Primary Vserver On Down : DISABLED
       Port Rewrite : DISABLED
       No. of Bound Services :  0 (Total)       0 (Active)
       Configured Method: LEASTCONNECTION
       Mode: IP
       Persistence: NONE
       Vserver IP and Port insertion: OFF
       Push: DISABLED  Push VServer:
       Push Multi Clients: NO
       Push Label Rule: none

1)     Policy : ns_cmp_msapp Priority:50
2)     Policy : cf-pol Priority:1      Inherited
 Done
```

To bind a rewrite policy to a bind point by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to AppExpert > Rewrite > Policies.
2. In the details pane, select the rewrite policy you want to globally bind, and then click Policy Manager.
3. In the Rewrite Policy Manager dialog box, in the Bind Points menu, do one of the following:
   1. If you want to configure bindings for HTTP rewrite policies, click HTTP, and then click either Request or Response, depending on whether you want to configure request-based rewrite policies or response-based rewrite policies.
   2. If you want to configure bindings for TCP rewrite policies, click TCP, and then click either Client or Server, depending on whether you want to configure client-side TCP rewrite policies or server-side TCP rewrite policies.
4. Click the bind point to which you want to bind the rewrite policy. The Rewrite Policy Manager dialog box displays all the rewrite policies that are bound to the selected bind point.
5. Click Insert Policy to insert a new row and display a drop-down list with all available, unbound rewrite policies.
6. Click the policy you want to bind to the bind point. The policy is inserted into the list of rewrite policies bound to the bind point.
7. In the Priority column, you can change the priority to any positive integer. For more information about this parameter, see priority in "Parameters for binding a rewrite policy."
8. If you want to skip over policies and go directly to a specific policy in the event that the current policy is matched, change the value in the Goto Expression column to equal the priority of the next policy to be applied.. For more information about this parameter, see gotoPriorityExpression in "Parameters for binding a rewrite policy."
9. To modify a policy, click the policy, and then click Modify Policy.
10. To unbind a policy, click the policy, and then click Unbind Policy.
11. To modify an action, in the Action column, click the action you want to modify, and then click Modify Action.
12. To modify an invoke label, in the Invoke column, click the invoke label you want to modify, and then click Modify Invoke Label.
13. To regenerate the priorities of all the policies that are bound to the bind point you are currently configuring, click Regenerate Priorities. The policies retain their existing priorities relative to the other policies, but the priorities are renumbered in multiples of ten.
14. Click Apply Changes.
15. Click Close. A message appears in the status bar, stating that the Policy has been configured successfully.

To bind a rewrite policy to a specific virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane list of virtual servers, select the virtual server to which you want to bind the rewrite policy, and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, select the Policies tab. All policies configured on your NetScaler appear on the list.
4. Select the check box next to the name of the policy you want to bind to this virtual server.
5. Click OK. A message appears in the status bar, stating that the Policy has been configured successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Rewrite Policy Labels

Aug 30, 2013

If you want to build a more complex policy structure than is supported by single policies, you can create policy labels and then bind them as you would policies. A policy label is a user-defined point to which policies are bound. When a policy label is invoked, all the policies bound to it are evaluated in the order of the priority you configured. A policy label can include one or multiple policies, each of which can be assigned its own result. A match on one policy in the policy label can result in proceeding to the next policy, invoking a different policy label or appropriate resource, or an immediate end to policy evaluation and return of control to the policy that invoked the policy label.

A rewrite policy label consists of a name, a transform name that describes the type of policy included in the policy label, and a list of policies bound to the policy label. Each policy that is bound to the policy label contains all of the elements described in "Configuring a Rewrite Policy."

Note: You can use either the command line interface or the configuration utility to create and configure rewrite policy labels. Users who are not thoroughly familiar with the command line interface and the NetScaler Policy Infrastructure (PI) language will usually find using the configuration utility much easier.

To configure a rewrite policy label by using the command line interface

To add a new rewrite policy label, at the command prompt, type the following command:

```
add rewrite policylabel <labelName> <transform>
```

For example, to add a rewrite policy label named polLabelHTTPResponses to group all policies that work on HTTP responses, you would type the following:

```
add rewrite policylabel polLabelHTTPResponses http_res
```

To modify an existing rewrite policy label, at the NetScaler command prompt, type the following command:

```
set rewrite policy <name> <transform>
```

Note: The set rewrite policy command takes the same options as the add rewrite policy command.
To remove a rewrite policy label, at the NetScaler command prompt, type the following command:

```
rm rewrite policy<name>
```

For example, to remove a rewrite policy label named polLabelHTTPResponses, you would type the following:

```
rm rewrite policy polLabelHTTPResponses
```

To configure a rewrite policy label by using the configuration utility

1. Navigate to AppExpert > Rewrite > Policy Labels.
2. In the details pane, do one of the following:
   - To create a new policy label, click Add.
   - To modify an existing policy label, select the policy, and then click Open.
3. Add or remove policies from the list that is bound to the policy label.
   - To add a policy to the list, click Insert Policy, and choose a policy from the drop-down list. You can create a new policy and add it to the list by choosing New Policy in the list, and following the instructions in "Configuring a Rewrite

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy]."

- To remove a policy from the list, select that policy, and then click Unbind Policy.

4. Modify the priority of each policy by editing the number in the Priority column.

   You can also automatically renumber policies by clicking Regenerate Priorities.

5. Click Create or OK, and then click Close.

   To remove a policy label, select it, and then click Remove. To rename a policy label, select it and then click Rename. Edit the name of the policy, and then click OK to save your changes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Default Rewrite Action

Aug 30, 2013

An undefined event is triggered when the NetScaler cannot evaluate a policy, usually because it detects a logical or other error in the policy or an error condition on the NetScaler. When the rewrite policy evaluation results in an error, the specified undefined action is carried out. Undefined actions configured at the rewrite policy level are carried out before a globally configured undefined action.

The NetScaler supports following three types of undefined actions:

undefAction NOREWRITE

Aborts rewrite processing, but does not alter the packet flow. This means that the NetScaler continues to process requests and responses that do not match any rewrite policy, and eventually forwards them to the requested URL unless another feature intervenes and blocks or redirects the request. This action is appropriate for normal requests to your Web servers, and is the default setting.

undefAction RESET

Resets the client connection. This means that the NetScaler tells the client that it must re-establish its session with the Web server. This action is appropriate for repeat requests for Web pages that do not exist, or for connections that might be attempts to hack or probe your protected Web site(s).

undefAction DROP

Silently drops the request without responding to the client in any way. This means that the NetScaler simply discards the connection without responding to the client. This action is appropriate for requests that appear to be part of a DDoS attack or another sustained attack on your servers.

Note: Undefined events can be triggered for both request and response flow specific policies.

To configure the default action by using the command line interface

At the command prompt, type the following commands to configure the default action and verify the configuration:

- set rewrite param -undefAction (NOREWRITE | RESET | DROP)
- show rewrite param

Example

```
> set rewrite param -undefAction NOREWRITE
 Done
> show rewrite param
      Action Name: NOREWRITE
 Done
```

To configure the default action by using the configuration utility

1. Navigate to AppExpert > Rewrite.
2. In the details pane, under Rewrite Overview, click the Change Rewrite Settings link. The Set Rewrite Params dialog box appears.
3. Under Global Undefined-Result Action, select an option as follows:
   - NoRewrite—NOREWRITE
   - Reset—RESET
   - Drop—DROP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Click OK. The global undefined action is set to the value you chose.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Bypassing the Safety Check

Sep 29, 2016

When you create a rewrite action, the NetScaler verifies that the expression you used to create the action is safe. Expressions created by the NetScaler from run-time data, such as URLs contained in HTTP requests, can cause unexpected errors. The NetScaler reports expressions that cause such errors as unsafe expressions.

In some cases, the expressions may be safe. For example, the NetScaler cannot validate an expression that contains a URL that does not resolve, even if the URL does not resolve because the Web server is temporarily unavailable. You can manually bypass the Safety Check to allow these expressions.

To bypass the safety check by using the command line interface

At the command prompt, type the following commands to bypass the safety check and verify the configuration:

- set rewrite action <name> -bypassSafetyCheck YES
- show rewrite action <name>

Example

```
> set rewrite action insertact -bypassSafetyCheck YES
 Done
> show rewrite action insertact

      Name: insertact
      Operation: insert_http_header   Target:Client-IP
      Value:CLIENT.IP.SRC
      BypassSafetyCheck : YES
      Hits: 0
      Undef Hits: 0
      Action Reference Count: 2
 Done
```

To bypass safety check by using the configuration utility

1. Navigate to **AppExpert** > **Rewrite** > **Actions**.
2. In the details pane, click **Add** to add a rewrite expression that needs to be exempted from the safety check.
3. In the **Create Rewrite Action** dialog box, set the following parameters:
   1. Action Type
   2. Header Name
   3. Policy Expression
   4. Comments
4. Click **Create**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rewrite Action and Policy Examples

Jul 12, 2017

The examples in this section demonstrate how to configure rewrite to perform various useful tasks. The examples occur in the server room of Example Manufacturing Inc., a mid-sized manufacturing company that uses its Web site to manage a considerable portion of its sales, deliveries, and customer support.

Example Manufacturing has two domains: example.com for its Web site and email to customers, and example.net for its intranet. Customers use the Example Web site to place orders, request quotes, research products, and contact customer service and technical support.

As an important part of Example's revenue stream, the Web site must respond quickly and keep customer data confidential. Example therefore has several Web servers and uses Citrix NetScaler appliances to balance the Web site load and manage traffic to and from its Web servers.

The Example system administrators use the rewrite features to perform the following tasks:

Example 1: Delete old X-Forwarded-For and Client-IP Headers.

Example Inc. removes old X-Forwarded-For and Client-IP HTTP headers from incoming requests.

Example 2: Adding a Local Client-IP Header.

Example Inc. adds a new, local Client-IP header to incoming requests.

Example 3: Tagging Secure and Insecure Connections.

Example Inc. tags incoming requests with a header that indicates whether the connection is a secure connection.

Example 4: Mask the HTTP Server Type.

Example Inc. modifies the HTTP Server: header so that unauthorized users and malicious code cannot use that header to determine the HTTP server software it uses.

Example 5: Redirect an External URL to an Internal URL.

Example Inc. hides information about the actual names of its Web servers and the configuration of its server room from users, to make URLs on its Web site shorter and easier to remember, and to improve security on its site.

Example 6: Migrating Apache Rewrite Module Rules.

Example Inc. moved its Apache rewrite rules to a NetScaler appliance, translating the Apache PERL-based script syntax to the NetScaler rewrite rule syntax.

Example 7: Marketing Keyword Redirection.

The marketing department at Example Inc. sets up simplified URLs for certain predefined keyword searches on the company's Web site.

Example 8: Redirect Queries to the Queried Server.

Example Inc. redirects certain query requests to the appropriate server.

Example 9: Home Page Redirection.

Example Inc. recently acquired a smaller competitor, and it now redirects requests for the acquired company's home page to a page on its own Web site.

Each of these tasks requires that the system administrators create rewrite actions and policies and bind them to a valid bind point on the NetScaler.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 1: Delete Old X-Forwarded-For and Client-IP Headers

Aug 30, 2013

Example Inc. wants to remove old X-Forwarded-For and Client-IP HTTP headers from incoming requests, so that the only X-Forwarded-For headers that appear are the ones added by the local server. This configuration can be done through the NetScaler command line or the configuration utility. The Example Inc. system administrator is an old-school networking engineer and prefers to use a CLI where possible, but wants to be sure he understands the configuration utility interface so that he can show new system administrators on the team how to use it.

The examples below demonstrate how to perform each configuration with both the CLI and the configuration utility. The procedures are abbreviated on the assumption that users will already know the basics of creating rewrite actions, creating rewrite policies, and binding policies.

- For more detailed information about creating rewrite actions, see Configuring a Rewrite Action.
- For more detailed information about creating rewrite policies, see Configuring a Rewrite Policy.
- For more detailed information about binding rewrite policies, see Binding a Rewrite Policy.

To delete old X-Forwarded and Client-IP headers from a request by using the command line interface

At the command prompt, type the following commands in the order shown:

```
add rewrite action act_del_xfor delete_http_header x-forwarded-for
add rewrite action act_del_cip delete_http_header client-ip
add rewrite policy pol_check_xfor 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS' act_del_xfor
add rewrite policy pol_check_cip 'HTTP.REQ.HEADER("client-ip").EXISTS' act_del_cip
bind rewrite global pol_check_xfor 100 200
bind rewrite global pol_check_cip 200 300
```

To delete old X-Forwarded and Client-IP headers from a request by using the configuration utility

In the Create Rewrite Action dialog box, create two rewrite actions with the following descriptions.

Name	Type	Argument(s)
act_del_xfor	delete_http_header	x-forwarded-for
act_del_cip	delete_http_header	client-ip

In the Create Rewrite Policy dialog box, create two rewrite policies with the following descriptions.

Name	Expression	Action
pol_check_xfor	'HTTP.REQ.HEADER("x-forwarded-for").EXISTS'	act_del_xfor
pol_check_cip	'HTTP.REQ.HEADER("client-ip").EXISTS'	act_del_cip

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Bind both policies to global, assigning the priorities and goto expression values shown below.

Name	Priority	Goto Expression
pol_check_xfor	100	200
pol_check_cip	200	300

All old X-Forwarded-For and Client-IP HTTP headers are now deleted from incoming requests.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 2: Adding a Local Client-IP Header

Aug 30, 2013

Example Inc. wants to add a local Client-IP HTTP header to incoming requests. This example contains two slightly different versions of the same basic task.

To add a local Client-IP header by using the command line interface

At the command prompt, type the following commands in the order shown:

```
add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.IP.SRC'
add rewrite policy pol_ins_client 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' act_ins_client
bind rewrite global pol_ins_client 300 END
```

To add a local Client-IP header by using the configuration utility

In the Create Rewrite Action dialog box, create a rewrite action with the following description.

Name	Type	Argument(s)
act_ins_client	insert_http_header	NS-Client 'CLIENT.IP.SRC'

In the Create Rewrite Policy dialog box, create a rewrite policy with the following description.

Name	Expression	Action
pol_ins_client	'HTTP.REQ.HEADER("x-forwarded-for").EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS'	act_ins_client

Bind the policy to global, assigning the priorities and goto expression values shown below.

Name	Priority	GoTo
pol_ins_client	100	Next

A local Client-IP HTTP header is now added to incoming requests. You can also modify the configuration above to append all IPs from X-Forwarded-For headers to the new Client-IP header, as shown below.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 3: Tagging Secure and Insecure Connections

Mar 20, 2012

Example Inc. wants to tag incoming requests with a header that indicates whether or not the connection is a secure connection. This helps the server keep track of secure connections after the NetScaler has decrypted the connections.

To implement this configuration, you would begin by creating rewrite actions with the values shown in the following tables. These actions label connections to port 80 as insecure connections, and connections to port 443 as secure connections.

Action Name	Type of Rewrite Action	Header Name	Value
Action-Rewrite-SSL_YES	INSERT_HTTP_HEADER	SSL	YES

Action Name	Type of Rewrite Action	Header Name	Value
Action-Rewrite-SSL_NO	INSERT_HTTP_HEADER	SSL	NO

You would then create a rewrite policy with the values shown in the following tables. These policies check incoming requests to determine which requests are directed to port 80 and which are directed to port 443. The policies then add the correct SSL header.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-SSL_YES	Action-Rewrite-SSL_YES	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(443)
Policy-Rewrite-SSL_NO	Action-Rewrite-SSL_NO	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(80)

Finally, you would bind the rewrite policies to NetScaler, assigning the first policy a priority of 200, and the second a priority of 300, and setting the goto expression of both policies to END.

Each incoming connection to port 80 now has an SSL:NO HTTP header added to it and each incoming connection to port 443 has an SSL:YES HTTP header added to it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 4: Mask the HTTP Server Type

Mar 20, 2012

Example Inc. wants to modify the HTTP Server: header so that unauthorized users and malicious code cannot use the header to identify the software that the HTTP server uses.

To modify the HTTP Server: header, you would create a rewrite action and a rewrite policy with the values in the following tables.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Server_Mask	REPLACE	HTTP.RES.HEADER("Server")	"Web Server 1.0"

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Server_Mask	Action-Rewrite-Server_Mask	NOREWRITE	HTTP.RES.IS_VALID

You would then globally bind the rewrite policy, assigning a priority of 100 and setting the Goto Priority Expression of the policy to END.

The HTTP Server: header is now modified to read “Web Server 1.0,” masking the actual HTTP server software used by the Example Inc. Web site.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 5: Redirect an External URL to an Internal URL

Mar 20, 2012

Example Inc. wants to hide its actual server room configuration from users to improve security on its Web servers.

To do this, you would create a rewrite action with the values as shown in the following tables. For request headers, the action in the table modifies www.example.com to web.hq.example.net. For response headers, the action does the opposite, translating web.hq.example.net to www.example.com.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Request_Server_Replace	REPLACE	HTTP.REQ.HOSTNAME.SERVER	"Web.hq.example.net"
Action-Rewrite-Response_Server_Replace	REPLACE	HTTP.RES.HEADER("Server")	"www.example.com"

Next, you would create rewrite policies using the values shown in the following tables. The first policy checks incoming requests to see if they are valid, and if they are, it performs the Action-Rewrite-Request_Server_Replace action. The second policy checks responses to see if they originate at the server web.hq.example.net. If they do, it performs the Action-Rewrite-Response_Server_Replace action.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Request_Server_Replace	Action-Rewrite-Request_Server_Replace	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")
Policy-Rewrite-Response_Server_Replace	Action-Rewrite-Response_Server_Replace	NOREWRITE	HTTP.RES.HEADER("Server").EQ("web.hq.example.net")

Finally, you would bind the rewrite policies, assigning each a priority of 500 because they are in different policy banks and therefore will not conflict. You should set the goto expression to NEXT for both bindings.

All instances of www.example.com in the request headers are now changed to web.hq.example.net, and all instances of web.hq.example.net in response headers are now changed to www.example.com.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 6: Migrating Apache Rewrite Module Rules

Mar 20, 2012

Example Inc., is currently using the Apache rewrite module to process search requests sent to its Web servers and redirect those requests to the appropriate server on the basis of information in the request URL. Example Inc. wants to simplify its setup by migrating these rules onto the NetScaler platform.

Several Apache rewrite rules that Example currently uses are shown below. These rules redirect search requests to a special results page if they do not have a SiteID string or if they have a SiteID string equal to zero (0), or to the standard results page if these conditions do not apply.

The following are the current Apache rewrite rules:

- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} !SiteId= [OR]
- RewriteCond %{QUERY_STRING} SiteId=0
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ /results2.html [P,L]
- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ /results.html [P,L]

To implement these Apache rewrite rules on the NetScaler, you would create rewrite actions with the values in the following tables.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Display_Results_NulSiteID	REPLACE	HTTP.REQ.URL	"/results2.html"
Action-Rewrite-Display_Results	REPLACE	HTTP.REQ.URL	"/results2.html"

You would then create rewrite policies with the values as shown in the tables below.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Display_Results_NulSiteID	Action-Rewrite-Display_Results_NulSiteID	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MODE(IGNORECASE).EQ("/search") && (!HTTP.REQ.URL.QUERY.CONTAINS("SiteId=") || HTTP.REQ.URL.QUERY.CONTAINS("SiteId=0") || HTTP.REQ.URL.QUERY.SET_TEXT_MODE(IGNORECASE).CONTAINS("CallName=DisplayResults"))
Policy-Rewrite-Display_Results	Action-Rewrite-Display_Results	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MODE(IGNORECASE).EQ("/search") || HTTP.REQ.URL.QUERY.SET_TEXT_MODE(IGNORECASE).CONTAINS("CallName=DisplayResults"))

Finally, you would bind the rewrite policies, assigning the first a priority of 600 and the second a priority of 700, and then set the goto expression to NEXT for both bindings.

The NetScaler now handles these search requests exactly as the Web server did before the Apache rewrite module rules were migrated.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 7: Marketing Keyword Redirection

Mar 20, 2012

The marketing department at Example Inc. wants to set up simplified URLs for certain predefined keyword searches on the company's Web site. For these keywords, it wants to redefine the URL as shown below.

- External URL: http://www.example.com/<marketingkeyword>
- Internal URL: http://www.example.com/go/kwsearch.asp?keyword=<marketingkeyword>

To set up redirection for marketing keywords, you would create a rewrite action with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target location	String expression for replacement text
Action-Rewrite-Modify_URL	INSERT_BEFORE	HTTP.REQ.URL.PATH.GET(1)	""go/kwsearch.aspkeyword="l"

You would then create a rewrite policy with the values in the following table.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Modify_URL	Action-Rewrite-Modify_URL	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")

Finally, you would bind the rewrite policy, assigning it a priority of 800. Unlike the previous rewrite policies, this policy should be the last to be applied to a request that matches its criteria. For this reason, NetScaler administrator sets its Goto Priority Expression to END.

Any request using a marketing keyword is redirected to the keyword search CGI page, whereupon a search is performed and all remaining policies are skipped.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 8: Redirect Queries to the Queried Server

Mar 20, 2012

Example Inc. wants to redirect query requests to the appropriate server, as shown here.

- Request: GET /query.cgi?server=5HOST: www.example.com
- Redirect URL: http://web-5.example.com/

To implement this redirection, you would first create a rewrite action with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Replace_Hostheader	REPLACE	HTTP.REQ.HEADER("Host").BEFORE_STR(".example.com")	"server-" + HTTP.REQ.URL.QUERY.VALUE("web")

You would then create a rewrite policy with the values in the following table.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Replace_Hostheader	Action-Rewrite-Replace_Hostheader	NOREWRITE	HTTP.REQ.HEADER("Host").EQ("www.example.com")

Finally, you would bind the rewrite policy, assigning it a priority of 900. Because this policy should be the last policy applied to a request that matches its criteria, you set the goto expression to END.

Incoming requests to any URL that begins with http://www.example.com/query.cgi?server= are redirected to the server number in the query.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 9: Home Page Redirection

Mar 20, 2012

New Company, Inc. recently acquired a smaller competitor, Purchased Company, and wants to redirect the home page for Purchased Company to a new page on its own Web site, as shown here.

- Old URL: http://www.purchasedcompany.com/*
- New URL: http://www.newcompany.com/products/page.htm

To redirect requests to the Purchased Company home page, you would create rewrite actions with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Replace_URLr	REPLACE	HTTP.REQ.URL.PATH_AND_QUERY	"/products/page.htm"
Action-Rewrite-Replace_Host	REPLACE	HTTP.REQ.HOSTNAME	"www.newcompany.com"

You would then create rewrite policies with the values in the following table.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Replace-None	Action-Rewrite-Replace-None	NOREWRITE	!HTTP.REQ.HOSTNAME.SERVER.EQ("www.purchasedcompany.com")
Policy-Rewrite-Replace-Host	Action-Rewrite-Replace_Host	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ("www.purchasedcompany.com")
Policy-Rewrite-Replace-URL	Action-Rewrite-Replace_URL	NOREWRITE	HTTP.REQ.IS_VALID

Finally, you would bind the rewrite policies globally, assigning the first a priority of 100, the second a priority of 200, and the third a priority of 300. These policies should be the last policies applied to a request that matches the criteria. For this reason, set the goto expression to END for the first and third policies, and to 300 for the second policy. This ensures that all remaining requests are processed correctly.

Requests to the acquired company's old Web site are now redirected to the correct page on the New Company home page.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL Transformation

Mar 09, 2018

The URL transformation feature provides a method for modifying all URLs in designated requests from an external version seen by outside users to an internal URL seen only by your Web servers and IT staff. You can redirect user requests seamlessly, without exposing your network structure to users. You can also modify complex internal URLs that users may find difficult to remember into simpler, more easily remembered external URLs.

> ## Note
>
> Before you can use the URL transformation feature, you must enable the Rewrite feature. To enable the Rewrite feature, see [Enabling the Rewrite Feature].
>
> URL transform rewrites URLs in HTML response body and is not applied to JavaScript and other variables.

To begin configuring URL transformation, you create profiles, each describing a specific transformation. Within each profile, you create one or more actions that describe the transformation in detail. Next, you create policies, each of which identifies a type of HTTP request to transform, and you associate each policy with an appropriate profile. Finally, you globally bind each policy to put it into effect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring URL Transformation Profiles

Oct 29, 2013

A profile describes a specific URL transformation as a series of actions. The profile functions primarily as a container for the actions, determining the order in which the actions are performed. Most transformations transform an external hostname and optional path into a different, internal hostname and path. Most useful transformations are simple and require only a single action, but you can use multiple actions to perform complex transformations.

You cannot create actions and then add them to a profile. You must create the profile first, and then add actions to it. In the CLI, creating an action and configuring the action are separate steps. Creating a profile and configuring the profile are separate steps in both the CLI and the configuration utility.

To create a URL transformation profile by using the NetScaler command line

At the NetScaler command prompt, type the following commands, in the order shown, to create a URL transformation profile and verify the configuration. You can then repeat the second and third commands to configure additional actions:

- add transform profile <profileName> -type URL [-onlyTransformAbsURLinBody (ON | OFF)] [-comment <comment>]
- add transform action <name> <profileName> <priority>
- set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainFrom <expression>] [-cookieDomainInto <expression>] [-state (ENABLED | DISABLED)] [-comment "<string>"]
- show transform profile <name>

Example

```
> add transform profile shoppingcart -type URL
 Done
> add transform action actshopping shoppingcart 1000
 Done
> set transform action actshopping -priority 1000 -reqUrlFrom 'shopping.example.com' -reqUrlInto 'www.example.net/shopping' -resUrlFrom 'www.example.net/shopping' -resUrlInto 'shopping.example.cor
 Done
> show transform profile shoppingcart
      Name: shoppingcart
           Type: URL      onlyTransformAbsURLinBody: OFF
      Comment:
      Actions:

1)           Priority 1000   Name: actshopping      ENABLED
 Done
```

To modify an existing URL transformation profile or action by using the NetScaler command line

At the NetScaler command prompt, type the following commands to modify an existing URL transformation profile or action and verify the configuration:

Note: Use a set transform profile or set transform action command, respectively. The set transform profile command takes the same arguments as does the add transform profile command, and set transform action is the same command that was used for initial configuration.

- set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainInto <expression>] [-state (ENABLED | DISABLED)] [-comment "<string>"]
- show transform profile <name>

Example

```
> set transform action actshopping -priority 1000 -reqUrlFrom 'searching.example.net' -reqUrlInto 'www.example.net/searching' -resUrlFrom 'www.example.net/searching' -resUrlInto 'searching.example.cc
 Done
> show transform profile shoppingcart
      Name: shoppingcart
           Type: URL      onlyTransformAbsURLinBody: OFF
      Comment:
      Actions:

1)           Priority 1000   Name: actshopping      ENABLED
 Done
```

To remove a URL transformation profile and actions by using the NetScaler command line

First remove all actions associated with that profile by typing the following command once for each action:

- rm transform action <name> After you have removed all actions associated with a profile, remove the profile as shown below.
- rm transform profile <name>

To create a URL transformation profile by using the configuration utility

1. In the navigation pane, expand Rewrite, expand URL Transformation, and then click Profiles.
2. In the details pane, click Add.
3. In the Create URL Transformation Profile dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation profiles" as follows (asterisk indicates a required parameter):
   - Name*—name
   - Comment—comment
   - Only transform absolute URLs in response body—onlyTransformAbsURLinBody
4. Click Create, and then click Close. A message appears in the status bar, stating that the Profile has been configured successfully.

To configure a URL transformation profile and actions by using the configuration utility

1. In the navigation pane, expand Rewrite, expand URL Transformation, and then click Profiles.
2. In the details pane, select the profile you want to configure, and then click Open.
3. In the Configure URL Transformation Profile dialog box, do one of the following.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- To create a new action, click Add.
- To modify an existing action, select the action, and then click Open.

4. Fill in the Create URL Transformation Action or Modify URL Transformation Action dialog box by typing or selecting values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation profiles" as follows (asterisk indicates a required parameter):
   - Action Name*—name
   - Comments—comment
   - Priority*—priority
   - Request URL from—reqUrlFrom
   - Request URL into—reqUrlInto
   - Response URL from—resUrlFrom
   - Response URL into—resUrlInto
   - Cookie Domain from—cookieDomainFrom
   - Cookie Domain into—cookieDomainInto
   - Enabled—state
5. Save your changes.
   - If you are creating a new action, click Create, and then Close.
   - If you are modifying an existing action, click OK.

   A message appears in the status bar, stating that the Profile has been configured successfully.
6. Repeat step 3 through step 5 to create or modify any additional actions.
7. To delete an action, select the action, and then click Remove. When prompted, click OK to confirm the deletion.
8. Click OK to save your changes and close the Modify URL Transformation Profile dialog box.
9. To delete a profile, in the details pane select the profile, and then click Remove. When prompted, click OK to confirm the deletion.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring URL Transformation Policies

Oct 29, 2013

After you create a URL transformation profile, you next create a URL transformation policy to select the requests and responses that the NetScaler should transform by using the profile. URL transformation considers each request and the response to it as a single unit, so URL transformation policies are evaluated only when a request is received. If a policy matches, the NetScaler transforms both the request and the response.

Note: The URL transformation and rewrite features cannot both operate on the same HTTP header during request processing. Because of this, if you want to apply a URL transformation to a request, you must make sure that none of the HTTP headers it will modify are manipulated by any rewrite action.

To configure a URL transformation policy by using the NetScaler command line

You must create a new policy. On the command line, an existing policy can only be removed. At the NetScaler command prompt, type the following commands to configure a URL transformation policy and verify the configuration:

- add transform policy <name> <rule> <profileName>
- show transform policy <name>

Example

```
> add transform policy polsearch  HTTP.REQ.URL.SUFFIX.EQ("Searching") prosearching
 Done
> show transform policy polsearch
1)     Name: polsearch
       Rule: HTTP.REQ.URL.SUFFIX.EQ("Searching")
       Profile: prosearching
       Priority: 0
       Hits: 0
 Done
```

To remove a URL transformation policy by using the NetScaler command line

At the NetScaler command prompt, type the following command to remove a URL transformation policy:

rm transform policy <name>

Example

```
> rm transform policy polsearch
Done
```

To configure a URL transformation policy by using the configuration utility

1. In the navigation pane, expand Rewrite, expand URL Transformation, and then click Policies.
2. In the details pane, do one of the following:
   - To create a new policy, click Add.
   - To modify an existing policy, select the policy, and then click Open.
3. In the Create URL Transformation Policy or Configure URL Transformation Policy dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation policies" as follows (asterisk indicates a required parameter):
   - Name*—name (Cannot be changed for a previously configured policy.)
   - Profile*—profileName
   - Expression—rule

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

If you want help with creating an expression for a new policy, you can either hold down the Control key and press the space bar while your cursor is in the Expression text box. To create the expression, you can type it directly as described below, or you can use the Add Expression dialog box as described in To add an expression by using the Add Expression dialog box.

1. Click Prefix, and choose the prefix for your expression.
   Your choices are:

   - HTTP—The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - SYS—The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - CLIENT—The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - SERVER—The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.
   - URL—The URL of the request. Choose this if you want to examine some aspect of the URL to which the request was sent.
   - TEXT—Any text string in the request. Choose this if you want to examine a text string in the request.
   - TARGET—The target of the request. Choose this if you want to examine some aspect of the request target.
     After you choose a prefix, the NetScaler displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom. The choices depend on which prefix you chose.

2. Select your next term.
   If you chose HTTP as your prefix, your choices are REQ, which specifies HTTP requests, and RES, which specifies HTTP responses. If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

   When you are certain which choice you want, double-click it to insert it into the Expression window.

3. Type a period, and then continue selecting terms from the list boxes that appear to the right of the previous list box. You type the appropriate text strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

4. Click Create or OK, depending on whether you are creating a new policy or modifying an existing policy.
5. Click Close. A message appears in the status bar, stating that the Policy has been configured successfully.

To add an expression by using the Add Expression dialog box

1. In the Create Responder Action or Configure Responder Action dialog box, click Add.
2. In the Add Expression dialog box, in the first list box choose the first term for your expression.
   HTTP

   The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   SYS

   The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   CLIENT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

SERVER

The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

URL

The URL of the request. Choose this if you want to examine some aspect of the URL to which the request was sent.

TEXT

Any text string in the request. Choose this if you want to examine a text string in the request.

TARGET

The target of the request. Choose this if you want to examine some aspect of the request target.

When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.

3. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
4. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Globally Binding URL Transformation Policies

Oct 29, 2013

After you have configured your URL transformation policies, you bind them to Global or a bind point to put them into effect. After binding, any a request or response that matches a URL transformation policy is transformed by the profile associated with that policy.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler OS, policy priorities work in reverse order - the higher the number, the lower the priority.

Because the URL transformation feature implements only the first policy that a request matches, not any additional policies that it might also match, policy priority is important for achieving the results that you intend. If you give your first policy a low priority (such as 1000), you tell the NetScaler to perform it only if other policies with a higher priority do not match a request. If you give your first policy a high priority (such as 1), you tell the NetScaler to perform it first, and skip any other policies that might also match. You can leave yourself plenty of room to add other policies in any order, without having to reassign priorities, by setting priorities with intervals of 50 or 100 between each policy when you globally bind your policies.

Note: URL transformation policies cannot be bound to TCP-based virtual servers.

To bind a URL transformation policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to globally bind a URL transformation policy and verify the configuration:

- bind transform global <policyName> <priority>
- show transform global

Example

```
> bind transform global polisearching 100
 Done
> show transform global
1)     Policy Name: polisearching
       Priority: 100

 Done
```

To bind a URL transformation policy by using the configuration utility

1. In the navigation pane, expand Rewrite, then expand URL Transformation, and then click Policies.
2. In the details pane, click Policy Manager.
3. In the Transform Policy Manager dialog box, choose the bind point to which you want to bind the policy. The choices are:
   - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance, and are applied before any other policies.
   - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
- **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance.
- **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.

4. Select Insert Policy to insert a new row and display a drop-down list with all available, unbound URL transformation policies.
5. Select the policy you want to bind, or select New Policy to create a new policy. The policy that you selected or created is inserted into the list of globally bound URL transformation policies.
6. Make any additional adjustments to the binding.
   - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
   - To modify the policy expression, double click that field to open the Configure Transform Policy dialog box, where you can edit the policy expression.
   - To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
   - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression
7. Repeat steps 3 through 6 to add any additional URL transformation policies you want to globally bind.
8. Click OK to save your changes. A message appears in the status bar, stating that the Policy has been configured successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

RADIUS Support for the Rewrite Feature

Oct 27, 2015

The NetScaler expressions language includes expressions that can extract information from and manipulate RADIUS messages in requests and responses. These expressions enable you to use the rewrite feature to modify portions of a RADIUS message before sending it to its destination. Your rewrite policies and actions can use any expression that is appropriate or relevant to a RADIUS message. The available expressions enable you to identify the RADIUS message type, extract any attribute-value pair (AVP) from the connection, and modify RADIUS AVPs. You can also create policy labels for RADIUS connections.

You can use the new RADIUS expressions in Rewrite rules for a number of purposes. For example, you could:

- Remove the domain\ portion of the RADIUS user-name AVP to simplify single sign-on (SSO).
- Insert a vendor-specific AVP, such as the MSISDN field used in telephone company operations to contain subscriber information.

You can also create policy labels to route specific types of RADIUS requests through a series of policies that are appropriate to those requests.

Note: RADIUS for Rewrite has the following limitations:
- The NetScaler ADC does not re-sign rewritten RADIUS requests or responses. If the RADIUS authentication server requires signed RADIUS messages, authentication will fail.
- The currently available RADIUS expressions do not work with RADIUS IPv6 attributes.

The NetScaler documentation for expressions that support RADIUS assumes familiarity with the basic structure and purpose of RADIUS communications. If you need more information about RADIUS, see your RADIUS server documentation or search online for an introduction to the RADIUS protocol.

Configuring Rewrite Policies for RADIUS

The following procedure uses the NetScaler command line to configure a rewrite action and policy and bind the policy to a rewrite-specific global bind point.

To configure a Rewrite action and policy, and bind the policy

At the command prompt, type the following commands:

- add rewrite action <actName> <actType>
- add rewrite policy <polName> <rule> <actName>
- bind rewrite policy <polName> <priority> <nextExpr> -type <bindPoint>
  where <bindPoint> represents one of the rewrite-specific global bind points.

RADIUS Expressions for Rewrite

In a rewrite configuration, you can use the following NetScaler expressions to refer to various portions of a RADIUS request or response.

Identifying the Type of Connection

RADIUS.IS_CLIENT

Returns TRUE if the connection is a RADIUS client (request) message.

RADIUS.IS_SERVER

Returns TRUE if the connection is a RADIUS server (response) message.

Request Expressions

RADIUS.REQ.CODE

Returns the number that corresponds to the RADIUS request type. A derivative of the num_at class. For example, a RADIUS access request would return 1 (one). A RADIUS accounting request would return 4.

RADIUS.REQ.LENGTH

Returns the length of the RADIUS request, including the header. A derivative of the num_at class.

RADIUS.REQ.IDENTIFIER

Returns the RADIUS request identifier, a number assigned to each request that allows the request to be matched to the corresponding response. A derivative of the num_at class.

RADIUS.REQ.AVP(<AVP Code No>).VALUE

Returns the value of first occurrence of this AVP as a string of type text_t.

RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)

Returns the specified instance of the AVP as a string of type RAVP_t. A specific RADIUS AVP can occur multiple times in a RADIUS message. INSTANCE (0) returns the first instance, INSTANCE (1) returns second instance, and so on, up to sixteen instances.

RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)

Returns the value of specified instance of the AVP as a string of type text_t.

RADIUS.REQ.AVP(<AVP code no>).COUNT

Returns the number of instances of a specific AVP in a RADIUS connection, as an integer.

RADIUS.REQ.AVP(<AVP code no>).EXISTS

Returns TRUE if the specified type of AVP exists in the message, or FALSE if it does not.

Response Expressions

RADIUS response expressions are identical to RADIUS request expressions, except that RES replaces REQ.

Typecasts of AVP Values

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The ADC supports expressions to typecast RADIUS AVP values to the text, integer, unsigned integer, long, unsigned long, ipv4 address, ipv6 address, ipv6 prefix and time data types. The syntax is the same as for other NetScaler typecast expressions.

Example

The ADC supports expressions to typecast RADIUS AVP values to the text, integer, unsigned integer, long, unsigned long, ipv4 address, ipv6 address, ipv6 prefix and time data types. The syntax is the same as for other NetScaler typecast expressions.

RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at

AVP Type Expressions

The NetScaler ADC supports expressions to extract RADIUS AVP values by using the assigned integer codes described in RFC2865 and RFC2866. You can also use text aliases to accomplish the same task. Some examples follow.

RADIUS.REQ.AVP (1).VALUE or RADIUS.REQ.USERNAME.value

Extracts the RADIUS user-name value.

RADIUS.REQ.AVP (4). VALUE or RADIUS.REQ. ACCT_SESSION_ID.value

Extracts the Acct-Session-ID AVP (code 44) from the message.

RADIUS.REQ.AVP (26). VALUE or RADIUS.REQ.VENDOR_SPECIFIC.VALUE

Extracts the vendor-specific value.

The values of most commonly-used RADIUS AVPs can be extracted in the same manner.

RADIUS Bind Points

Four global bind points are available for policies that contain RADIUS expressions.

RADIUS_REQ_OVERRIDE

Priority/override request policy queue.

RADIUS_REQ_DEFAULT

Standard request policy queue.

RADIUS_RES_OVERRIDE

Priority/override response policy queue.

RADIUS_RES_DEFAULT

Standard response policy queue.

RADIUS Rewrite-Specific Expressions

RADIUS.NEW_AVP

Returns the specified RADIUS AVP as a string.

RADIUS.NEW_AVP_INTEGER32

Returns the specified RADIUS AVP as an integer.

RADIUS.NEW_AVP_UNSIGNED32

Returns the specified RADIUS AVP as an unsigned integer.

RADIUS.NEW_VENDOR_SPEC_AVP(<ID>, <definition>)

Adds the specified extended vendor specific AVPs to the connection. For <ID>, substitute a long number. For <definition>, substitute a string that contains the data for the AVP.

RADIUS.REQ.AVP_START

Returns the location between the end of the RADIUS header and the start of the AVPs. Used in rewrite actions.

Example:

add rewrite action insert1 insert_after radius.req.avp_start radius.new_avp(33, "NEW AVP")

RADIUS.REQ.AVP_END

Returns the location at the end of radius message (or in other words end of all AVPs) in radius message. Used when performing rewrite actions.

Example:

add rewrite action insert2 insert_before radius.req.avp_end "radius.new_avp(33, \"NEW AVP\")"

RADIUS.REQ.AVP_LIST

Returns the location at the start of the AVPs in a RADIUS message, and the length of the RADIUS message, excluding the header. In other words, returns all AVPs in a RADIUS message. Used to perform Rewrite actions.

Example:

add rewrite action insert3 insert_before_all radius.req.avp_list "radius.new_avp(33, \"NEW AVP\")" -search "avp(33)"

Valid Rewrite-Action Types for RADIUS

The Rewrite action types that can be used with RADIUS expressions are:

- INSERT_AFTER
- INSERT_BEFORE
- INSERT_AFTER_ALL
- INSERT_BEFORE_ALL

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- DELETE
- DELETE_ALL
- REPLACE
- REPLACE_ALL

All INSERT_ actions can be used to insert a RADIUS AVP into a RADIUS connection.

Use Cases

Following are use cases for RADIUS with rewrite.

Rewriting the User-Name AVP

To configure the rewrite feature to remove the Domain\ string from the RADIUS user-name AVP, begin by creating a rewrite REPLACE action as shown in the example below. Use the action in a Rewrite policy that selects all RADIUS requests. Bind the policy to a global bind point. When you do so, set the priority the appropriate level to allow any block or reject policies to take effect first, but ensure that all requests that are not blocked or rejected are rewritten. Set the Goto Expression (gotoPriorityExpr) to NEXT to continue policy evaluation, and attach the policy to the RADIUS_REQ_DEFAULT queue.

Example:

```
add rewrite action rwActRadiusDomainDel replace radius.req.user_name 'radius.new_avp(0, "radius.req.user_name.value.AFTER_STR(\\)")/
add rewrite policy rwPolRadiusDomainDel true rwActRadiusDomainDelbind rewrite global rwPolRadiusDomainDel 100 NEXT -type RADIUS_REQ_DEFAULT
```

Inserting a Vendor-Specific AVP

To configure Rewrite action to insert a Vendor-Specific AVP containing the contents of the MSISDN field, begin by creating a rewrite INSERT action that inserts the MSISDN field into the request. Use the action in a Rewrite policy that selects all RADIUS requests. bind the policy to global, setting the priority to an appropriate level and the other parameters as shown in the following example.

Example:

```
add rewrite action rwActRadiusInsMSISDN insert_after radius.req.avp_start RADIUS.NEW_VENDOR_SPEC_AVP(<VENDOR ID>, "RADIUS.NEW_AVP(<Attribute Code>, <MSISDN>)")
add rewrite policy rwPolRadiusInsMSISDN true rwActRadiusInsMSISDN
bind rewrite global rwPolRadiusInsMSISDN 100 NEXT -type RADIUS_REQ_DEFAULT
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Diameter Support for Rewrite

Apr 09, 2014

The Rewrite feature now supports the Diameter protocol. You can configure Rewrite to modify Diameter requests and response as you would HTTP or TCP requests and responses, allowing you to use Rewrite to manage the flow of Diameter requests and make necessary modifications. For example, if the "Origin-Host" value in a Diameter request is inappropriate, you can use Rewrite to replace it with a value that is acceptable to the Diameter server.

To configure Rewrite to modify a Diameter request

To configure the Rewrite feature to replace the Origin-Host in a diameter request with a different value, at the command prompt, type the following commands:

- add rewrite action <actname> replace "DIAMETER.REQ.AVP(264,\"netscaler.example.net\")"
  For <actname>, substitute a name for your new action. The name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For netscaler.example.net, substitute the Host-Origin that you want to use instead of the original Host-Name.
- add rewrite policy <polname> "diameter.req.avp(264).value.eq(\"host.example.com\")" <actname>
  For <polname>, substitute a name for your new policy. As with <actname>, the name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For host.example.com, substitute the name of the Host-Origin that you want to change. For <actname>, substitute the name of the action that you just created.
- bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST
  For <vservername>, substitute the name of the load balancing virtual server to which you want to bind the policy. For <polname>, substitute the name of the policy you just created. For <priority>, substitute a priority for the policy.

Example

To create a Rewrite action and policy to modify all Diameter Host-Origins of "host.example.com" to "netscaler.example.net", you could add the following action and policy, and bind the policy as shown.

```
> add rewrite action rw_act_replace_avp replace "diameter.req.avp(264)" "diameter.new.avp(264,\"netscaler.example.net\")"
> add rewrite policy rw_diam_pol "diameter.req.avp(264).value.eq(\"client.realm2.net\")" rw_act_replace_avp
> bind lb vserver vs1 -policyName rw_diam_pol -priority 10 -type REQUEST

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

DNS Support for the Rewrite Feature

Oct 28, 2015

You can configure the rewrite feature to modify DNS requests and responses, as you would for HTTP or TCP requests and responses. You can use rewrite to manage the flow of DNS requests, and make necessary modifications in the header, or in the answer section. For example, if the DNS response does not have the AA bit set in the header flag, you can use rewrite to set the AA bit in the DNS response and send it to the client.

DNS Expressions

In a rewrite configuration, you can use the following NetScaler expressions to refer to various portions of a DNS request or response:

Expressions	Descriptions
DNS.REQ.HEADER.FLAGS.IS_SET <FLAG>	Returns True if any of the following flags are set in the DNS request. • QR • AA • TC • RD • RA • AD • CD
DNS.REQ.HEADER.FLAGS. SET <FLAG>	Sets the specified flag. Note: You can set only the RD or CD flag in the header.
DNS.REQ.HEADER.FLAGS. UNSET <FLAG>	Unsets the specified flag. Note: You can unset only the RD or CD flag in the header.
DNS.REQ.HEADER.OPCODE.EQ <pcode types>	Checks the opcode type in the DNS request. Returns True or False, indicating whether the opcode type in the DNS request matches the specified opcode type.
DNS.RES.HEADER.FLAGS.IS_SET <FLAG>	Returns True if any of the following specified flags are set in the DNS response. • QR • AA • TC • RD • RA • AD • CD
DNS.RES.HEADER.FLAGS.SET <FLAG>	Sets the specified flag.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expressions	Descriptions
DNS.RES.HEADER.FLAGS.UNSET <FLAG>	Unsets the specified flag.
DNS.REQ.HEADER.OPCODE.NE <opcode type>	Checks the opcode type in the DNS request. Returns True or False, indicating whether the opcode type in the DNS request matches the specified opcode type.
DNS.REQ.HEADER.OPCODE.SET <opcode type>	Sets the specified opcode type in the DNS request.
DNS.RES.HEADER.RCODE.SET <rcode type>	Sets the rcode type in the DNS response.
DNS.NEW_RRSET_A <ip_add, ttl>	Replaces the Answer section in the DNS response with the specified IPv4 address and TTL value.
DNS.NEW_RRSET_AAAA <ipv6, ttl>	Replaces the Answer section in the DNS response with the specified IPv6 address and TTL value.
DNS.REQ.HEADER.FLAGS.GET_STRING_REPRESENTATION	Returns the DNS flags in string format that can be used for audit logging.
DNS.RES.HEADER.FLAGS.GET_STRING_REPRESENTATION	Returns the DNS flags in string format that can be used for audit logging.

DNS Bind Points

The following global bind points are available for policies that contain DNS expressions.

Bind Points	Description
DNS_REQ_OVERRIDE	Override request policy queue.
DNS_REQ_DEFAULT	Standard request policy queue.
DNS_RES_OVERRIDE	Override response policy queue.
DNS_RES_DEFAULT	Standard response policy queue.

In addition to the default bind points, you can create policy labels of type DNS_REQ or DNS_RES and bind DNS policies to them.

Rewrite Action Types for DNS

- **replace_dns_answer_section**—This action replaces the DNS answers section with the defined expression in the DNS policy.
- **replace_dns_header_field**—Checks the opcode type in the DNS request. Returns True or False, indicating whether the opcode type in the DNS request matches the specified opcode type. This action replaces the DNS header section with

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the defined expression in the DNS policy.

Configuring Rewrite Policies for DNS

The following procedure uses the NetScaler command line to configure a rewrite action and policy and bind the policy to a rewrite-specific global bind point.

To configure Rewrite action and policy, and bind the policy for DNS

At the command prompt, type the following commands:

1. add rewrite action <actName> <actType>
   For <actname>, substitute a name for your new action. The name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For <actType>, specify the rewrite action types provided for DNS expressions.

2. add rewrite policy <polName> <rule> <actName>
   For <polname>, substitute a name for your new policy. For <actname>, the name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For <actname>, substitute the name of the action that you just created.

3. bind rewrite global <polName> <priority> < gotoPriorityExpression> -type <bindPoint>
   For <polName>, substitute the name of the policy that you just created. For <priority>, specify the priority of the policy. For <bindPoint>, substitute one of the rewrite -specific global bind points.

Example

Set the AA bit of DNS request to load balance virtual server

The following commands configure the NetScaler appliance to act as an authoritative DNS server for all the queries that it serves.

```
add rewrite action set_aa replace_dns_header_field dns.req.header.flags.set(aa)
add rewrite policy pol !dns.req.header.flags.is_set(aa)  set_aa
bind rewrite global  pol  100  -type dns_res_override
```

Modify the response answer and header section

If the server responds with an NX domain, you can set the rewrite action to replace the response with specified IP address. A NOPOLICY-REWRITE enables you to invoke an enternal bank without processing an expression (a rule). This entry is a dummy policy that does not contain a rule but directs the entry to a policy label or virtual server specific policy banks.

```
add rewrite action set_aa_res replace_dns_header_field "dns.res.header.flags.set(aa)"
add rewrite action modify_nxdomain_res replace_dns_answer_section "dns.new_rrset_a(\"10.102.218.160\",300)"
add rewrite policy set_res_aa true set_aa_res
add add rewrite policy modify_answer "dns.RES.HEADER.RCODE.EQ(nxdomain) && dns.RES.QUESTION.TYPE.EQ(A)"
modify_nxdomain_res
add rewrite policylabel MODIFY_NODATA dns_res
bind rewrite policylabel MODIFY_NODATA modify_answer 10 END
bind rewrite policylabel MODIFY_NODATA set_res_aa 11 END
bind lb vserver v1 -policyName NOPOLICY-REWRITE -priority 11 -gotoPriorityExpression END -type
RESPONSE -invoke policylabel MODIFY_NODATA
```

Limitations

- Rewrite policies are evaluated only if the NetScaler appliance is configured as a DNS proxy server and there is a cache miss.
- If the Recursion Available (RA) flag in the header is set to YES, the RA flag will not be modified in the rewrites.
- If the RA flag in the header is set to YES, the CD flag in the header is modified regardless of any rewrite action.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

String Maps

Apr 20, 2015

You can use string maps to perform pattern matching in all NetScaler features that use the default policy syntax. A string map is a NetScaler entity that consists of key-value pairs. The keys and values are strings in either ASCII or UTF-8 format. String comparison uses two new functions, MAP_STRING(<string_map_name>) and IS_STRINGMAP_KEY(<string_map_name>).

A policy configuration that uses string maps performs better than one that does string matching through policy expressions, and you need fewer policies to perform string matching with a large number of key-value pairs. String maps are also intuitive, simple to configure, and result in a smaller configuration.

How String Maps Work

String maps are similar in structure to pattern sets (a pattern set defines a mapping of index values to strings; a string map defines a mapping of strings to strings) and the configuration commands for string maps (commands such as add, bind, unbind, remove, and show) are syntactically similar to configuration commands for pattern sets. Also, as with index values in a pattern set, each key in a string map must be unique across the map. The following table illustrates a string map called url_string_map, which contains URLs as keys and values.

Table 1. String Map "url_string_map"

Key	Value
/url_1.html	http://www.redirect_url_1.com/url_1.html
/url_2.html	http://www.redirect_url_2.com/url_2.html
/url_3.html	http://www.redirect_url_1.com/url_1.html

The following table describes the two functions that have been introduced to enable string matching with keys in a string map. String matching is always performed with the keys. Additionally, the following functions perform a comparison between the keys in the string map and the complete string that is returned by the expression prefix. The examples in the descriptions refer to the preceding example.

Table 2. String Map Functions

Function	Description
<TEXT>.MAP_STRING(<string_map_name>)	Checks whether the value returned by the expression prefix TEXT matches any of the keys in the string map, and returns the value that corresponds to the key. If no key in the string map matches the value returned by the expression prefix, the function returns a null string. The IGNORECASE and NOIGNORECASE functions can be used for case-insensitive and case-sensitive comparison, respectively. **Example 1:** HTTP.REQ.URL.MAP_STRING("url_string_map") checks whether the string returned by HTTP.REQ.URL is a key in the string map url_string_map. If the value of HTTP.REQ.URL is /url_1.html, the function returns http://www.redirect_url_1.com/url_1.html. **Example 2:** HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).MAP_STRING("url_string_map") checks whether the string returned by HTTP.REQ.URL is a key in the string map url_string_map. The comparison does not consider case. If the string returned by HTTP.REQ.URL is /URL_1.html, the function returns http://www.redirect_url_1.com/url_1.html. **Parameters:**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Function	Description
	string_map_name - The string map.
<TEXT>.IS_STRINGMAP_KEY(<string_map_name>)	Returns TRUE if the string returned by the expression prefix TEXT is a key in the string map. The IGNORECASE and NOIGNORECASE functions can be used for case-insensitive and case-sensitive string matching, respectively. **Example 1:** HTTP.REQ.URL.IS_STRINGMAP_KEY("url_string_map") returns TRUE if the value of HTTP.REQ.URL is one of the keys in url_string_map. **Example 2:** HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE). IS_STRINGMAP_KEY("url_string_map") returns TRUE if the value of HTTP.REQ.URL is one of the keys in url_string_map. In this case, key lookup does not consider case. Therefore, the function returns TRUE even if the value of HTTP.REQ.URL is /URL_3.html. **Parameters:** string_map_name - The string map.

Configuring a String Map

You first create a string map and then bind key-value pairs to it. You can create a string map from the command line interface (CLI) or the configuration utility.

To configure a string map by using the command line interface

At the command prompt, do the following:

1. Create a string map.
   add policy stringmap <name> -comment <string>

2. Bind a key-value pair to the string map.
   bind policy stringmap <name> <key> <value>

   Example:

   > bind policy stringmap url_string_map1 "/url_1.html" "http://www.redirect_url_1.com/url_1.html"
   To configure a string map by using the configuration utility

   Create a string map and bind the key-value pair to the created entity.

   Navigate to **AppExpert** > **String Maps**, click **Add** and specify the relevant details.

Example: Responder Policy With a Redirect Action

The following use case involves a responder policy with a redirect action. In the example below, the first four commands create the string map url_string_map and bind the three key-value pairs used in the earlier example. After creating the map and binding the key-value pairs, you create a responder action (act_url_redirects) that redirects the client to the corresponding URL in the string map or to www.default.com. You also configure a responder policy (pol_url_redirects) that checks whether requested URLs match any of the keys in url_string_map and then performs the configured action. Finally, you bind the responder policy to the content switching virtual server that receives the client requests that are to be evaluated.

add stringmap url_string_map

bind stringmap url_string_map /url_1.html http://www.redirect_url_1.com/url_1.html

bind stringmap url_string_map /url_2.html http://www.redirect_url_2.com/url_2.html

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
bind stringmap url_string_map /url_3.html http://www.redirect_url_1.com/url_1.html

add responder action act_url_redirects redirect 'HTTP.REQ.URL.MAP_STRING("url_string_map") ALT "www.default.com"' -bypassSafetyCheck yes

add responder policy pol_url_redirects TRUE act_url_redirects

bind cs vserver csw_redirect -policyname pol_url_redirects -priority 1 -type request
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Traffic Management

May 13, 2016

The following sections cover configuration and installation information for NetScaler traffic management features.

Cache Redirection	Analyzes incoming requests and forwards the requests for already cached data to cache servers. Dynamic HTTP requests and non-cacheable requests are forwarded to the origin servers. Cache redirection is a policy-based feature.
Content Switching	Analyzes client requests and redirects the requests to specific servers on the basis of geographical area, authorization credentials, and device from which the request was initiated.
DataStream	Ensures optimal distribution of traffic from the application and web servers to the database servers. Enables you to segment traffic according to information in the SQL query and on the basis of database names, user names, character sets, and packet size.
Domain Name System	Provides authoritative domain name server (ADNS server) functionality for a domain. The NetScaler appliance functions as a DNS end resolver and forwarder, and also helps in name resolution when fully qualified domain names are not configured.
Firewall Load Balancing	Distributes the traffic across multiple firewalls, providing fault tolerance, increased throughput, and high availability.
Global Server Load Balancing	Enables disaster recovery and ensures continuous availability of applications by protecting against points of failure in a wide area network (WAN).
Link Load Balancing	Load balances outbound traffic across multiple Internet connections to transmit packets seamlessly over the best possible link.
Load Balancing	Distributes user requests for web pages and other protected applications across multiple servers to prevent server overloading and failure. Load balancing also provides fault tolerance.
SSL Offload and Acceleration	Offloads SSL processing from a server to the NetScaler appliance to accelerate SSL transactions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cache Redirection

Mar 30, 2012

In a typical deployment, different clients ask web servers for the same content repeatedly. To relieve the origin web server of processing each request, a NetScaler® appliance with cache redirection enabled can serve this content from a cache server instead of from the origin server.

The NetScaler analyzes incoming requests, sends requests for cacheable data to cache servers, and sends non-cacheable requests and dynamic HTTP requests to origin servers.

Cache redirection is a policy-based feature. By default, requests that match a policy are sent to the origin server, and all other requests are sent to a cache server. For testing or maintenance, you might want to skip policy evaluation and direct all requests to the cache or to the origin server.

You can combine content switching with cache redirection to cache selective content and serve content from specific cache servers for specific types of requested content.

A NetScaler configured for cache redirection can be deployed at the edge of a network, in front of the origin server, or anywhere along the network backbone. In an edge deployment, commonly used by Internet Service Providers (ISPs), cable companies, content delivery distribution networks, and enterprise networks, the NetScaler resides directly in front of the clients. In a server-side deployment, the NetScaler is closer to the origin servers.

Cache redirection is used most commonly with the HTTP service type, but it also supports the secure HTTPS protocol.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cache Redirection Policies

Mar 30, 2012

A cache redirection virtual server applies cache redirection policies to each incoming request. By default, if a request matches one of the configured policies, it is considered non-cacheable, and the NetScaler appliance sends it to the origin server. Other requests are sent to a cache server. This behavior can be reversed, so that requests that match configured cache redirection policies are sent to cache servers.

The NetScaler provides a set of policies for cache redirection. If these built-in policies are not adequate for your deployment, you can configure user-defined cache redirection policies.

Note: Once you have determined which built-in cache redirection policies to use, or have created user-defined policies, proceed with configuring cache redirection. To use this feature, you must configure at least one cache redirection virtual server, and, for normal operation, you must bind at least one cache redirection policy to that virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Built-in Cache Redirection Policies

May 14, 2015

The NetScaler appliance provides built-in cache redirection policies that handle typical cache requests. These policies are based on HTTP methods, the URL or URL tokens of the incoming request, the HTTP version, or the HTTP headers and their values in the request.

Built-in cache redirection policies can be directly bound to a virtual server and do not need further configuration.

Cache redirection policies use two types of NetScaler expressions languages, classic and default syntax. For more information about these languages, see Policies and Expressions.

Built-in Classic Cache Redirection Policies

Built-in cache redirection policies based on classic expressions are called *classic cache redirection policies*. For a complete description of classic expressions and how to configure them, see Policies and Expressions.

The classic cache redirection policies evaluate basic characteristics of traffic and other data. For example, classic cache redirection policies can determine whether an HTTP request or response contains a particular type of header or URL.

The NetScaler appliance provides the following built-in classic cache redirection policies:

Built-In Policy Name	Description
bypass-non-get	Bypass the cache if the request uses an HTTP method other than GET.
bypass-cache-control	Bypass the cache if the request header contains a Cache-Control: no-cache or Cache-Control: no-store header, or if the HTTP request contains a pragma header.
bypass-dynamic-url	Bypass the cache if the URL suggests that the content is dynamic, as indicated by the presence of any of the following extensions: • cgi • asp • exe • cfm • ex • shtml • htx Also bypass the cache if the URL starts with any of the following: • /cgi-bin/ • /bin/ • /exec/
bypass-urltokens	Bypass the cache because the request is dynamic, as indicated by one of the following tokens in the URL: ?, !, or =.
bypass-cookie	Bypass the cache for any URL that has a cookie header and an extension other than .gif or .jpg.

Built-in Default Syntax Cache Redirection Policies

Built-in cache redirection policies based on default syntax expressions are called *default syntax cache redirection policies*. For a complete description of default syntax expressions and how to configure them, see Policies and Expressions.

In addition to the same types of evaluations done by classic cache redirection policies, default syntax cache redirection policies enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example, directing the request to either cache or origin server).

NetScaler appliances provide the following two built-in actions for the default syntax cache redirection policies:

- CACHE
- ORIGIN

As implied by their names, they direct the request to the cache server or the origin server, respectively.

Note: If you are using the built-in default syntax cache redirection policy, you cannot modify the action.

The NetScaler appliance provides the following built-in default syntax cache redirection policies:

Built-In Policy Name	Description
bypass-non-get_adv	Bypass the cache if the request uses an HTTP method other than GET.
bypass-cache-control_adv	Bypass the cache if the request header contains a Cache-Control: no-cache or Cache-Control: no-store header, or if the HTTP request contains a pragma header.
bypass-dynamic-url_adv	Bypass the cache if the URL suggests that the content is dynamic, as indicated by the presence of any of the following extensions: • cgi • asp • exe • cfm • ex • shtml

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Built-In Policy Name	Description
	• htx Also bypass the cache if the URL starts with any of the following: • /cgi-bin/ • /bin/ • /exec/
bypass-urltokens_adv	Bypass the cache because the request is dynamic, as indicated by one of the following tokens in the URL: ?, !, or =.
bypass-cookie_adv	Bypass the cache for any URL that has a cookie header and an extension other than .gif or .jpg.

Displaying the Built-in Cache Redirection Policies

Updated: 2013-08-23

You can display the available cache redirection polices by using the command line interface or the configuration utility.

To display the built-in cache redirection policies by using the command line interface

At the command prompt, type:

show cr policy [<policyName>]

Example

```
> show cr policy
1)      Cache-By-Pass RULE: NS_NON_GET          Policy:bypass-non-get
2)      Cache-By-Pass RULE: (NS_CACHECONTROL_NOSTORE || NS_CACHECONTROL_NOCACHE || NS_HEADER_PRAGMA)    Policy:bypass-cache-control
3)      Cache-By-Pass RULE: (NS_EXT_CGI || NS_EXT_ASP || NS_EXT_EXE || NS_EXT_CFM || NS_EXT_EX || NS_EXT_SHTML || NS_EXT_HTX) || (NS_URL_PATH_CGIBIN || NS_URL_PATH_E
4)      Cache-By-Pass RULE: NS_URL_TOKENS       Policy:bypass-urltokens
5)      Cache-By-Pass RULE: (NS_HEADER_COOKIE && NS_EXT_NOT_GIF && NS_EXT_NOT_JPEG)     Policy:bypass-cookie
 Done
>
```

To display the built-in cache redirection policies by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Policies. The configured cache redirection policies appear in the details pane.
2. Select one of the configured policies to view details.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Cache Redirection Policy

Jan 01, 2018

A cache redirection policy includes one or more expressions (also called *rules*). Each expression represents a condition that is evaluated when the client request is compared to the policy.

You do not explicitly configure actions for cache redirection policies. By default, the NetScaler appliance considers any request that matches a policy to be non-cacheable and directs the request to the origin server instead of the cache.

Cache redirection policies based on the classic policy format are called *classic cache redirection policies*. Each such policy has a name and includes a classic expression or a set of classic expressions that are combined by using logical operators.

For classic cache redirection policies, you do not explicitly configure actions for the policies. By default, the NetScaler appliance considers any request that matches a policy to be non-cacheable and directs the request to the origin server instead of the cache.

Cache redirection policies based on the newer policy format are called *default syntax redirection policies*. Such policy has a name and includes a default syntax expression, or a set of default syntax expressions that are combined by using logical operators, and the following built-in actions:

- CACHE
- ORIGIN

For more information about classic expressions and default syntax expressions, see Policies and Expressions.

To add a cache redirection policy by using the command line interface

At the command prompt, type the following commands to add a cache redirection policy and verify the configuration:

- add cr policy <policyName> **-rule** <expression>
- show cr policy [<policyName>]

Examples

Policy with a simple expression:

```
> add cr policy Policy-CRD-1 -rule "REQ.HTTP.URL != /*.jpeg"
 Done
> show cr policy Policy-CRD-1
      Cache-By-Pass RULE: REQ.HTTP.URL != '/*.jpeg'   Policy:Policy-CRD-1
 Done
>
```

Policy with a compound expression:

```
> add cr policy Policy-CRD-2 -rule "REQ.HTTP.METHOD == POST && (REQ.HTTP.URL == /*.cgi || REQ.HTTP.URL != /*.gif)"
 Done
> show cr policy Policy-CRD-2
      Cache-By-Pass RULE: REQ.HTTP.METHOD == POST && (REQ.HTTP.URL == '/*.cgi' || REQ.HTTP.URL != '/*.gif')   Policy:Policy-CRD-2
 Done
>
```

Policy that evaluates a header:

```
> add cr policy Policy-CRD-3 -rule "REQ.HTTP.HEADER If-Modified-Since EXISTS"
 Done
> show cr policy Policy-CRD-3
      Cache-By-Pass RULE: REQ.HTTP.HEADER If-Modified-Since EXISTS    Policy:Policy-CRD-3
 Done
>
```

To add a default syntax cache redirection policy by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to add a cache redirection policy and verify the configuration:

- add cr policy <policyName> **-rule** <expression> [-action<string>] [-logAction<string>]
- show cr policy [<policyName>]

Examples

Policy with a simple expression:

```
> add cr policy crpol1 -rule !(HTTP.REQ.URL.ENDSWITH(".jpeg”)) -action origin
 Done
> show cr policy crpoll
      Policy: crpol1 Rule: !(HTTP.REQ.URL.ENDSWITH(".jpeg")) Action: ORIGIN
 Done
>
```

Policy with a compound expression:

```
> add cr policy crpol11 -rule "http.req.method.eq(post) && (HTTP.REQ.URL.ENDSWITH(\".gif\") || HTTP.REQ.URL.ENDSWITH(\".cgi\"))"  -action cache
 Done
> show cr policy crpol11
      Policy: crpol11 Rule: http.req.method.eq(post) && (HTTP.REQ.URL.ENDSWITH(".gif") || HTTP.REQ.URL.ENDSWITH(".cgi")) Action: CACHE
 Done
>
```

Policy that evaluates a header:

```
> add cr policy crpol12 -rule http.req.header("If-Modified-Since").exists -action origin
 Done
> show cr policy crpol12
      Policy: crpol12 Rule: http.req.header("If-Modified-Since").exists Action: ORIGIN
 Done
>
```

To modify or remove a cache redirection policy by using the command line interface

- To modify a cache redirection policy, use the set cr policy command, which is just like add cr policy command, except that you enter the name of an existing policy.
- To remove a policy, use the rm cr policy command, which accepts only the <name> argument. If the policy is bound to a virtual server, you have to unbind the policy, before you can remove it.

For the details of unbinding a cache redirection policy, see Unbinding a Policy from a Cache Redirection Virtual Server.

To configure a cache redirection policy with a simple expression by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Policies.
2. In the details pane, click Add.
3. In the Create Cache Redirection Policy dialog box, in the Name* text box, type the name of the policy, and then in the Expression area, click Add.
4. To configure a simple expression, enter the expression. Following is an example of an expression that checks for a .jpeg extension in a URL:

   - Expression Type-General
   - Flow Type -REQ
   - Protocol -HTTP
   - Qualifier -URL
   - Operator - !=
   - Value* - /*.jpeg

   The simple expression in the following example checks for an If-Modified-Since header in a request:

   - Expression Type -General

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Flow Type -REQ
- Protocol -HTTP
- Qualifier -HEADER
- Operator -EXISTS
- Header Name -If-Modified-Since

5. When you are finished entering the expression, click OK or Create, and then click Close.

To configure a cache redirection policy with a compound expression by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Policies.
2. In the details pane, click Add.
3. In the Name text box, enter a name for the policy.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols. You should choose a name that will make it easy for others to tell what type of content this policy was created to detect.

4. Choose the type of compound expression that you want to create. Your choices are:
   - **Match Any Expression**. The policy matches the traffic if one or more individual expressions match the traffic.
   - **Match All Expressions**. The policy matches the traffic only if every individual expression matches the traffic.
   - **Tabular Expressions**. Switches the Expressions list to a tabular format with three columns. In the rightmost column, you place one of the following operators:
     - The AND [&&] operator, to require that, to match the policy, a request must match both the current expression and the following expression.
     - The OR [||] operator, to require that, to match the policy, a request must match either the current expression or the following expression, or both. Only if the request does not match either expression does it not match the policy.

     You can also group expressions in nested subgroups by selecting an existing expression and clicking one of the following operators:

     - The BEGIN SUBGROUP [+ ( ] operator, which tells the NetScaler appliance to begin a nested subgroup with the selected expression. (To remove this operator from the expression, click -(.)
     - The END SUBGROUP [+)] operator, which tells the NetScaler appliance to end the current nested subgroup with the selected expression. (To remove this operator from the expression, click -) .)
   - **Advanced Free-Form**. Switches off the Expressions Editor entirely and turns the Expressions list into a text area in which you can type a compound expression. This is both the most powerful and the most difficult method of creating a policy expression, and is recommended only for those thoroughly familiar with the NetScaler classic expressions language.
     For more information about creating classic expressions in the Advanced Free-Form text area, see "Configuring Classic Policies and Expressions".

     Caution: If you switch to Advanced Free Form expression editing mode, you cannot switch back to any of the other modes. Do not choose this expression editing mode unless you are sure that you want to use it.
5. If you chose Match Any Expression, Match All Expressions, or Tabular Expressions, click **Add** to display the Add Expression dialog box.

   You should leave the expression type set to General for cache redirection policies.

6. In the Flow Type drop-down list, choose a flow type for your expression.

   The flow type determines whether the policy examines incoming or outgoing connections. You have two choices:

   - **REQ**. Configures the NetScaler appliance to examine incoming connections, or requests.
   - **RES**. Configures the appliance to examine outgoing connections, or responses.
7. In the Protocol drop-down list, choose a protocol for your expression.

   The protocol determines the type of information that the policy examines in the request or response. Depending upon whether you chose REQ or RES in the previous drop-down list, either all four or only three of the following choices are available:

   - **HTTP**. Configures the appliance to examine the HTTP header.
   - **SSL**. Configures the appliance to examine the SSL client certificate. Available only if you chose REQ (requests) in the previous drop-down list.
   - **TCP**. Configures the appliance to examine the TCP header.
   - **IP**. Configures the appliance to examine the source or destination IP address.
8. Choose a qualifier for your expression from the Qualifier drop-down list.

   The contents of the Qualifier drop-down list depend on which protocol you chose. The following table describes the choices available for each

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

protocol.

Table 1. Cache Redirection Policy Qualifiers Available for Each Protocol

Protocol	Qualifier	Definition
HTTP	METHOD	HTTP method used in the request.
	URL	Contents of the URL header.
	URLTOKENS	URL tokens in the HTTP header.
	VERSION	HTTP version of the connection.
	HEADER	Header portion of the HTTP request.
	URLLEN	Length of the contents of the URL header.
	URLQUERY	Query portion of the contents of the URL header.
	URLQUERYLEN	Length of the query portion of the URL header.
SSL	CLIENT.CERT	SSL client certificate as a whole.
	CLIENT.CERT.SUBJECT	Contents of the client certificate subject field.
	CLIENT.CERT.ISSUER	Client certificate issuer.
	CLIENT.CERT.SIGALGO	Signature algorithm used in the client certificate.
	CLIENT.CERT.VERSION	Client certificate version.
	CLIENT.CERT.VALIDFROM	Date from which the client certificate is valid. (The start date.)
	CLIENT.CERT.VALIDTO	Date after which the client certificate is no longer valid. (The end date.)
	CLIENT.CERT.SERIALNUMBER	Client certificate serial number.
	CLIENT.CIPHER.TYPE	Encryption method used in the client certificate.
	CLIENT.CIPHER.BITS	Number of significant bits in the encryption key.
	CLIENT.SSL.VERSION	SSL version of the client certificate.
TCP	SOURCEPORT	Source port of the TCP connection.
	DESTPORT	Destination port of the TCP connection.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Protocol	Qualifier	Definition
	MSS	Maximum segment size (MSS) of the TCP connection.
IP	SOURCEIP	Source IP address of the connection.
	DESTIP	Destination IP address of the connection.

9. Choose the operator for your expression from the Operator drop-down list.

   Your choices depend on the qualifier you chose in the previous step. The complete list of operators that can appear in this drop-down list is:

   - == . Matches the following text string exactly.
   - != . Does not match the following text string.
   - > . Is greater than the following integer.
   - CONTAINS . Contains the following text string.
   - CONTENTS . The contents of the designated header, URL, or URL query.
   - EXISTS . The specified header or query exists.
   - NOTCONTAINS . Does not contain the following text string.
   - NOTEXISTS . The specified header or query does not exist.

   If you want this policy to operate on requests sent to a specific Host, you can leave the default, the equals (==) sign.

10. If the Value text box is visible, type the appropriate string or number into the text box.
    For example, if you want this policy to select requests sent to the host shopping.example.com, you would type that string in the Value text box.

11. If you chose HEADER as the qualifier, type the header you want in the Header Name text box.
12. Click OK to add your expression to the Expression list.
13. Repeat steps 4 through 11 to create additional expressions.
14. Click Close to close the Add Expression dialog box and return to the Create Cache Redirection Policy dialog box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cache Redirection Configurations

Mar 30, 2012

Depending on your deployment and network topology, you can configure one of the following types of cache redirection:

- **Transparent**. A transparent cache can reside on a variety of points along a network backbone to alleviate traffic along the delivery route. In transparent mode, the cache redirection virtual server intercepts all traffic flowing to the NetScaler appliance and applies cache redirection policies to determine whether content should be served from the cache or from the origin server.
- **Forward proxy**. A forward proxy cache server resides on the edge of an enterprise LAN and faces the WAN. In the forward proxy mode, the cache redirection virtual server resolves the hostname of the incoming request by using a DNS server and forwards requests for non-cacheable content to the resolved origin servers. Cacheable requests are sent to the configured cache servers.
- **Reverse proxy**. Reverse proxy caches are configured for specific origin servers. Incoming traffic directed to the reverse proxy, can either be served from a cache server or be sent to the origin server with or without modification to the URL.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Transparent Redirection

Jan 01, 2018

When you configure transparent cache redirection, the NetScaler appliance evaluates all traffic it receives, to determine whether it is cacheable. This mode alleviates traffic along the delivery route and is often used when the cache server resides on the backbone of an ISP or carrier.

By default, cacheable requests are sent to a cache server, and non-cacheable requests to the origin server. For example, when the NetScaler appliance receives a request that is directed to a web server, it compares the HTTP headers in the request with a set of policy expressions. If the request does not match the policy, the appliance forwards the request to a cache server. If the request does match a policy, the appliance forwards the request, unchanged, to the web server.

For details on how to modify this default behavior, see Directing Policy Hits to the Cache instead of the Origin.

To configure transparent redirection, first enable cache redirection and load balancing, and configure edge mode. Then, create a cache redirection virtual server with a wildcard IP address (*), so that this virtual server can receive traffic coming to the NetScaler on any IP address the appliance owns. To this virtual server, bind cache redirection policies that describe the types of requests that should not be cached. Then, create a load balancing virtual server that will receive traffic from the cache redirection virtual server for cacheable requests. Finally, create a service that represents a physical cache server and bind it to the load balancing virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Cache Redirection and Load Balancing

Oct 31, 2013

The NetScaler cache redirection and load balancing features are not enabled by default. They must be enabled before any cache redirection configuration can take effect.

To enable cache redirection and load balancing by using the command line interface

At the command prompt, type the following command to enable cache redirection and load balancing and verify the settings:

- enable ns feature cr lb
- show ns feature

Example

```
> enable ns feature cr lb
 Done
> show ns feature

      Feature                  Acronym          Status
      -------                 -------          ------
 1)   Web Logging              WL               ON
 2)   Surge Protection         SP               ON
 3)   Load Balancing           LB               ON
 4)   Content Switching        CS               ON
 5)   Cache Redirection        CR               ON
 6)   Sure Connect

        ...
        ...
        ...

 23)  HTML Injection           HTMLInjection    ON
 24)  NetScaler Push           push             OF
 Done
>
```

To enable cache redirection and load balancing by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. To enable cache redirection, in the details pane, under Modes and Features, click Configure advanced features.
   1. In Configure Advanced Features dialog box, select the check box next to the Cache Redirection, and then click OK.
   2. In Enable/Disable Feature(s)? dialog box, click Yes.
3. To enable load balancing, in the details pane, under Modes and Features, click Configure basic features.
   1. In Configure Basic Features dialog box, select the check box next to the Load Balancing, and then click OK.
   2. In Enable/Disable Feature(s)? dialog box, click Yes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Edge Mode

Nov 07, 2016

When deployed at the edge of a network, the NetScaler appliance dynamically learns about the servers on that network. Edge mode enables the appliance to dynamically learn about up to 40,000 HTTP servers and proxy TCP connections for these servers.

This mode turns on the collection of statistics for the dynamically learned services and is typically used in transparent deployments for cache redirection.

To enable edge mode by using the command line interface

At the command prompt, type the following commands to enable edge mode and verify the setting:

- enable ns mode Edge
- show ns mode

Example

```
> enable ns mode edge
 Done

> show ns mode

      Mode                    Acronym          Status
      -------                 -------          ------
        ...
        ...
        ...
 6)   MAC-based forwarding       MBF              ON
 7)   Edge configuration         Edge             ON
 8)   Use Subnet IP              USNIP            OFF
        ...
        ...
        ...
 16)   Bridge BPDUs              BridgeBPDUs      OFF
 Done
>
```

To enable edge mode by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure modes.
3. In Configure Modes dialog box, select the check box next to the Edge Configuration, and then click OK.
4. In Enable/Disable Feature(s)? dialog box, click Yes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Cache Redirection Virtual Server

Sep 10, 2013

By default, a cache redirection virtual server forwards cacheable requests to the load balancing virtual server for the cache, and forwards non-cacheable requests to the origin server (except in a reverse proxy configuration, in which non-cacheable requests are sent to a load balancing virtual server). There are three types of cache redirection virtual servers: transparent, forward proxy, and reverse proxy.

A transparent cache redirection virtual server uses an IP address of * and a port number, usually 80, that can accept HTTP traffic sent to any IP address that the NetScaler represents. As a result, you can configure only one transparent cache redirection virtual server. Any additional cache redirection virtual servers that you configure must be forward proxy or reverse proxy redirection servers.

To add a cache redirection virtual server in transparent mode by using the command line interface

At the command prompt, type the following commands to add a cache redirection virtual server and verify the configuration:

- add cr vserver <name> <serviceType> [<IPAddress> <port>] [-cacheType <cacheType>] [-redirect <redirect>]
- show cr vserver [<name>]

Example

```
add cr vserver Vserver-CRD-1 HTTP * 80 -cacheType TRANSPARENT -redirect POLICY
> show cr vserver Vserver-CRD-1
    Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
    State: UP  ARP:DISABLED
    Client Idle Timeout: 180 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    Default:      Content Precedence: RULE       Cache: TRANSPARENT
    On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
    Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF
 Done
>
```

To modify or remove a cache redirection virtual server by using the command line interface

- To modify a virtual server, use the set cr vserver command, which is just like using the add cr vserver command, except that you enter the name of an existing virtual server.
- To remove a virtual server, use the rm cr vserver command, which accepts only the <name> argument.

To add a cache redirection virtual server in transparent mode by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Cache Redirection) dialog box, specify values for the following parameters as shown:

   - Name*—name

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Port*—port

  * A required parameter

4. In the Protocol drop-down list, select a supported protocol (for example, **HTTP**). If the virtual server is to receive traffic on a port other than the standard port for the selected protocol, enter a new value in the Port field.
5. Click the Advanced tab.
6. Verify that Cache Type is set to TRANSPARENT and Redirect is set to POLICY.
7. Click Create, and then click Close. The Cache Redirection Virtual Servers pane displays the new virtual server.
8. Select the new cache redirection virtual server to display the details of its configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding Policies to the Cache Redirection Virtual Server

Aug 22, 2013

Cache redirection policies are not automatically bound to the cache redirection virtual server. A policy based cache redirection virtual server cannot function unless you bind at least one policy to it.

To bind policies to a cache redirection virtual server by using the command line interface

At the command prompt, type:

- bind cr vserver <name> -policyName <string>
- show cr vserver [<name>]

Example

```
> bind cr vserver Vserver-CRD-1 -policyName bypass-cache-control
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-dynamic-url
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-urltokens
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-cookie
 Done

> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: RULE       Cache: TRANSPARENT
      On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: bypass-dynamic-url
3)     Cache bypass  Policy: bypass-urltokens
4)     Cache bypass  Policy: bypass-cookie
 Done
>
```

To bind a user-defined policy to a cache redirection virtual server by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. Click the virtual server that you want to configure, and click Open.
3. On the Policies tab, select type of the policy and then click Insert Policy.
4. Under Policy Name column, select the policy that you want to bind.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Unbinding a Policy from a Cache Redirection Virtual Server

Aug 23, 2013

When you unbind a policy from the cache redirection virtual server, the NetScaler appliance no longer applies the policy when evaluating client requests.

To unbind a policy from a cache redirection virtual server by using the command line interface

At the command prompt, type:

- unbind cr vserver <name> -policyName <string>
- show cr vserver [<name>]

Example

```
unbind cr vserver Vserver-CR-1 -policyName bypass-non-get
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: RULE       Cache: TRANSPARENT
      On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF

1)     Cache bypass  Policy: bypass-cache-control
 Done
>
```

To unbind a user-defined policy from a cache redirection virtual server by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. Click the virtual server that you want to configure, and then click Open.
3. On the Policies tab, under Policy Name, select the policy that you want to unbind.
4. Click Unbind Policy, and then click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating a Load Balancing Virtual Server

Sep 10, 2013

The cache redirection virtual server on the NetScaler appliance can send requests to either a cache server farm, if the request is cacheable, or to the origin server farm if the request is not cacheable.

Each cache server is represented on the appliance by a service, which is bound to a load balancing virtual server that receives requests from the cache redirection virtual server and forwards those requests to the servers.

For details on configuring load balancing virtual servers and other configuration options, see "Load Balancing."

To create a load balancing virtual server by using the command line interface

At the command prompt, type the following commands to create a load balancing virtual server and verify the configuration:

- add lb vserver <name> <serviceType> [<IPAddress>] [<port>]
- show lb vserver [<name>]

Example

```
> add lb vserver Vserver-LB-CR HTTP 10.102.20.30 80
 Done
> show lb vserver Vserver-LB-CR
      Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
      State: DOWN
      Last state change was at Fri Jul  2 08:47:52 2010
      Time since last state change: 0 days, 00:00:08.470
      Effective State: DOWN
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
      No. of Bound Services :  0 (Total)       0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Vserver IP and Port insertion: OFF
      Push: DISABLED  Push VServer:
      Push Multi Clients: NO
      Push Label Rule: none
 Done
>
```

To create a load balancing virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Name*-name
- IP Address*- IPAddress
- Port*-port

* A required parameter

4. In the Protocol* drop down list, select a supported protocol (for example, **HTTP**). If the virtual server is to receive traffic on a port other than the well-known port for the selected protocol, enter a new value in the Port field.
5. Click Create, and then click Close. The Load Balancing Virtual Servers pane displays the new virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring an HTTP Service

Sep 10, 2013

On the NetScaler appliance, a service represents a physical server on the network. In the transparent cache redirection configuration, the service represents the cache server. Cacheable requests are sent by the cache redirection virtual server to the load balancing virtual server, which in turn forwards each request to the correct service, which passes it on to the cache server.

To configure an HTTP service by using the command line interface

At the command prompt, type the following commands to create an HTTP service and verify the configuration:

- add service <name> <IP> <serviceType> <port> -cacheType <cacheType>
- show service [<name>]

Example

```
> add service Service-HTTP-1 10.102.29.40 HTTP 80 -cacheType TRANSPARENT
 Done
> show service Service-HTTP-1
      Service-HTTP-1 (10.102.29.40:80) - HTTP
      State: DOWN
      Last state change was at Fri Jul  2 09:14:17 2010
      Time since last state change: 0 days, 00:00:13.820
      Server Name: 10.102.29.40
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): NO
      HTTP Compression(CMP): YES
      Idle timeout: Client: 180 sec   Server: 360 sec
      Client IP: DISABLED
      Cache Type: TRANSPARENT Redirect Mode:
      Cacheable: NO
      SC: OFF
      SP: ON
      Down state flush: ENABLED

1)     Monitor Name: tcp-default
            State: DOWN     Weight: 1
            Probes: 3       Failed [Total: 3 Current: 3]
            Last response: Failure - Time out during TCP connection establishment stage
            Response Time: N/A
 Done
>
```

To modify or remove a service by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- To modify a service, use the set service command, which is just like using the add service command, except that you enter the name of an existing service.
- To remove a service, use the rm service command, which accepts only the <name> argument.

To add an HTTP service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:

   - Service Name*—name
   - Server*— IP
   - Port*—port

   * A required parameter

4. In the Protocol* drop-down list, select a supported protocol (for example, **HTTP**).
5. Click Create, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding/Unbinding a Service to/from a Load Balancing Virtual Server

Aug 22, 2013

You must bind a service to the load balancing virtual server. This enables the load balancer to forward the request to the server that the service represents. If your configuration changes, you can unbind a service from the load balancing virtual server.

To bind a service to a load balancing virtual server by using the command line interface

At the command prompt, type:

- bind lb vserver <name> <serviceName>
- show lb vserver [<name>]

Example

```
> bind lb vserver vserver-LB-CR service-HTTP-1
 Done
> show lb vserver Vserver-LB-CR
      Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
      State: DOWN
      Last state change was at Fri Jul  2 08:47:52 2010
      Time since last state change: 0 days, 00:42:25.610
      Effective State: DOWN
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
      No. of Bound Services :  1 (Total)       0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Vserver IP and Port insertion: OFF
      Push: DISABLED  Push VServer:
      Push Multi Clients: NO
      Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
>
```

To unbind a service from a load balancing virtual server by using the command line interface

To unbind a service, use the unbind lb vserver command instead of bind lb vserver.

To bind/unbind a service from a load balancing virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers
2. In the details pane, select the virtual server from which you want to bind/unbind the service, and then click Open.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. On the Services tab, in the Active column, select/clear the check box next to the Service Name.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Disabling the Use the Proxy Port Setting for Transparent Caching

Sep 11, 2013

If the use source IP (USIP) option is disabled on a cache service configured on the NetScaler appliance, the appliance forwards client requests to the cache service by using a NetScaler-owned subnet IP (SNIP) address or mapped IP (MIP) address as the source IP address and a random port as the source port. The randomly selected port is called the proxy port.

However, if you want to configure a fully transparent cache (a cache configuration in which the cache service receives the client's IP address and port number), you must not only enable the USIP option, either globally or on the cache service, but also disable the Use Proxy Port setting, either globally or on the cache service. Disabling the Use Proxy Port setting enables the appliance to use the client's source port as the source port when it connects to the cache service, and ensures a fully transparent cache configuration.

For more information about configuring the Use Proxy Port option globally or on a service, see "Configuring the Source Port for Server-Side Connections."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Assigning a Port Range to the NetScaler

Aug 22, 2013

Sharing of the client IP address may create a conflict that makes network devices, such as routers, cache servers, origin servers, and other NetScaler appliances, unable to determine the appliance, and therefore the client, to which the response should be sent.

A method to solve this problem is to assign a source port range to the NetScaler appliance. This allotment enables network devices to unambiguously identify the NetScaler appliance that sent the request.

To assign a source port range to a NetScaler appliance by using the command line interface

At the command prompt, type:

```
set ns param -crPortRange <startPortNumber-endPortNumber>
```

To assign a source port range to a NetScaler appliance by using the NetScaler configuration utility

1. In the navigation pane, click System, and then click Settings.
2. In the Settings group, click the Change global system settings link.
3. In the Cache Redirection Port Range group, specify the port range for the NetScaler by typing a port number for Start Port and a port number for End Port.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Load Balancing Virtual Servers to Redirect Requests to Cache

Aug 23, 2013

If a load balancing virtual server is configured to listen on a particular IP address and port combination, it takes precedence over the cache redirection virtual server for any requests destined for that address-port combination. Therefore, the cache redirection virtual server does not process those requests.

If you want to override this functionality and let the cache redirection virtual server decide whether the request should be served from the cache or not, configure the particular load balancing virtual server to be cacheable.

Such a configuration is typically used when an ISP uses a NetScaler appliance at the edge of its network and all traffic flows through the appliance.

To enable load balancing virtual servers to redirect requests to the cache by using the command line interface

At the command prompt, type:

- set lb vserver <name> [-cacheable (YES | NO)]
- show lb vserver [<name>]

Example

```
set lb vserver Vserver-LB-CR –cacheable YES
> show lb vserver vserver-LB-CR
    Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
    State: DOWN
    Last state change was at Fri Jul  2 08:47:52 2010
    Time since last state change: 0 days, 01:05:51.510
    Effective State: DOWN
    Client Idle Timeout: 180 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    Port Rewrite : DISABLED
    No. of Bound Services :  1 (Total)      0 (Active)
    Configured Method: LEASTCONNECTION
    Mode: IP
    Persistence: NONE
    Cacheable: YES  PQ: OFF SC: OFF
    Vserver IP and Port insertion: OFF
    Push: DISABLED  Push VServer:
    Push Multi Clients: NO
    Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

For transparent cache redirection, the NetScaler intercepts all traffic and evaluates every request to determine whether it is cacheable. Non-cacheable requests are sent unchanged to the origin server.

When using transparent cache redirection, you may want to turn off cache redirection for load balancing virtual servers that always direct traffic to origin servers.

To turn off caching for a load balancing virtual server by using the command line interface

To turn off caching for a load balancing virtual, use the unset lb vserver command instead of set lb vserver. Specify a value of NO value for the **-cacheable** parameter**.**

To enable or disable load balancing virtual servers to redirect requests to the cache by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server from which you want to enable/disable the caching, and then click Open.
3. On the Advanced tab, select/clear Cache Redirection check box.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Forward Proxy Redirection

Jan 01, 2018

A forward proxy is a single point of contact for a client or group of clients. In this configuration, the NetScaler appliance redirects non-cacheable requests to an origin server and redirects cacheable requests to either a forward proxy cache or a transparent cache.

When the NetScaler is configured as a forward proxy, users must modify their browsers so that the browser sends requests to the forward proxy instead of the destination servers.

A forward proxy cache redirection virtual server on the NetScaler compares the request with a policy for caching. If the request is not cacheable, the NetScaler queries a DNS load balancing virtual server for resolution of the destination, and then sends the request to the origin server. If the request is cacheable, the NetScaler forwards the request to a load balancing virtual server for the cache.

The NetScaler relies on a host domain name or IP address in the request's HOST header to determine the requested destination. If there is no HOST header in the request, the appliance inserts a HOST header based on the destination IP address in the request.

Typically, the NetScaler appliance acts as a forward proxy in an enterprise LAN. In such a configuration, the appliance resides at the edge of an enterprise LAN and intercepts client requests before they are fanned out to the WAN. Configuring the appliance in the forward proxy mode reduces traffic on the WAN.

To configure forward proxy cache redirection, first enable load balancing and cache redirection on the NetScaler. Then, configure a DNS load balancing virtual server and associated services. Also configure a load balancing virtual server and bind to it appropriate services for the cache. Configure a forward proxy cache redirection virtual server and bind the DNS and load balancing virtual servers to it. You must also configure caching policies and bind them to the cache redirection virtual server. To complete the setup, configure the client browsers to use the forward proxy.

For details on how to enable cache redirection and load balancing on the NetScaler, see "Enabling Cache Redirection and Load Balancing."

For details on how to create a load balancing virtual server, see "Creating a Load Balancing Virtual Server."

For details on how to configure services that represent the cache server, see "Configuring an HTTP Service."

For details on how to bind the service to a virtual server, see "Binding/Unbinding a Service to/from a Load Balancing Virtual Server."

For details on how to create a forward proxy cache redirection server, see "Configuring a Cache Redirection Virtual Server", and create a virtual server of type TRANSPARENT or FORWARD.

For details on binding cache redirection policies to the cache redirection virtual server, see "Configuring a Cache Redirection Policy."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating a DNS Service

Sep 10, 2013

A DNS service is a representation, on the NetScaler appliance, of a physical DNS server in the network. A DNS load balancing virtual server sends DNS requests to the DNS server in the network through such a service.

To create a DNS service by using the command line interface

At the command line, type the following commands to create a DNS service and verify the configuration :

- add service <name> <IP> <serviceType> <port>
- show service [<name>]

Example

```
add service Service-DNS-1 10.102.29.41 DNS 53
show service Service-DNS-1
 Service-DNS-1 (10.102.29.41:53) - DNS
     State: DOWN
     Last state change was at Fri Jul  2 10:14:32 2010
     Time since last state change: 0 days, 00:00:13.550
     Server Name: 10.102.29.41
     Server ID : 0   Monitor Threshold : 0
     Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
     Use Source IP: NO
     Client Keepalive(CKA): NO
     Access Down Service: NO
     TCP Buffering(TCPB): NO
     HTTP Compression(CMP): NO
     Idle timeout: Client: 120 sec   Server: 120 sec
     Client IP: DISABLED
     Cacheable: NO
     SC: OFF
     SP: OFF
     Down state flush: ENABLED

1)     Monitor Name: ping-default
           State: DOWN     Weight: 1
           Probes: 3       Failed [Total: 3 Current: 3]
           Last response: Failure - Probe timed out.
           Response Time: 2000.0 millisec
 Done
```

To add an DNS service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:

   - Service Name*—name

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Server*—IP
- Port*—port

* A required parameter

4. In the Protocol* drop down list, select a supported protocol (for example, **DNS**).
5. Click Create, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating a DNS Load Balancing Virtual Server

Aug 23, 2013

The DNS virtual server enables the forward proxy to perform DNS resolution before forwarding a client request to an origin server. The DNS load balancing virtual server is associated with the DNS service that represents the physical DNS server on the network.

To create a DNS load balancing virtual server by using the command line interface

At the command line, type the following commands to create a DNS load balancing virtual server and verify the configuration:

- add lb vserver <name> <serviceType>
- show lb vserver [<name>]

Example

```
> add lb vserver Vserver-DNS-1 DNS
 Done
> show lb vserver Vserver-DNS-1
      Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
      State: DOWN
      Last state change was at Fri Jul  2 10:32:28 2010
      Time since last state change: 0 days, 00:00:08.10
      Effective State: DOWN  ARP:DISABLED
      Client Idle Timeout: 120 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      No. of Bound Services :  0 (Total)       0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
 Done
>
```

To create a DNS load balancing virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, in the Name box, type a name for the virtual server.
4. In the Protocol* drop down list, select a supported protocol (for example, **DNS**).
5. Click Create, and then click Close. The DNS Virtual Servers pane displays the new virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding the DNS Service to the Virtual Server

Aug 22, 2013

For the DNS server to respond to DNS requests, the service representing the DNS server must be bound to the DNS virtual server.

To bind the DNS service to the load balancing virtual server:

At the command prompt, type the following commands to bind the DNS service to the load balancing virtual server and verify the configuration:

- bind lb vserver <name> <serviceName>
- show lb vserver <name>

Example

```
> bind lb vserver Vserver-DNS-1 Service-DNS-1
 Done
> show lb vserver Vserver-DNS-1
      Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
      State: DOWN
      Last state change was at Fri Jul  2 10:32:28 2010
      Time since last state change: 0 days, 00:12:16.80
      Effective State: DOWN  ARP:DISABLED
      Client Idle Timeout: 120 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      No. of Bound Services :  1 (Total)       0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE

1) Service-DNS-1 (10.102.29.41: 53) - DNS State: DOWN   Weight: 1
 Done
>
```

To unbind a DNS service from the load balancing virtual server:

Use the unbind lb vserver command instead of bind lb vserver.

To Bind/Unbind a DNS service to/from a load balancing virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers
2. In the details pane, select the virtual server to/from which you want to bind/unbind the DNS service, and then click Open.
3. On the Services tab, in the Active column, select/clear the check box next to the Service Name.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Client Web Browser to Use a Forward Proxy

Mar 30, 2012

When you configure the NetScaler appliance as forward proxy cache redirection virtual server in the network, you must configure the client Web browser to send requests to the forward proxy. Typically, when you use a forward proxy, the only route to the servers in the network is through the forward proxy.

Refer the documentation for your browser to configure the browser to use a forward proxy. Specify the IP address and port number of the forward proxy cache redirection virtual server for this configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Reverse Proxy Redirection

Jan 01, 2018

A reverse proxy resides in front of one or more Web servers and shields the origin server from client requests. Often, a reverse proxy cache is a front-end for all client requests to a server. An administrator assigns a reverse proxy cache to a specific origin server. This is unlike transparent and forward proxy caches, which cache frequently requested content for all requests to any origin server, and the choice of a server is based on the request.

Unlike a transparent proxy cache, the reverse proxy cache has its own IP address and can replace destination domains and URLs in a non-cacheable request with new destination domains and URLs.

You can deploy reverse proxy cache redirection at the origin-server side or at the edge of a network. When deployed at the origin server, the reverse proxy cache redirection virtual server is a front-end for all requests to the origin server.

In the reverse proxy mode, when the NetScaler receives a request, a cache redirection virtual server evaluates the request and forwards it to either a load balancing virtual server for the cache or a load balancing virtual server for the origin. The incoming request can be transformed by changing the host header or the host URL before they it is sent to the backend server.

To configure reverse proxy cache redirection, first enable cache redirection and load balancing. Then, configure a load balancing virtual server and services to send cacheable requests to the cache servers. Also configure a load balancing virtual server and associated services for the origin servers. Then, configure a reverse proxy cache redirection virtual server and bind relevant cache redirection policies to it. Finally, configure mapping policies and bind them to the reverse proxy cache redirection virtual server.

The mapping policies have an associated action that enables the cache redirection virtual server to forward any non-cacheable request to the load balancing virtual server for the origin.

Be sure to create the default cache server destination.

For details on how to enable cache redirection and load balancing on the NetScaler, see Enabling Cache Redirection and Load Balancing.

For details on how to create a load balancing virtual server, see Creating a Load Balancing Virtual Server.

For details on how to configure services that represent the cache server, see Configuring an HTTP Service.

For details on how to bind the service to a virtual server, see Binding/Unbinding a Service to/from a Load Balancing Virtual Server.

For details on how to create a reverse proxy cache redirection server, see Configuring a Cache Redirection Virtual Server, and create a virtual server of type REVERSE.

For details on binding built-in cache redirection policies to the cache redirection virtual server, see Binding Policies to the Cache Redirection Virtual Server.

Configuring Mapping Policies

If an incoming request is non-cacheable, the reverse-proxy cache redirection virtual server replaces the domain and URL in the request with the domain and URL of a target origin server and forwards the request to the load balancing virtual server for the origin.

A mapping policy enables the reverse proxy cache redirection virtual server to replace the destination domain and URL and forward the request to the load balancing virtual server for the origin.

A mapping policy must first translate the domain and the URL, and then pass the request on to the origin load balancing virtual

© 1999-2017 Citrix Systems, Inc. All rights reserved.

server.

A mapping policy can map a domain, a URL prefix, and a URL suffix, as follows:

- Domain mapping: You can map a domain without a prefix or suffix. The domain mapping is the default mapping for the virtual server (for example, mapping www.mycompany.com to www.myrealcompany.com).
- Prefix mapping: You can replace a specified pattern prefixed as part of the URL (for example, mapping www.mycompany.com/sports/index.html to www.mycompany.com/news/index.html).
- Suffix mapping: You can replace the file suffix in the URL (for example, mapping www.mycompany.com/sports/index.html to www.mycompany.com/sports/index.asp).

The source and the destination strings being mapped must be similar. If you specify a source domain, you must specify a destination domain, and if you specify a source suffix, you must specify a destination suffix. Similarly, if you specify an exact URL from the source, the target URL must also be an exact URL.

Once you configure mapping policies for the reverse proxy mode, you must bind them to the cache redirection virtual server.

You can use combinations of the source URL, target URL, and source and target domains to configure all three types of domain mapping.

To configure a mapping policy for reverse proxy mode by using the command line interface

At the command prompt, type the following command to add a policy map and verify the configuration:

- add policy map <mapPolicyName> -sd <string> [-su <string>] [-td <string>] [-tu <string>]
- show policy map [<mapPolicyName>]

Example

The following command maps a domain in a client request to a target domain:

```
> add policy map myMappingPolicy -sd www.mycompany.com -td www.myrealcompany.com
 Done
> show policy map myMappingPolicy
1)     Name: myMappingPolicy
       Source Domain: www.mycompany.com       Source Url:
       Target Domain: www.myrealcompany.com    Target Url:
 Done
>
```

Following is an example of mapping a URL suffix to a different URL suffix:

```
> add policy map myOtherMappingPolicy -sd www.mycompany.com -td www.myrealcompany.com -su /news.html -tu /realnews.html
 Done
> show policy map myOtherMappingPolicy
1)     Name: myOtherMappingPolicy
       Source Domain: www.mycompany.com       Source Url: /news.html
       Target Domain: www.myrealcompany.com    Target Url: /realnews.html
 Done
>
```

To configure a mapping policy for reverse proxy mode by using the configuration

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

utility

1. Navigate to Traffic Management > Cache Redirection > Map Policies.
2. In the details pane, click Add.
3. In the Create Map Policy dialog box, specify values for the following parameters as shown:

   - Name*- mapPolicyName
   - Source Domain*-sd
   - Target Domain*-td
   - Source URL-su
   - Target URL-tu

   * A required parameter

4. Click Create, and then click Close. The Map pane displays the new mapping policy.

To bind the mapping policy to the cache redirection virtual server by using the command line interface

At the command prompt, type the following commands to bind the mapping policy to the cache redirection virtual server and verify the configuration:

- bind cr vserver <name> -policyName <string> [<targetVserver>]
- show cr vserver <name>

Example

```
> bind cr vserver Vserver-CRD-3 -policyName myMappingPolicy Vserver-LB-CR
 Done
> show cr vserver Vserver-CRD-3
      Vserver-CRD-3 (10.102.29.50:88) - HTTP  Type: CONTENT
      State: UP
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default: Vserver-LB-CR  Content Precedence: RULE       Cache: REVERSE
      On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF

1)     Policy:        Target: Vserver-LB-CR   Priority: 0     Hits: 0
1)     Map: myMappingPolicy Target: Vserver-LB-CR
 Done
>
```

To bind the mapping policy to the cache redirection virtual server by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server from which you want to bind the mapping policy, and then click Open.
3. In the Configure Virtual Server(Cache Redirection), on the Policies tab, select Map, and then click Insert Policy.
4. In the Policy Name column, select the policy from drop down list.
5. In the Target column, click the down arrow, and then select the vserver from drop down list.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

6. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Selective Cache Redirection

Jan 01, 2018

Selective cache redirection sends requests for particular types of content, for example, images, to one cache server or group of cache servers and sends other types of content to a different cache server or group of cache servers. You can configure advanced cache redirection in transparent, reverse proxy, or forward proxy modes.

In selective cache redirection, the NetScaler appliance intercepts a client request and forwards non-cacheable requests to the original destination in the client request. For cacheable requests, the appliance sends the requests to the destination cache server that can serve content of a specific content type.

Selective cache redirection involves configuring content switching policies in addition to cache redirection policies. The NetScaler first evaluates the cache redirection policies that are bound to the cache redirection virtual server. If a request matches a cache redirection policy, the cache redirection virtual server sends the request to the origin server or a load balancing virtual server for the origin. If no cache redirection policies match the request, the NetScaler evaluates the content switching policies bound to the cache redirection virtual server. If a content switching policy matches the request, the cache redirection virtual server redirects the request to a load balancing virtual server for the cache.

To configure selective cache redirection, first enable cache redirection, load balancing, and content switching on the NetScaler appliance. Then, configure a load balancing virtual server for the cache and an associated HTTP service. After this, configure a cache redirection virtual server and bind both the cache redirection and content switching policies to it. Once you have bound the policies, you can configure the virtual server to give precedence to either rule based or URL based content-switching policies.

When configured for transparent mode cache redirection in an edge deployment topology, the NetScaler sends all cacheable HTTP traffic to a transparent cache farm. Clients access the Internet through the NetScaler, which is configured as a Layer 4 switch that receives traffic on port 80.

The NetScaler can direct requests for images (for example, .gif and .jpg files) to one server in the transparent cache farm, and all other requests for static content to other servers in the farm. For this configuration, you configure content switching policies to send images to the image cache and send all other cacheable content to a default cache.

Note: The configuration described here is for transparent selective cache redirection. Therefore, it does not require a load balancing virtual server for the origin, as would a reverse proxy configuration.
To configure this type of selective cache redirection, first enable cache redirection, load balancing, and content switching. Then, configure a load balancing virtual server for the cache and configure an associated HTTP service. Then, configure a cache redirection virtual server and create and bind both cache redirection and content switching policies to this virtual server.

For details on how to enable cache redirection and load balancing on the NetScaler, see Cache Redirection Configurations.

.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Content Switching

Nov 08, 2013

To configure selective cache redirection, after you enable both the load balancing and cache redirection features on the NetScaler, you must enable content switching.

To enable content switching by using the command line interface

At the command prompt, type:

- enable ns feature CS
- show ns feature

Example

```
> enable ns feature cs
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging            WL               ON
 2)   Surge Protection       SP               ON
 3)   Load Balancing         LB               ON
 4)   Content Switching      CS               ON
 5)   Cache Redirection      CR               ON
      ...
      ...
      ...
 23)  HTML Injection         HTMLInjection    ON
 24)  NetScaler Push         push             OFF
 Done
```

To enable cache redirection and load balancing by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure basic features.
3. In Configure Basic Features dialog box, select the check box next to the Content Switching, and then click OK.
4. In Enable/Disable Feature(s)? dialog box, click Yes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Load Balancing Virtual Server for the Cache

Jan 01, 2018

Create a load balancing virtual server and an HTTP service for each type of cache server that will be used. For example, if you want to serve JPEG files from one cache server and GIF files from another cache server, and use a third cache server for the rest of the content, create an HTTP service and virtual server for each of the three types of cache servers. Then bind each service to its respective virtual server.

For details on how to create a load balancing virtual server, see "Creating a Virtual Server."

For details on how to configure services that represent the cache server, see Configuring an HTTP Service.

For details on how to bind the service to a virtual server, see Binding/Unbinding a Service to/from a Load Balancing Virtual Server.

For details on how to create a transparent proxy cache redirection server, see Configuring a Cache Redirection Virtual Server, and create a virtual server of type TRANSPARENT.

For details on binding built-in cache redirection policies to the cache redirection virtual server, see Binding Policies to the Cache Redirection Virtual Server.

Configuring a Cache Redirection Policy for a Specific Type of Content

To identify requests that contain a .gif or .jpeg extension as cacheable, you configure a cache redirection policy and bind it to the cache redirection virtual server.

Note: If a request matches a policy, the NetScaler appliance forwards it to the origin server. As a result, in the following procedure, you configure policies to match requests that do *not* have ".gif" or ".jpeg" extensions.
To configure cache redirection for a specific type of content, configure a policy that uses a simple expression, as described in Configuring a Cache Redirection Policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Policies for Content Switching

Jan 01, 2018

You must create a content switching policy to identify specific types of content to be cached in one cache server or farm and identify other types of content to serve from another cache server or farm. For example, you can configure a policy to determine the location for image files with .gif and .jpeg extensions.

After defining the content switching policy, you bind it to a cache redirection virtual server and specify a load balancing virtual server. Requests that match the policy are forwarded to the named load balancing virtual server. Requests that do not match the content switching policy are forwarded to the default load balancing virtual server for the cache.

For more details about the content switching feature and configuring content switching policies, see Content Switching.

You must first create the content switching policy and then bind it to the cache redirection virtual server.

To create a content switching policy by using the command line interface

At the command line, type:

- add cs policy <policyName> [-url <string> | -rule <expression>]
- show cs policy [<policyName>]

Examples

```
> add cs policy Policy-CS-JPEG -rule "REQ.HTTP.URL == '/*.jpeg'"
 Done
> show cs policy Policy-CS-JPEG
        Rule: REQ.HTTP.URL == '/*.jpeg'        Policy: Policy-CS-JPEG
        Hits: 0
 Done
>

> add cs policy Policy-CS-GIF -rule "REQ.HTTP.URL == '/ *.gif'"
 Done
> show cs policy Policy-CS-GIF
        Rule: REQ.HTTP.URL == '/ *.gif'        Policy: Policy-CS-GIF
        Hits: 0
 Done
>

> add cs policy Policy-CS-JPEG-URL -url /*.jpg
 Done
> show cs policy Policy-CS-JPEG-URL
        URL: /*.jpg     Policy: Policy-CS-JPEG-URL
        Hits: 0
 Done
>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> add cs policy Policy-CS-GIF-URL -url /*.gif
 Done
> show cs policy Policy-CS-GIF-URL
        URL: /*.gif     Policy: Policy-CS-GIF-URL
        Hits: 0
 Done
>
```

To create a URL-based content switching policy by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Policies.
2. In the details pane, click Add.
3. In the Create Content Switching Policy dialog box, in the Name text box, type a name for the policy.
4. Select the URL radio button.
5. In the Value text box, type the string value (for example, **/sports**).
6. Click Create and click Close. The policy you created appears in the Content Switching Policies page.

To create a rule-based content switching policy by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Policies.
2. In the details pane, click Add.
3. In the Create Content Switching Policy dialog box, in the Name text box, type a name for the policy.
4. Select the Expression radio button, and then click Configure.
5. In the Create Expression dialog box, choose the expression syntax that you want to use.
   - If you want to use default syntax, accept the default and proceed to the next step.
   - If you want to use classic syntax, click Switch to Classic Syntax.

   The Expression portion of the dialog box changes to match your choice. The default syntax Expression view has fewer elements than does the classic syntax Expression view. In the default syntax Expression view, instead of a preview window, a button provides access to an expression evaluator. The evaluator evaluates the expression you entered, to verify that it is valid, and displays an analysis of the expression's effect.

6. Enter your policy expressions.
   - If you are using classic syntax and need further instructions, see "Configuring Classic Policies and Expressions."
   - If you are using the default syntax and need further instructions, see "Configuring Default Syntax Expressions: Getting Started."
7. Click Create and click Close. The policy you created appears in the Content Switching Policies pane.

To bind the content switching policy to a cache redirection virtual server by using the command line interface

At the command prompt, type the following commands to bind the content switching policy to a cache redirection virtual server and verify the configuration:

- bind cs vserver <name> <targetVserver> [-policyName <string>]
- show cs vserver [<name>]

Example

```
> bind cs vserver Vserver-CR-1 lbcachejpeg -policyName Policy-CS-JPEG
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 Done
> bind cs vserver Vserver-CR-1 lbcachegif -policyName Policy-CS-GIF
 Done
> show cs vserver Vserver-CR-1
      Vserver-CR-1 (10.102.29.60:80) - HTTP   Type: CONTENT
      State: UP
      Last state change was at Fri Jul  2 12:53:45 2010
      Time since last state change: 0 days, 00:00:58.920
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
      State Update: DISABLED
      Default:       Content Precedence: RULE
      Cacheable: YES
      Vserver IP and Port insertion: OFF
      Case Sensitivity: ON
      Push: DISABLED  Push VServer:
      Push Label Rule: none

1)     Policy: Policy-CS-JPEG  Target: lbcachejpeg     Priority: 0     Hits: 0
2)     Policy: Policy-CS-GIF   Target: lbcachegif      Priority: 0     Hits: 0
 Done
>
```

To bind the content switching policy to a cache redirection virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, select the virtual server for which you want to bind the policy (for example, **Vserver-CS-1**), and then click Open.
3. In the Configure Virtual Server (Content Switching) dialog box, on the Policies tab, click CSW, and then click Insert Policy.
4. In the Policy Name column, select the policy that you want to configure for the content switching virtual server.
5. In the Target column, click the green arrow, and select the target load balancing virtual server from the list.
6. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Precedence for Policy Evaluation

Aug 23, 2013

You can configure a content switching policy based on either a rule, which is a generic configuration to accommodate various content types, or a URL, which is more specific and defines exactly the type of content that has to be sent to a particular cache server. Essentially, the same content can be defined by either a rule based policy or a URL based policy.

Once you bind content switching policies of either type to a cache redirection virtual server, you can configure the virtual server to give precedence to either rule based or URL based policies. This will, in turn, decide which servers the particular requests are directed to.

To configure precedence for policy evaluation, use the precedence parameter, which specifies the type of policy (URL or RULE) that takes precedence on the content redirection virtual server.

Possible values: RULE, URL

Default value: RULE

To configure precedence for policy evaluation by using the command line interface

At the command prompt, type the following commands to configure precedence for policy evaluation and verify the configuration:

- set cr vserver <name> [-precedence (RULE | URL)]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -precedence URL
 Done
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: URL Cache: TRANSPARENT
      On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
>
```

To configure precedence for policy evaluation by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure precedence, (for example, **Vserver-CS-1**), and then click Open.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Configure Virtual Server (Content Switching) dialog box, on the Advanced tab, next to Precedence, click Rule or URL, and then click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Administering a Cache Redirection Virtual Server

Mar 30, 2012

To administer a cache redirection virtual server, you need to view cache redirection statistics. You might need to enable or disable cache redirection servers, or direct policy hits to the cache instead of the origin. Administrative tasks also include backing up a cache redirection virtual server and managing client connections.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing Cache Redirection Virtual Server Statistics

Aug 23, 2013

You can view properties of a cache redirection virtual server and statistics on the traffic that has passed through a cache redirection virtual server. You can also view the cache redirection virtual servers and policies that you have bound to load balancing virtual servers.

To view statistics for a specific cache redirection virtual servers, use the name parameter to specify the name of the virtual server for which statistics will be displayed. Otherwise, statistics for all cache redirection virtual servers are displayed. Maximum Length: 127

To view statistics for a cache redirection virtual server by using the command line interface

At the command prompt, type:

stat cr vserver [<name>]

Example

```
> stat cr vserver Vserver-CRD-1

Vserver Summary
                    IP  port    Protocol      State
Vser...CRD-1        0.0.0.0   80       HTTP         UP

VServer Stats:
                              Rate (/s)            Total
Requests                              0                 0
Responses                              0                 0
Request bytes                          0                 0
Response bytes                          0                 0

 Done
>
```

To view statistics for a cache redirection virtual server by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers
2. In the details pane, select the virtual server for which you want to view statistics, (for example, **Vserver-CRD-1**), and then click Statistics.

Omit the server name to display basic statistics for all cache redirection virtual servers. Include the server name to display detailed statistics for that virtual server, including number and size of requests and responses that pass through the virtual server

To view the statistics of a cache redirection virtual server by using the monitoring and dashboard utilities

1. To view the statistics by using the monitoring utilities, click the Monitoring tab.
2. In the Select Group drop-down menu, choose CR Virtual Servers. A list of cache redirection virtual servers appears.
3. To view the statistics by using the dashboard utilities, click the Dashboard tab.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Click Applet Client or Web Start Client next to Statistical Utility.
5. In the Select Group drop-down menu, choose CR Virtual Servers. The dashboard displays summary statistics for the cache redirection virtual servers.
6. To see a chart of virtual server activity, click Chart. A graphical representation of the virtual server statistics appears.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling or Disabling a Cache Redirection Virtual Server

Aug 23, 2013

When you create a cache redirection virtual server, it is enabled by default. If you disable a cache redirection virtual server, its state changes to OUT OF SERVICE and it stops redirecting cacheable client requests. However, the NetScaler appliance continues to respond to ARP and ping requests for the IP address of this virtual server.

To Enable or Disable a cache redirection virtual servers by using the command line interface

At the command line, type one of the following commands:

- enable cr vserver <name>
- show cr vserver <name>
- disable cr vserver <name>
- show cr vserver <name>

Examples

```
> enable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: URL Cache: TRANSPARENT
      On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
>

> disable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: OUT OF SERVICE  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: URL Cache: TRANSPARENT
      On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
>
```

To Enable or Disable a cache redirection virtual servers by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the navigation pane, expand Cache Redirection, and then click Virtual Servers.
3. In the details pane, select the virtual server that you want to enable or disable, (for example, **Vserver-CRD-1**), and then click Statistics.
4. In the Proceed dialog box, click Yes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Directing Policy Hits to the Cache Instead of the Origin

Aug 23, 2013

By default, when a request matches a policy, the NetScaler appliance forwards the request either to the origin server directly, or to a load balancing virtual server for the origin, depending on how you have configured cache redirection.

You can change the default behavior so that when a request matches a policy, the request is forwarded to a load balancing virtual server for the cache.

To change the destination for a policy hit to the origin or the cache, use the onPolicyMatch parameter, which specifies where to send requests that match the cache redirection policy.

The valid options are:

1. CACHE - Directs all matching requests to the cache.
2. ORIGIN - Directs all matching requests to the origin server.

Note: For this option to work, you must select the cacheredirection type as POLICY.
Possible values: CACHE, ORIGIN

Default value: ORIGIN

To change the destination for a policy hit to the origin or the cache by using the command line interface

At the command prompt, type the following commands to change the destination for a policy hit and verify the configuration:

- set cr vserver <name> [-onPolicyMatch (ORIGIN | CACHE)]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -onPolicyMatch CACHE
 Done
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: URL Cache: TRANSPARENT
      On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To change the destination for a policy hit to the origin or the cache by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to change the destination for a policy hit, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, click Advanced tab.
4. Select CACHE or ORIGIN from the Redirect To drop-down list.
5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Backing Up a Cache Redirection Virtual Server

Aug 22, 2013

Cache redirection can fail if the primary virtual server fails, or if it is unable to handle excessive traffic. You can specify a backup virtual server to take over the processing of traffic when the primary virtual server fails.

To specify a backup cache redirection virtual server, use the backupVServer parameter, which specifies Backup Virtual Server. Maximum Length: 127

To specify a backup cache redirection virtual server by using the command line interface

At the command prompt, type the following commands to specify a backup cache redirection virtual server and verify the configuration:

- set cr vserver <name> [-backupVServer <string>]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -backupVServer Vserver-CRD-2
 Done
> show cr vserver Vserver-CRD-1
      Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
      State: UP  ARP:DISABLED
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Default:       Content Precedence: URL Cache: TRANSPARENT
      On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
      Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF
      Backup: Vserver-CRD-2

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
```

To specify a backup cache redirection virtual server by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > irtual Servers.
2. In the details pane, select the virtual server for which you want to change the destination for a policy hit, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. In the Backup Virtual Server drop-down list, select the virtual server.
5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Client Connections for a Virtual Server

Jan 01, 2018

You can configure timeouts on a cache redirection virtual server so that client connections are not kept open indefinitely. You can also insert Via headers in requests. To possibly reduce network congestion, you can reuse open TCP connections. You can enable or disable delayed cleanup of cache redirection virtual server connections.

You can configure the NetScaler to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP VSERVER RESPONSE.

The following settings can be made on a virtual server:

- When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, NetScaler always responds.
- When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds even if one virtual server is UP.
- When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds even if one virtual server set to ACTIVE is UP.

This section includes the following topics:

- Configuring Client Timeout
- Inserting Via Headers in the Requests
- Reusing TCP Connections
- Configuring Delayed Connection Cleanup

Configuring Client Timeout

You can specify expiration of client requests by setting a timeout value for the cache redirection virtual server. The timeout value is the number of seconds for which the cache redirection virtual server waits to receive a response for the client request.

To configure a time-out value, use the cltTimeout parameter, which specifies the time, in seconds, after which the NetScaler appliance closes any idle client connections. The default value is 180sec for HTTP/SSL-based services and 9000sec for TCP-based services.

To configure client timeout by using the command line interface

At the command prompt, type the following commands to configure client timeout and verify the configuration:

- set cr vserver <name> [-cltTimeout <secs>]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -cltTimeout 6000
 Done
> show cr vserver Vserver-CRD-1
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
    State: UP  ARP:DISABLED
    Client Idle Timeout: 6000 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    Default:       Content Precedence: URL Cache: TRANSPARENT
    On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
    Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF
    Backup: Vserver-CRD-2

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
```

To configure client timeout by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. In the Client Time-out(secs) text box, enter the time-out value in seconds.
5. Click OK.

Inserting Via Headers in the Requests

A Via header lists the protocols and recipients between the start and end points for a request or a response and informs the server of proxies through which the request was sent. You can configure the cache redirection virtual server to insert a Via header in each HTTP request. The via parameter is enabled by default when you create a cache redirection virtual server.

To enable or disable Via-header insertion in client requests, use the via parameter, which specifies the state of the system in inserting a Via header in the HTTP requests.

Possible values: ON, OFF

Default value: ON

To enable or disable Via-header insertion in client requests by using the command line interface

At the command prompt, type:

- set cr vserver <name> [-via (ON | OFF)]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -via ON
 Done
> show cr vserver Vserver-CRD-1
    Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    State: UP  ARP:DISABLED
    Client Idle Timeout: 6000 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    Default:       Content Precedence: URL Cache: TRANSPARENT
    On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
    Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF
    Backup: Vserver-CRD-2

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
>
```

To enable or disable Via-header insertion in client requests by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. Select the Via check box.
5. Click OK.

Reusing TCP Connections

You can configure the NetScaler appliance to reuse TCP connections to the cache and origin servers across client connections. This can improve performance by saving the time required to establish a session between the server and the NetScaler. The reuse option is enabled by default when you create a cache redirection virtual server.

To enable or disable the reuse of TCP connections, use the reuse parameter, which specifies the state of reuse of TCP connections to the cache or origin servers across client connections.

Possible values: ON, OFF

Default value: ON

To enable or disable the reuse of TCP connections by using the command line interface

At the command prompt, type:

- set cr vserver <name> [-reuse (ON|OFF)]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -reuse ON
 Done
> show cr vserver Vserver-CRD-1
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
    State: UP  ARP:DISABLED
    Client Idle Timeout: 6000 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    Default:       Content Precedence: URL Cache: TRANSPARENT
    On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
    Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF
    Backup: Vserver-CRD-2

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
 Done
```

To enable or disable the reuse of TCP connections by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. Select the Reuse check box.
5. Click OK.

Configuring Delayed Connection Cleanup

The down state flush option performs delayed cleanup of connections on a cache redirection virtual server. The down state flush option is enabled by default when you create a cache redirection virtual server.

To enable or disable the down state flush option, set the downStateFlush parameter.

Possible values: ENABLED, DISABLED

Default value: ENABLED

To enable of disable the down state flush option by using the command line interface

At the command prompt, type the following commands to configure delayed connection clean up and verify the configuration:

- set cr vserver <name> [-downStateFlush (ENABLED | DISABLED)]
- show cr vserver <name>

Example

```
> set cr vserver Vserver-CRD-1 -downStateFlush ENABLED
 Done
> show cr vserver Vserver-CRD-1
    Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
    State: UP  ARP:DISABLED
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Client Idle Timeout: 6000 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Default:      Content Precedence: URL Cache: TRANSPARENT
On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
Redirect: POLICY       Reuse: ON       Via: ON ARP: OFF
Backup: Vserver-CRD-2

1)     Cache bypass  Policy: bypass-cache-control
2)     Cache bypass  Policy: Policy-CRD
Done
```

To enable or disable the reuse of TCP connections by using the configuration utility

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, click Advanced tab.
4. Select the Down state flush check box.
5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

N-Tier Cache Redirection

Mar 30, 2012

To efficiently handle large amounts of cached data, typically several gigabytes per second, an Internet Service Provider (ISP) deploys several dedicated cache servers. The cache redirection feature of the NetScaler appliance can help load balance the cache servers, but a single appliance or a couple of appliances might not efficiently handle the large volume of traffic.

You can solve the problem by deploying the NetScaler appliances in two tiers (layers), where the appliances in the upper tier load balance those in the lower tier and the appliances in the lower tier load balance the cache servers. This arrangement is called *n-tier cache redirection*.

For purposes such as auditing and security, an ISP has to track client details such as the IP address, information provided, and the time of the interaction. Therefore, client connections through a NetScaler appliance have to be fully transparent. However, if you configure transparent cache redirection, with the NetScaler appliances deployed in parallel, the IP address of the client has to be shared among all the appliances. Sharing of the client IP address creates a conflict that makes network devices, such as routers, cache servers, origin servers, and other NetScaler appliances, unable to determine the appliance, and therefore the client, to which the response should be sent.

How N-tier Cache Redirection Is Implemented

To solve the problem, NetScaler n-tier cache redirection splits the source port range among the appliances in the lower tier and includes the client IP address in the request sent to the cache servers. The upper-tier NetScaler appliances are configured to do sessionless load balancing in order to avoid unnecessary load on the appliances.

When the lower-tier NetScaler appliance communicates with a cache server, it uses a mapped IP address (MIP) to represent the source IP address. Therefore, the cache server can identify the NetScaler from which it received the request and send the response to the same NetScaler.

The lower-tier NetScaler appliance inserts the client IP address into the header of the request sent to the cache server. The client IP in the header helps the NetScaler to determine the client to which the packet should be forwarded when it receives the response from a cache server, or the origin server in case of a cache miss. The origin server determines the response to be sent according to the client IP inserted in the request header.

The origin server sends the response to an upper-tier NetScaler, including the source port number from which the origin server received the request. The entire source port range, 1024 to 65535, is distributed among the lower-tier NetScaler appliances. Each lower-tier appliance is exclusively assigned a group of addresses within the range. This allotment enables the upper-tier appliance to unambiguously identify the lower-tier NetScaler appliance that sent the request to the origin server. The upper-tier appliance can therefore forward the response to the correct lower-tier appliance.

The upper-tier NetScaler appliances are configured to do policy-based routing, and the routing policies are defined to determine the IP address of the destination NetScaler from the source port range.

Setup Necessary for Configuring N-Tier CRD

The following setup is necessary for the functioning of n-tier cache redirection:

For each upper-tier NetScaler appliance:

- Enable Layer 3 mode.
- Define policies for policy-based routes (PBRs) so that traffic is forwarded according to the range of the destination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

port.

- Configure a load balancing virtual server.
- Configure the virtual server to listen to all the traffic coming from the client. Set the Service Type/Protocol to be ANY and IP Address as asterisk (*).
- Enable sessionless load balancing with MAC-based redirection mode to avoid unnecessary load on the upper-tier NetScaler appliances.
- Make sure that the Use Proxy Port option is enabled.
- Create a service for each lower-tier NetScaler and bind all the services to the virtual server.

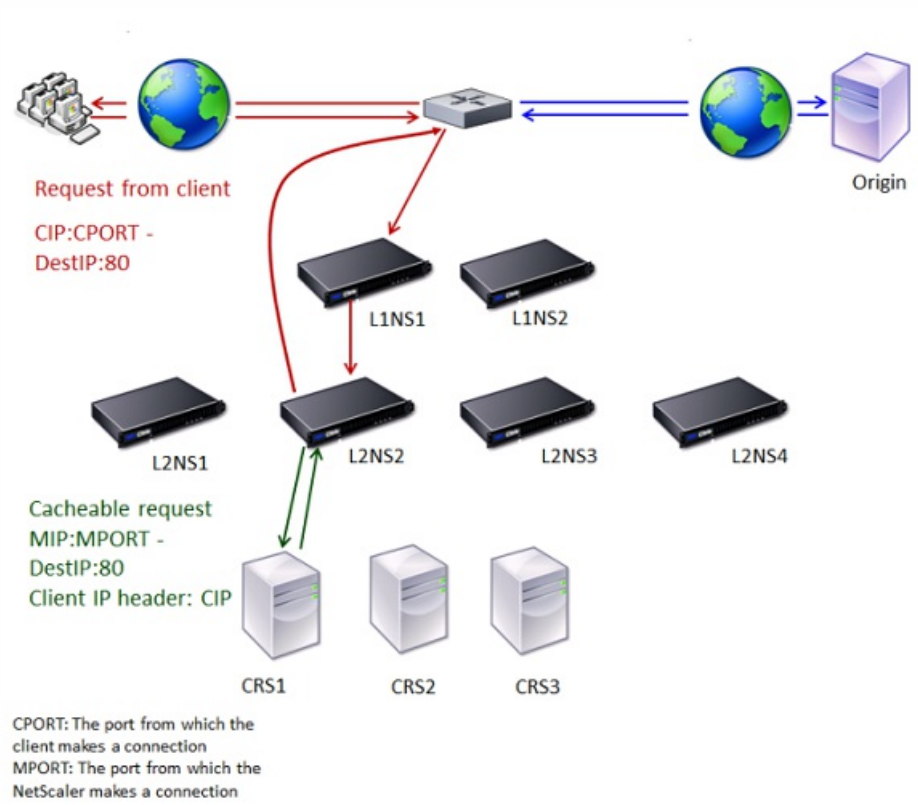
For each lower-tier NetScaler appliance,

- Configure the cache redirection port range on the NetScaler. Assign an exclusive range to each lower-tier NetScaler.
- Configure a load balancing virtual server and enable MAC-based redirection.
- Create a service for each cache server that is to be load balanced by this NetScaler. When creating the service, enable insertion of client IP in the header. Then, bind all the services to the load balancing virtual server.
- Configure a transparent mode cache redirection virtual server with the following settings:
  - Enable the Origin USIP option.
  - Add a source IP expression to include the client IP in the header.
  - Enable the Use Port Range option.

How N-Tier Cache Redirection Works During a Cache Hit

The following figure shows how cache redirection works when a client request is cacheable and the response is sent from a cache server.

Figure 1. Cache Redirection in Case of a Cache Hit



Two NetScaler appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four NetScaler appliances, L2NS1, L2NS2,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

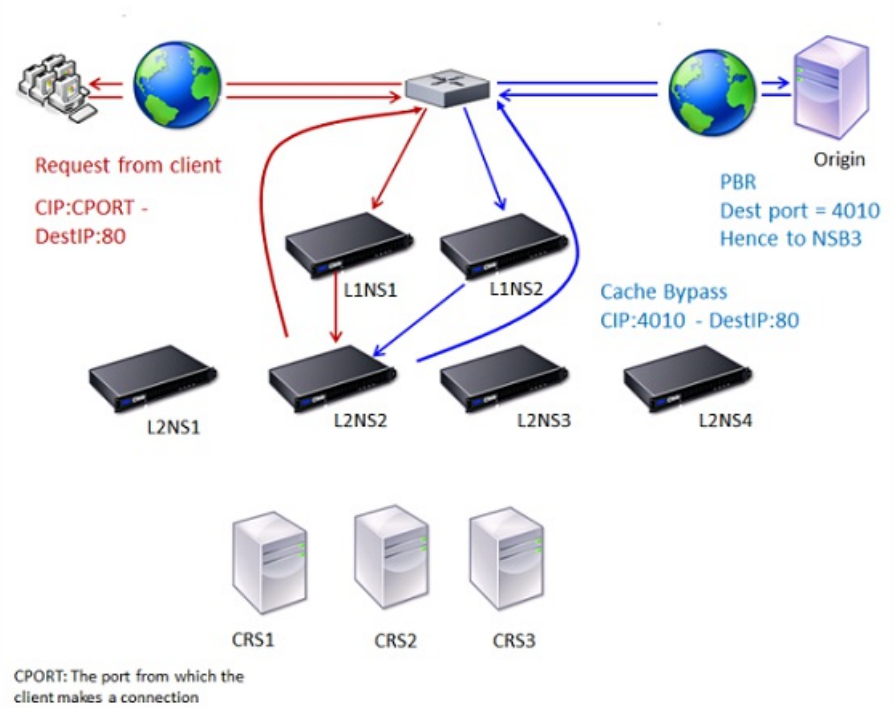
Traffic Flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. L2NS2 load balances the request to the cache server CRS1, and the request is cacheable. L2NS2 includes the client IP in the request header.
4. CRS1 sends the response to L2NS2 because L2NS2 used its MIP as the source IP address when connecting to CRS1.
5. With the help of the client IP address in the request header, L2NS2 identifies the client from which the request came. L2NS2 directly sends the response to the router, avoiding unnecessary load on the NetScaler in the upper tier.
6. The router forwards the response to Client A.

How N-Tier Cache Redirection Works During a Cache Bypass

The following figure shows how cache redirection works when a client request is sent to an origin server for a response.

Figure 2. Cache Redirection in Case of a Cache Bypass



Two NetScaler appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four NetScaler appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

Traffic Flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. The request is uncacheable (cache bypass). Therefore, L2NS2 sends the request to the origin server through the router.
4. The origin server sends the response to an upper-tier NetScaler, L1NS2.

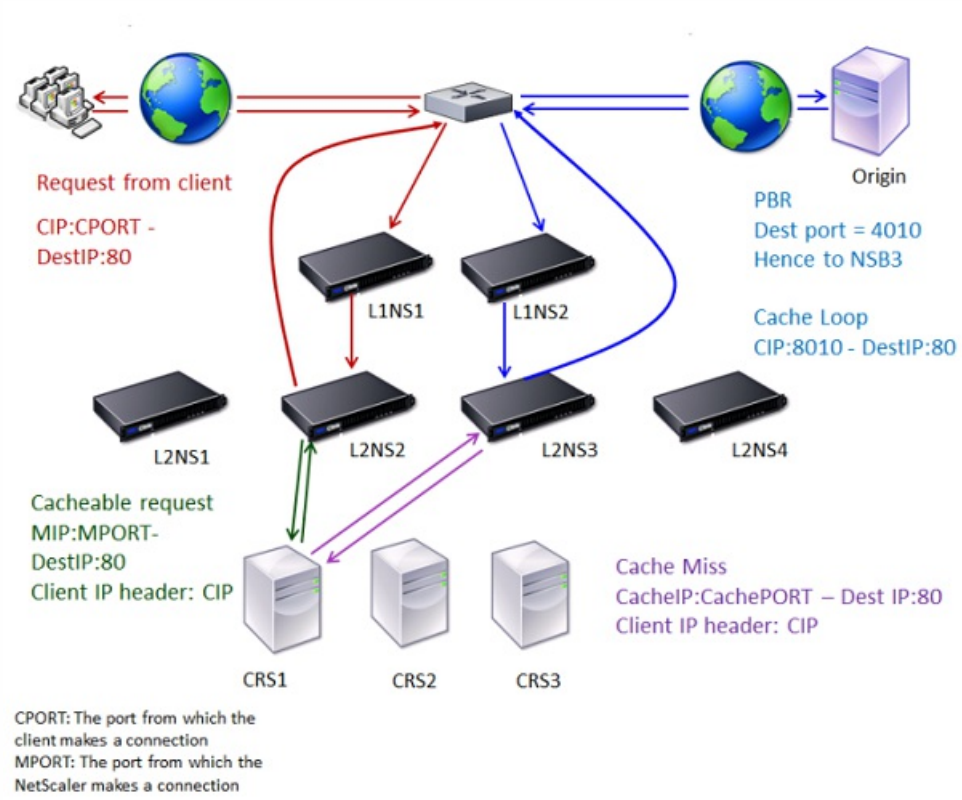
 © 1999-2017 Citrix Systems, Inc. All rights reserved.

5. According to the PBR policies, L1NS2 forwards the traffic to the appropriate NetScaler in the lower tier, L2NS2.
6. L2NS2 uses the client IP address in the request header to identify the client from which the request came and sends the response directly to the router, avoiding unnecessary load on the NetScaler in the upper tier.
7. The router forwards the response to Client A.

How N-Tier Cache Redirection Works During a Cache Miss

The following figure shows how cache redirection works when a client request is not cached.

Figure 3. Cache Redirection in Case of a Cache Miss



Two NetScaler appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four NetScaler appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

Traffic Flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. L2NS2 load balances the request to the cache server CRS1 because the request is cacheable.
4. CRS1 does not have the response (cache miss). CRS1 forwards the request to the origin server through the NetScaler in the lower tier. L2NS3 intercepts the traffic.
5. L2NS3 takes the client IP from the header and forwards the request to the origin server. The source port included in the packet is the L2NS3 port from which the request is sent to the origin server.
6. The origin server sends the response to an upper-tier NetScaler, L1NS2.
7. According to the PBR policies, L1NS2 forwards the traffic to the appropriate NetScaler in the lower tier, L2NS3.
8. L2NS3 forwards the response to the router.
9. The router forwards the response to Client A.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Upper-Tier NetScaler Appliances

Oct 31, 2013

Configure each of the upper-tier NetScaler appliances as follows.

To configure an upper-tier appliance for n-tier cache redirection by using the command line interface

At the command prompt, type the following commands:

- add service <name>@ <serviceIP> <serviceType> <port>
  Run this command for each service to be added.

- add lb vserver <name>@ ANY * <port> -persistenceType <persistenceMethod> -lbMethod <lbMethod> -m MAC -sessionless ENABLED -cltTimeout <client_Timeout_Value>
- bind lb vserver <name>@ <serviceName>
  Run this command for each service to be bound.

- enable ns mode l3
- add ns pbr <name> <action> -srcPort <sourcePortNumber> -destPort <startPortNumber-endPortNumber> -nexthop <serviceIpAddress> -protocol TCP
- apply ns pbrs

  Run this command after adding all the necessary PBRs.

To configure an upper-tier appliance for n-tier cache redirection by using the configuration utility

1. Enable L3 mode:
   1. In the navigation pane, click System, and then click Settings.
   2. In the Settings group, click the Configure modes link.
   3. Select the Layer 3 Mode (IP Forwarding) check box.
   4. Click OK.
2. Configure policy-based routing (PBR):
   1. Navigate to System > Network > PBRs.
   1. In the Policy-Based Routing (PBRs) pane, click Add.
   2. Type a name for the PBR.
   3. Select the action as Allow.
   4. In the Next Hop box, type the IP address of the service, which represents a lower-tier NetScaler.
   5. Select TCP from the Protocol drop-down list.
   6. Type the source port and the range of the destination port corresponding to the lower-tier NetScaler being added.
   7. Click Create.
   8. In the details pane, select the PBR and click Apply.
   9. Repeat Step (i) to Step (vii) for each lower-tier NetScaler.
3. Create a service for each lower-tier NetScaler:
   1. Navigate to Traffic Management > Load Balancing > Services.
   1. In the details pane, click Add.
   2. Specify the name, protocol, IP address, and port. The protocol should be ANY.
   3. Click Create.
4. Configure a load balancing virtual server:
   1. Navigate to Traffic Management > Load Balancing > Virtual Servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. In the details pane, click Add.
2. Specify the name, protocol, IP address, and port. The protocol should be ANY and the IP address should be *.
3. In the Services tab, select the services that represent the lower-tier NetScaler appliances.
4. In the Advanced tab, select the Redirection Mode as MAC Based and select the Sessionless check box.
5. Click Create.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Lower-Tier NetScaler Appliances

Oct 31, 2013

Configure each of the lower-tier NetScaler appliances as follows.

To configure a lower-tier appliance for n-tier cache redirection by using the command line interface

At the command prompt, type the following commands:

- add service <name>@ <cacheServiceIP> <serviceType> <port> -cip ENABLED "ClientIP" –cachetype transparent
  Repeat for each cache server.

- add lb vserver <name>@ <serviceType> -m MAC
- bind lb vserver <name>@ <cacheServiceName>
  Repeat for each cache server.

- add cr vserver <name> <serviceType> * <port> -srcIPExpr "HTTP.REQ.HEADER(\"ClientIP\")" -originusip ON –usePortRange ON
- set ns param-crPortRange <startPortNumber-endPortNumber>

To configure a lower-tier appliance for n-tier cache redirection by using the configuration utility

1. Create a service for each cache server. To create a service:
   1. Navigate to Traffic Management > Load Balancing > Services.
   1. In the details pane, click Add, and specify the name and protocol. Clear the Directly Addressable check box.
   2. In the Advanced tab, select the Override Global check box and the Client IP check box, and then in the Header box, type ClientIP.
   3. In the Cache Type box, select Transparent Cache.
   4. Click Create.
2. Configure a load balancing virtual server:
   1. Navigate to Traffic Management > Load Balancing > Virtual Services.
   1. In the details pane, click Add and specify the name, protocol, IP address, and port. The IP address should be an asterisk (*).
   2. In the Services tab, select the services that represent the cache servers.
   3. In the Advanced tab, for Redirection Mode, select MAC Based.
   4. Click Create.
3. Configure a cache redirection virtual server:
   1. Navigate to Traffic Management > Load Balancing > Virtual Services.
   1. In the details pane, click Add and specify the name, protocol, IP address, and port. The IP address should be *.
   2. For Cache Type, select Transparent.
   3. On the Advanced tab, in the Cache Server box, select the new load balancing virtual server and check the Origin USIP and Use Port Range check boxes. In the Source IP Expression box, type HTTP.REQ.HEADER("ClientIP").
   4. Click Create.
4. Assign a source port range for the NetScaler:
   1. In the navigation pane, click System, and then click Settings.
   2. In the Settings group, click the Change global system settings link.
   3. In the Cache Redirection Port Range group, specify the port range for the NetScaler by typing a port number for Start Port and a port number for End Port.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Content Switching

Aug 22, 2016

In today's complex Web sites, you may want to present different content to different users. For example, you may want to allow users from the IP range of a customer or partner to have access to a special Web portal. You may want to present content relevant to a specific geographical area to users from that area. You may want to present content in different languages to the speakers of those languages. You may want to present content tailored to specific devices, such as smartphones, to those who use the devices. The Citrix NetScaler content switching feature enables the NetScaler appliance to distribute client requests across multiple servers on the basis of specific content that you wish to present to those users.

To configure content switching, first create a basic content switching setup, and then customize it to meet your needs. This entails enabling the content switching feature, setting up load balancing for the server or servers that host each version of the content that is being switched, creating a content switching virtual server, creating policies to choose which requests are directed to which load balancing virtual server, and binding the policies to the content switching virtual server. You can then customize the setup to meet your needs by setting precedence for your policies, protecting your setup by configuring a backup virtual server, and improving the performance of your setup by redirecting requests to a cache.

How Content Switching Works

Content Switching enables the NetScaler appliance to direct requests sent to the same Web host to different servers with different content. For example, you can configure the appliance to direct requests for dynamic content (such as URLs with a suffix of .asp, .dll, or .exe) to one server and requests for static content to another server. You can configure the appliance to perform content switching based on TCP/IP headers and payload.

You can also use content switching to configure the appliance to redirect requests to different servers with different content on the basis of various client attributes. Some of those client attributes are:

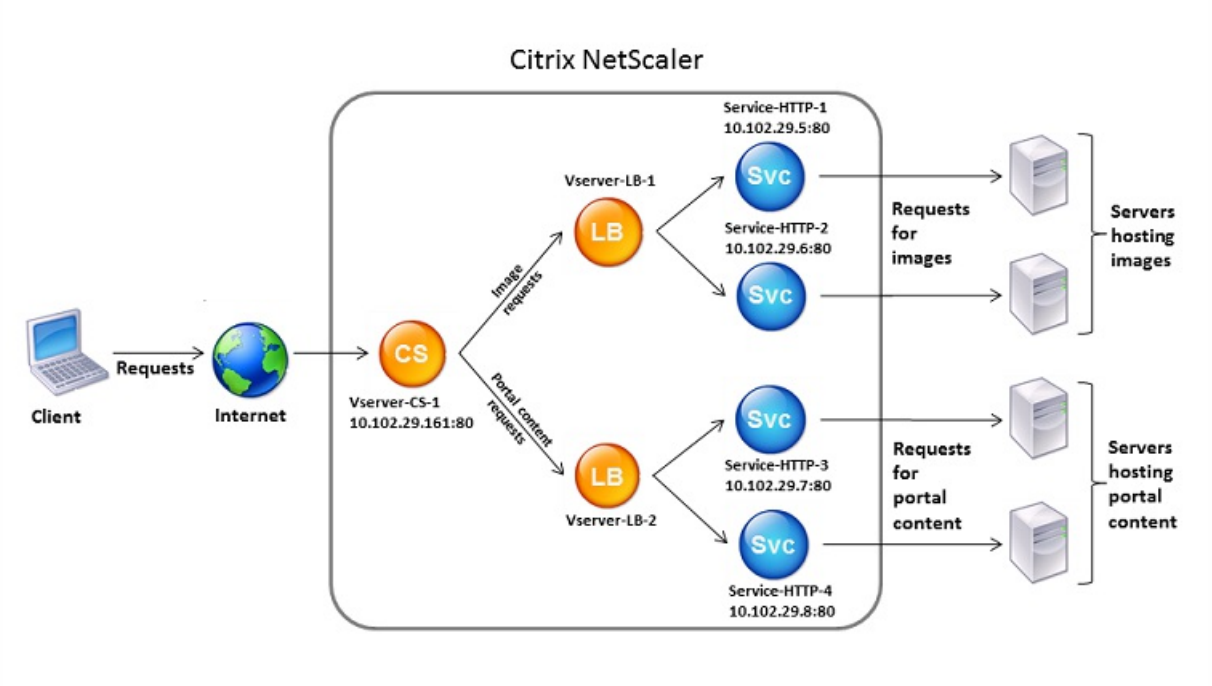
- **Device Type.** The appliance examines the user agent or custom HTTP header in the client request for the type of device from which the request originated. Based on the device type, it directs the request to a specific Web server. For example, if the request came from a cell phone, the request is directed to a server that is capable of serving content that the user can view on his or her cell phone. A request from a computer is directed to a different server that is capable of serving content designed for a computer screen.
- **Language.** The appliance examines the Accept-Language HTTP header in the client request and determines the language used by the client's browser. The appliance then sends the request to a server that serves content in that language. For example, using content switching based on language, the appliance can send someone whose browser is configured to request content in French to a server with the French version of a newspaper. It can send someone else whose browser is configured to request content in English to a server with the English version.
- **Cookie.** The appliance examines the HTTP request headers for a cookie that the server set previously. If it finds the cookie, it directs requests to the appropriate server, which hosts custom content. For example, if a cookie is found that indicates that the client is a member of a customer loyalty program, the request is directed to a faster server or one with special content. If it does not find a cookie, or if the cookie indicates that the user is not a member, the request is directed to a server for the general public.
- **HTTP Method.** The appliance examines the HTTP header for the method used, and sends the client request to the right server. For example, GET requests for images can be directed to an image server, while POST requests can be directed to a faster server that handles dynamic content.
- **Layer 3/4 Data.** The appliance examines requests for the source or destination IP, source or destination port, or any

© 1999-2017 Citrix Systems, Inc. All rights reserved.

other information present in the TCP or UDP headers, and directs the client request to the right server. For example, requests from source IPs that belong to customers can be directed to a custom web portal on a faster server, or one with special content.

A typical content switching deployment consists of the entities described in the following diagram.

Figure 1. Content Switching Architecture



A content switching configuration consists of a content switching virtual server, a load balancing setup consisting of load balancing virtual servers and services, and content switching policies. To configure content switching, you must configure a content switching virtual server and associate it with policies and load balancing virtual servers. This process creates a *content group*—a group of all virtual servers and policies involved in a particular content switching configuration.

Content switching can be used with HTTP, HTTPS, TCP, and UDP connections. For HTTPS, you must enable SSL Offload.

When a request reaches the content switching virtual server, the virtual server applies the associated content switching policies to that request. The priority of the policy defines the order in which the policies bound to the content switching virtual server are evaluated. If you are using default syntax policies, when you bind a policy to the content switching virtual server, you must assign a priority to that policy. If you are using NetScaler classic policies, you can assign a priority to your policies, but are not required to do so. If you assign priorities, the policies are evaluated in the order that you set. If you do not, the NetScaler appliance evaluates your policies in the order in which they were created.

In addition to configuring policy priorities, you can manipulate the order of policy evaluation by using Goto expressions and policy bank invocations. For more details about default syntax policy configuration, see "Configuring Default Syntax Policies."

After it evaluates the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

Content switching virtual servers can only send requests to other virtual servers. If you are using an external load balancer,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

you must create a load balancing virtual server for it and bind its virtual server as a service to the content switching virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Basic Content Switching

Jun 09, 2015

Before you configure content switching, you must understand how content switching is set up and how the services and virtual servers are connected.

To configure a basic, functional content switching setup, first enable the content switching feature. Then, create at least one content group. For each content group, create a content switching virtual server to accept requests to a group of web sites that use content switching. Also create a load balancing setup, which includes a group of load balancing virtual servers to which the content switching virtual server directs requests. To specify which requests to direct to which load balancing virtual server, create at least two content switching policies, one for each type of request that is to be redirected. When you have created the virtual servers and policies, bind the policies to the content switching virtual server. You can also bind a policy to multiple content switching virtual servers. When you bind a policy, you specify the load balancing virtual server to which requests that match the policy are to be directed.

In addition to binding individual policies to a content switching virtual server, you can bind policy labels. If you create additional content groups, you can bind a policy or policy label to more than one of the content switching virtual servers.

Note: After creating a content group, you can modify its content switching virtual serve to customize the configuration. For information on modifying the configuration of an existing content switching virtual server, see "Customizing the Basic Content Switching Configuration." For information on disabling and re-enabling entities, unbinding policies, and removing entities, see "Managing a Content Switching Setup."

This section includes the following details:

- Enabling Content Switching
- Creating Content Switching Virtual Servers
- Configuring a Load Balancing Setup for Content Switching
- Configuring a Content Switching Action
- Configuring Content Switching Policies
- Configuring Content Switching Policy Labels
- Binding Policies to a Content Switching Virtual Server
- Configuring Policy Based Logging for Content Switching
- Verifying the Configuration

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Content Switching

Oct 31, 2013

To use the content switching feature, you must enable content switching. You can configure content switching entities even though the content switching feature is disabled. However, the entities will not work.

To enable content switching by using the command line interface

At the command prompt, type the following commands to enable content switching and verify the configuration:

- enable ns feature CS
- show ns feature

Example

```
> enable  feature ContentSwitch
 Done
> show feature

      Feature                  Acronym          Status
      -------                 -------          ------
 1)   Web Logging              WL               OFF
 2)   Surge Protection         SP               ON
 3)   Load Balancing           LB               ON
 4)   Content Switching        CS               ON
 .
 .
 .
 22)  Responder                RESPONDER        ON
 23)  HTML Injection           HTMLInjection    ON
 24)  NetScaler Push           push             OFF
 Done
```

To enable content switching by using the configuration utility

Navigate to System > Settings and, in the Modes and Features group, select Configure Basic Features, and select Content Switching.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating Content Switching Virtual Servers

Apr 07, 2014

You can add, modify, and remove content switching virtual servers. The state of a virtual server is DOWN when you create it, because the load balancing virtual server is not yet bound to it.

To create a virtual server by using the command line interface

At the command prompt, type:

add cs vserver <name> <protocol> <IPAddress> <port>

Example

add cs vserver Vserver-CS-1 HTTP 10.102.29.161 80

To add a content switching virtual server by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, and add a virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Load Balancing Setup for Content Switching

Dec 17, 2013

The content switching virtual server redirects all requests to a load balancing virtual server. You must create one load balancing virtual server for each version of the content that is being switched. This is true even when your setup has only one server for each version of the content, and you are therefore not doing any load balancing with those servers. You can also configure actual load balancing with multiple load-balanced servers that mirror each version of the content. In either scenario, the content switching virtual server needs to have a specific load balancing virtual server assigned to each version of the content that is being switched.

The load balancing virtual server then forwards the request to a service. If it has only one service bound to it, it selects that service. If it has multiple services bound to it, it uses its configured load balancing method to select a service for the request, and forwards that request to the service that it selected.

To configure a basic load balancing setup, you need to perform the following tasks:

- Create load balancing virtual servers
- Create services
- Bind services to the load balancing virtual server

For more information on load balancing, see "Load Balancing." For detailed instructions on setting up a basic load balancing configuration, see Setting Up Basic Load Balancing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Content Switching Action

Sep 30, 2013

You specify the target load balancing virtual server for a content switching policy when binding the policy to the content switching virtual server. Consequently, you have to configure one policy for each load balancing virtual server to which to direct traffic.

However, if your content switching policy uses a default syntax rule, you can configure an action for the policy. In the action, you can specify the name of the target load balancing virtual server, or you can configure a request-based expression that, at run time, computes the name of the load balancing virtual server to which to send the request. The action expression must be specified in the default syntax.

The expression option can drastically reduce the size of your content switching configuration, because you need only one policy per content switching virtual server. Content switching policies that use an action can also be bound to multiple content switching virtual servers, because the target load balancing virtual server is no longer specified in the content switching policy. The ability to bind a single policy to multiple content switching virtual servers helps to further reduce the size of your content switching configuration.

After you create an action, you create a content switching policy and specify the action in the policy, so that the action is performed when that policy matches a request.

Note: You can also, for a content switching policy that uses a default syntax rule, specify the target load balancing virtual server when binding the policy to a content switching virtual server, instead of using a separate action. For domain-based policies, URL-based policies, and rule based policies that use classic expressions, an action is not available. So, for these types of policies, you specify the name of the target load balancing virtual server when binding the policy to a content switching virtual server. For more information, see "Binding Policies to a Content Switching Virtual Server."

Configuring an Action that Specifies the Name of the Target Load Balancing Virtual Server

If you choose to specify the name of the target load balancing virtual server in a content switching action, you need as many content switching policies as you have target load balancing virtual servers. Content switching decisions, in this case, are based on the rule in the content switching policy, and the action merely specifies the target load balancing virtual server. When a request matches the policy, the request is forwarded to the specified load balancing virtual server.

To create and verify a content switching action that specifies the name of the target load balancing virtual server, by using the command line interface

At the command prompt, type:

- add cs action <name> -targetLBVserver <string> [-comment <string>]
- show cs action <name>

Example

```
> add cs action mycsaction -targetLBVserver mylbvserver -comment "Forwards requests to mylbvserver."
 Done
> show cs action mycsaction
 Name: mycsaction
 Target LB Vserver: mylbvserver
 Hits: 0
 Undef Hits: 0
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Action Reference Count: 0
Comment: "Forwards requests to mylbvserver."

Done
>
```

To configure a content switching action that specifies the name of the target load balancing virtual server, by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Actions.
2. Configure a content switching action, and specify the name of the target load balancing virtual server.

Configuring an Action that Specifies an Expression for Selecting the Target at Run Time

If you choose to configure a request-based expression that can dynamically compute the name of the target load balancing virtual server, you need to configure only one content switching policy to select the appropriate virtual server. The rule for the policy can be a simple TRUE (the policy matches all requests) because, in this case, content switching decisions are based on the expression in the action. By configuring an expression in an action, you can drastically reduce the size of your content switching configuration.

If you choose to configure a request-based expression for computing the name of the target load balancing virtual server at run time, you must carefully consider how to name the load balancing virtual servers in the configuration. You must be able to derive their names by using the request-based policy expression in the action.

For example, if you are switching requests on the basis of the URL suffix (file extension of the requested resource), when naming the load balancing virtual servers, you can follow the convention of appending the URL suffix to a predetermined string, such as mylb_. For example, load balancing virtual servers for HTML pages and PDF files could be named mylb_html and mylb_pdf, respectively. In that case, the rule that you can use in the content switching action, to select the appropriate load balancing virtual server, is "mylb_"+HTTP.REQ.URL.SUFFIX. If the content switching virtual server receives a request for an HTML page, the expression returns mylb_html, and the request is switched to virtual server mylb_html.

To create a content switching action that specifies an expression, by using the command line interface

At the command line, type the following commands to create a content switching action that specifies an expression and verify the configuration:

- add cs action <name> -targetVserverExpr <expression>) [-comment <string>]
- show cs action <name>

Example

```
> add cs action mycsaction1 -targetvserverExpr '"mylb_" + HTTP.REQ.URL.SUFFIX'
 Done
> show cs action mycsaction1
 Name: mycsaction1
 Target Vserver Expression: "mylb_" + HTTP.REQ.URL.SUFFIX
 Target LB Vserver: No_Target
            ...
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

>

To configure a content switching action that specifies an expression by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Actions.
2. Configure a content switching action, and specify an expression that will dynamically compute the name of the target load balancing virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Content Switching Policies

Aug 22, 2013

A content switching policy defines a type of request that is to be directed to a load balancing virtual server. These policies are applied in the order of the priorities assigned to them or (if you are using NetScaler classic policies and do not assign priorities when binding them) in the order in which the policies were created.

The policies can be:

- **Domain-based policies.** The NetScaler appliance compares the domain of an incoming URL with the domains specified in the policies. The appliance then returns the most appropriate content. Domain-based policies must be classic policies; default syntax policies are not supported for this type of content switching policy.
- **URL-based policies.** The appliance compares an incoming URL with the URLs specified in the policies. The appliance then returns the most appropriate URL-based content, which is usually the longest matching configured URL. URL-based policies must be classic policies; default syntax policies are not supported for this type of content switching policy.
- **Rule-based policies.** The appliance compares incoming data to expressions specified in the policies. You create rule-based policies by using either a classic expression or a default syntax expression. Both classic and default syntax policies are supported for rule-based content switching policies.

  Note: A rule based policy can be configured with an optional action. A policy with an action can be bound to multiple virtual servers or policy labels.

  If you set a priority when binding your policies to the content switching virtual server, the policies are evaluated in order of priority. If you do not set specific priorities when binding your policies, the policies are evaluated in the order in which they were created.

For information about NetScaler classic policies and expressions, see "Configuring Classic Policies and Expressions." For information about Default Syntax policies, see "Configuring Default Syntax Expressions."

To create a content switching policy by using the command line interface

At the command prompt, type one of the following commands:

- add cs policy <policyName> -domain <domain>
- add cs policy <policyName> -url <URLValue>
- add cs policy <policyName> -rule <RULEValue>
- add cs policy <policyName> -rule <RULEValue> -action <actionName>

Example

```
add cs policy Policy-CS-1 -url "http://example.com"

add cs policy Policy-CS-1 -domain "example.com"

add cs policy Policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(24).EQ(10.217.84.0)"

add cs policy Policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2009 Nov,GMT 2009 Dec)"
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add cs policy Policy-CS-3 -rule "http.req.method.eq(GET)" -action act1
```

To rename a content switching policy by using the command line interface

At the command prompt, type:

```
rename cs policy <policyName> <newName>
```

Example

```
rename cs policy myCSPolicy myCSPolicy1
```

To rename a content switching policy by using the configuration utility

Navigate to Traffic Management > Content Switching > Policies, select a policy and, in the Action list, select Rename.

To create a content switching policy by using the configuration utility

Navigate to Traffic Management > Content Switching > Policies, and configure a content switching policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Content Switching Policy Labels

Oct 31, 2013

A policy label is a user-defined bind point to which policies are bound. When a policy label is invoked, all the policies bound to it are evaluated in the order of the priority that you assigned to them. A policy label can include one or more policies, each of which can be assigned its own result. A match on one policy in the policy label can result in proceeding to the next policy, invoking a different policy label or appropriate resource, or an immediate end to policy evaluation and return of control to the policy that invoked the policy label. You can create policy labels for default syntax policies only.

For information about policy labels, see the "Creating Policy Labels."

A content switching policy label consists of a name, a label type, and a list of policies bound to the policy label. The policy label type specifies the protocol that was assigned to the policies bound to the label. It must match the service type of the content switching virtual server to which the policy that invokes the policy label is bound. For example, you can bind TCP Payload policies to a policy label of type TCP only. Binding TCP Payload policies to a policy label of type HTTP is not supported.

Each policy in a content switching policy label is associated with either a target (which is equivalent to the action that is associated with other types of policies, such as rewrite and responder policies) or a gotoPriorityExpression option and/or an invoke option. That is, for a given policy in a content switching policy label, you can specify a target, or you can set the gotoPriorityExpression option and/or the invoke option. Additionally, if multiple policies evaluate to true, only the target of the last policy that evaluates to true is considered.

You can use either the NetScaler command line or the configuration utility to configure content switching policy labels. In the NetScaler command-line interface (CLI), you first create a policy label by using the add cs policylabel command. Then, you bind policies to the policy label, one policy at a time, by using the bind cs policylabel command. In the NetScaler configuration utility, you perform both tasks in a single dialog box.

To create a content switching policy label by using the command line interface

At the command prompt, type:

add cs policylabel <labelName> <cspolicylabelTypetype>

Example

```
add cs policylabel testpollab http
```

To rename a content switching policy label by using the command line interface

At the command prompt, type:

rename cs policylabel <labelName> <newName>

Example

```
rename cs policylabel oldPolicyLabelName newPolicyLabelName
```

To rename a content switching policy label by using the configuration utility

Navigate to Traffic Management > Content Switching > Policy Labels , select a policy label and, in the Action list, select Rename.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To bind a policy to a content switching policy label by using the command line interface

At the command prompt, type the following commands to bind a policy to a policy label and verify the configuration:

- bind cs policylabel <labelName> <policyName> <priority>[[-targetVserver <string>] | [-gotoPriorityExpression <expression>] | [-invoke <labeltype> <labelName>]]
- show cs policylabel <labelName>

Example

```
bind cs policylabel testpollab test_Pol 100 -targetVserver LBVIP
show cs policylabel testpollab
    Label Name: testpollab
    Label Type: HTTP
    Number of bound policies: 1
    Number of times invoked: 0
1)   Policy Name: test_Pol
    Priority: 100
    Target Virtual Server: LBVIP
```

Note: If a policy is configure with an action, the target virtual server (targetVserver), go to priority expression (gotoPriorityExpression), and invoke (invoke) parameters are not required. If a policy is not configure with an action, you need to configure at least one of the following parameters: targetVserver, gotoPriorityExpression, and invoke.

To unbind a policy from a policy label by using the command line interface

At the command prompt, type the following commands to unbind a policy from a policy label and verify the configuration:

- unbind cs policylabel <labelName> <policyName>
- show cs policylabel <labelName>

Example

```
unbind cs policylabel testpollab test_Pol
show cs policylabel testpollab
    Label Name: testpollab
    Label Type: HTTP
    Number of bound policies: 0
    Number of times invoked: 0
```

To remove a policy label by using the command line interface

At the command prompt, type:

rm cs policylabel <labelName>

To manage a content switching policy label by using the configuration utility

Navigate to Traffic Management > Content Switching > Policy Labels, configure a policy label, bind policies to the label, and optionally specify a priority, gotoPriority expression, and an invoke option.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding Policies to a Content Switching Virtual Server

Sep 03, 2014

After you create your content switching virtual server and policies, you bind each policy to the content switching virtual server. When binding the policy to the content switching virtual server, you specify the target load balancing virtual server.

Note: If your content switching policy uses a default syntax rule, you can configure a content switching action for the policy. If you configure an action, you must specify the target load balancing virtual server when you are configuring the action, not when you are binding the policy to the content switching virtual server. For more information about configuring a content switching action, see Configuring a Content Switching Action.

To bind a policy to a content switching virtual server and select a target load balancing virtual server by using the command line interface

At the command prompt, type:

bind cs vserver <name>[-lbvserver<string> -targetLBVServer<string> -policyname <string> -priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type (REQUEST | RESPONSE)] [-invoke (<labelType> <labelName>)]

Example

```
bind cs vserver csw-vip2 -policyname csw-ape-policy2 -priority 14 -gotoPriorityExpression NEXT

bind cs vserver csw-vip3 -policyname rewrite-policy1 -priority 17 -gotoPriorityExpression
'q.header("a").count' -flowtype REQUEST -invoke policylabel label1

bind cs vserver Vserver-CS-1 Vserver-LB-1 -policyname Policy-CS-1 -priority 20
```

Note: The parameters, target load balancing virtual server (targetVserver), go to priority expression (gotoPriorityExpression), and invoke method (invoke) cannot be used if a policy has an action.

To bind a policy to a content switching virtual server and select a target load balancing virtual server by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, open a virtual server and, in the Content Switching Policy Binding section, bind a policy to the virtual server, and specify a target load balancing virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Policy Based Logging for Content Switching

Jun 05, 2017

You can configure policy based logging for a content switching policy. Policy based logging enables you to specify a format for log messages. The contents of the log message are defined by using a default syntax expression in the content switching policy. When the content switching action specified in the policy is performed, the NetScaler appliance constructs the log message from the expression and writes the message to the log file. Policy based logging is particularly useful if you want to test and troubleshoot a configuration in which content switching actions identify the target load balancing virtual server at run time.

Note: If multiple policies bound to a given virtual server evaluate to TRUE and are configured with an audit message action, the NetScaler appliance does not perform all the audit message actions. It performs only the audit message action that is configured for the policy whose content switching action is performed.
To configure policy based logging for a content switching policy, you must first configure an audit message action. For more information about configuring an audit message action, see Configuring the NetScaler Appliance for Audit Logging. After you configure the audit message action, you specify the action in a content switching policy.

To configure policy based logging for a content switching policy by using the command line interface

At the command line, type the following commands to configure policy based logging for a content switching policy and verify the configuration:

- set cs policy <policyName> -logAction <string>
- show cs policy <policyName>

Example

```
> set cs policy cspol1 -logAction csLogAction
 Done
> show cs policy cspol1

 Policy: cspol1 Rule: TRUE Action: csact1
 LogAction: csLogAction
 Hits: 0

1) CS Vserver: csvs1
 Priority: 10
 Done
>
```

To configure policy based logging for a content switching policy by using the configuration utility

Navigate to Traffic Management > Content Switching > Policies, open a policy and, in the Log Action list, select a log action for the policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Verifying the Configuration

Mar 15, 2012

To verify that your content switching configuration is correct, you need to view the content switching entities. To verify proper operation after your content switching configuration has been deployed, you can view the statistics that are generated as the servers are accessed.

Viewing the Properties of Content Switching Virtual Servers

Updated: 2013-10-31

You can view the properties of content switching virtual servers that you have configured on the NetScaler. You can use the information to verify whether the virtual server is correctly configured and, if necessary, to troubleshoot. In addition to details such as name, IP address, and port, you can view the various policies bound to a virtual server, and its traffic-management settings.

The content switching policies are displayed in the order of their priority. If more than one policy has the same priority, they are shown in the order in which they are bound to the virtual server.

Note: If you have configured the content switching virtual server to forward traffic to a load balancing virtual server, you can also view the content switching policies by viewing the properties of the load balancing virtual server.

To view the properties of content switching virtual servers by using the command line interface

To list basic properties of all content switching virtual servers in your configuration, or detailed properties of a specific content switching virtual server, at the command prompt, type one of the following commands:

- show cs vserver
- show cs vserver <name>

Example

1.

```
show cs vserver Vserver-CS-1
Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
Last state change was at Thu Jun 30 10:48:59 2011
Time since last state change: 6 days, 20:03:00.760
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
Default: Content Precedence: RULE
Vserver IP and Port insertion: OFF
Case Sensitivity: ON
Push: DISABLED Push VServer:
Push Label Rule: none
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

...
1) Policy : __ESNS_PREBODY_POLICY Priority:0
2) Policy : __ESNS_POSTBODY_POLICY Priority:0

1) Compression Policy Name: __ESNS_CMP_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

1) Rewrite Policy Name: __ESNS_REWRITE_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

1) Cache Policy Name: dfbx Priority: 10
GotoPriority Expression: END
Flowtype: REQUEST

1) Responder Policy Name: __ESNS_RESPONDER_POLICY Priority: 2147483647
GotoPriority Expression: END

1) Policy: wiki Target: LBVIP2 Priority: 25 Hits: 0
2) Policy: plain Target: LBVIP1 Priority: 90 Hits: 0
3) Policy: DispOrderTest2 Target: KerbAuthLBVS Priority: 91 Hits: 0
4) Policy: test_Pol Target: LBVIP1 Priority: 92 Hits: 0
5) Policy: PolicyNameTesting Target: LBVIP1 Priority: 100 Hits: 0
Done
>

2.
show cs vserver
1) Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
…
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED

2) apubendpt (10.111.111.1:80) - HTTP Type: CONTENT
State: UP
…
Client Idle Timeout: 180 sec
Down state flush: DISABLED
…

3) apubendpt1 (10.111.111.2:80) - HTTP Type: CONTENT
State: UP
…
Disable Primary Vserver On Down : DISABLED

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
...
```

Viewing Content Switching Policies

You can view the properties of the content switching policies that you defined, such as the name, domain, and URL or expression, and use the information to find any mistakes in the configuration, or to troubleshoot if something is not working as it should.

To view the properties of content switching policies by using the command line interface

To list either basic properties of all content switching policies in your configuration or detailed properties of a specific content switching policy, at the command prompt, type one of the following commands:

- show cs policy
- show cs policy <PolicyName>

Example

```
show cs policy
```

```
show cs policy Policy-CS-1
```

To view the properties of content switching policies by using the configuration utility

Navigate to Traffic Management > Content Switching > Policies, select a policy and, in the Action list, select Show Bindings.

Viewing a Content Switching Virtual Server Configuration by Using the Visualizer

The Content Switching Visualizer is a tool that you can use to view a content switching configuration in graphical format. You can use the visualizer to view the following configuration items:

- A summary of the load balancing virtual servers to which the content switching virtual server is bound.
- All services and service groups that are bound to the load balancing virtual server and all monitors that are bound to the services.
- The configuration details of any displayed element.
- Any policies bound to the content switching virtual server. These policies need not be content switching policies. Many types of policies, such as Rewrite policies, can be bound to a content switching virtual server.

After you configure the various elements in a content switching and load balancing setup, you can export the entire configuration to an application template file.

Note: The Visualizer requires a graphical interface, so it is available only through the configuration utility.

To view a content switching configuration by using the Visualizer in the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.
3. In the Content Switching Visualizer window, you can adjust the viewable area as follows:
   - Click the Zoom In and Zoom Out icons to increase or decrease the viewable area.
   - Click the Save Image icon to save the graph as an image file.
   - In the Search in text field, begin typing the name of the item you are looking for. When you have typed enough characters to identify the item, its location is highlighted. To restrict the search, click the drop-down menu and select the type of element that you want to search for.
4. To view configuration details for entities that are bound to this virtual server, you can do the following:
   - To view policies that are bound to the virtual server, in the tool bar at the top of the dialog box select one or more feature-specific policy icons. If policy labels are configured, they appear in the main view area.
   - To view the configuration details for a bound service or service group, click the icon for the service, click the Related Tasks tab, and then click Show Member Services.
   - To view the configuration details for a monitor, click the icon for the monitor, click the Related Tasks tab, and then click View Monitor.
5. To view detailed statistics for any virtual server in the content switching configuration, click the virtual server for which you want to view statistics, then click the Related Tasks tab, and then click Statistics.
6. To view a comparative list of the parameters whose values either differ or are not defined across service containers for a load balancing virtual server, click the icon for a container, click the Related Tasks tab, and then click Service Attributes Diff.
7. To view monitor binding details for the services in a container, in the Service Attributes Diff dialog box, in the Group column for the container, click Details. This comparative list helps you determine which service container has the configuration you want to apply to all the service containers.
8. To view the number of requests received per second at a given point in time by the virtual servers in the configuration, and the number of hits per second at a given point in time for rewrite, responder, and cache policies, click Show Stats. The statistical information is displayed on the respective nodes in the Visualizer. This information is not updated in real time. It has to be refreshed manually. To refresh the information, click Refresh Stats.
   Note: This option is available only on NetScaler nCore builds.
9. To copy configuration details for an element to a document or spreadsheet, click the icon for that element, click Related Tasks, click Copy Properties, and then paste the information into a document.
10. To export the entire configuration that is displayed in the Visualizer to an application template file, click the icon for the content switching virtual server, click Related Tasks, and then click Create Template. When creating the application template, you can configure variables in some policy expressions and actions. For more information about creating the application template file and configuring variables for a template, see AppExpert.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Customizing the Basic Content Switching Configuration

Jun 08, 2015

After you configure a basic content switching setup, you might need to customize it to meet your requirements. If your web servers are UNIX-based and rely on case sensitive pathnames, you can configure case sensitivity for policy evaluation. You can also set precedence for evaluation of the content switching policies that you configured. You can configure HTTP and SSL content switching virtual servers to listen on multiple ports instead of creating separate virtual servers. If you want to configure content switching for a specific a virtual LAN, you can configure a content switching virtual server with a listen policy.

To customize the basic content switching configuration, see the following sections:

- Configuring Case Sensitivity for Policy Evaluation
- Setting the Precedence for Policy Evaluation
- Support for Multiple Ports for HTTP and SSL Type Content Switching Virtual Servers
- Configuring per-VLAN Wildcarded Virtual Servers
- Configuring the Microsoft SQL Server Version Setting

Configuring Case Sensitivity for Policy Evaluation

You can configure the content switching virtual server to treat URLs as case sensitive in URL-based policies. When case sensitivity is configured, the NetScaler appliance considers case when evaluating policies. For example, if case sensitivity is off, the URLs /a/1.htm and /A/1.HTM are treated as identical. If case sensitivity is on, those URLs are treated as separate and can be switched to different targets.

To configure case sensitivity by using the command line interface

At the command prompt, type:

set cs vserver <name> -caseSensitive (ON | OFF)

Example

set cs vserver Vserver-CS-1 -caseSensitive ON

To configure case sensitivity by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and then select Case Sensitive.

Setting the Precedence for Policy Evaluation

Precedence refers to the order in which policies that are bound to a virtual server are evaluated. You do not normally have to configure precedence: the default precedence works correctly in many cases. If you want to make sure that one policy or set of policies is applied first, however, and another policy or set of policies is applied only if the first set does not match a request, you can configure either URL-based precedence or rule-based precedence.

Precedence with URL-Based Policies

If there are multiple matching URLs for the incoming request, the precedence (priority) for URL-based policies is:

1. Domain and exact URL
2. Domain, prefix, and suffix
3. Domain and suffix
4. Domain and prefix
5. Domain only
6. Exact URL
7. Prefix and suffix
8. Suffix only
9. Prefix only
10. Default

If you configure precedence based on URL, the request URL is compared to the configured URLs. If none of the configured URLs match the request URL, then rule-based policies are checked. If the request URL does not match any rule-based policies, or if the content group selected for the request is down, then the request is processed as follows:

- If you configure a default group for the content switching virtual server, then the request is forwarded to the default group.
- If the configured default group is down or if no default group is configured, then an “HTTP 404 Not Found” error message is sent to the client.

Note: You should configure URL-based precedence if the content type (for example, images) is the same for all clients. However, if different types of content must be served based on client attributes (such as Accept-Language), you must use rule-based precedence.

Precedence with Rule-Based Policies

If you configure precedence based on rules, which is the default setting, the request is tested on the basis of the rule-based policies you have configured. If the request does not match any rule-based policies, or if the content group selected for the incoming request is down, the request is processed in the following manner:

- If a default group is configured for the content switching virtual server, the request is forwarded to the default group.
- If the configured default group is down or if no default group is configured, an “HTTP 404 Not Found” error message is sent to the client.

To configure precedence by using the command line interface

At the command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set cs vserver <name> -precedence ( RULE | URL )
```

Example

```
set cs vserver Vserver-CS-1 -precedence RULE
```

To configure precedence by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and then specify Precedence.

Support for Multiple Ports for HTTP and SSL Type Content Switching Virtual Servers

You can configure the NetScaler ADC so that HTTP and SSL content switching virtual servers listen on multiple ports, without having to configure separate virtual servers. This feature is especially useful if you want to base a content switching decision on a part of the URL and other L7 parameters. Instead of configuring multiple virtual servers with the same IP address and different ports, you can configure one IP address and specify the port as *. As a result, the configuration size is also reduced.

To configure an HTTP or SSL content switching virtual server to listen on multiple ports by using the command line

At the command prompt, type:

```
add cs vserver <name> <serviceType> <IPAddress> Port *
```

Example

```
> add cs vserver cs1 HTTP 10.102.92.215 *
 Done
> sh cs vserver cs1
      cs1 (10.102.92.215:*) - HTTP    Type: CONTENT
      State: UP
      Last state change was at Tue May 20 01:15:49 2014
      Time since last state change: 0 days, 00:00:03.270
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Appflow logging: ENABLED
      Port Rewrite : DISABLED
      State Update: DISABLED
      Default:       Content Precedence: RULE
      Vserver IP and Port insertion: OFF
      L2Conn: OFF     Case Sensitivity: ON
      Authentication: OFF
      401 Based Authentication: OFF
      Push: DISABLED  Push VServer:
      Push Label Rule: none
      IcmpResponse: PASSIVE
      RHIstate:  PASSIVE
      TD: 0
 Done
```

To configure an HTTP or SSL content switching virtual server to listen on multiple ports by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and create a virtual server of type HTTP or SSL.
2. Use an asterisk (*) to specify the port.

Configuring per-VLAN Wildcarded Virtual Servers

If you want to configure content switching for traffic on a specific virtual local area network (VLAN), you can create a wildcarded virtual server with a listen policy that restricts it to processing traffic only on the specified VLAN.

To configure a wildcarded virtual server that listens to a specific VLAN by using the command line interface

At the command prompt, type:

```
add cs vserver <name> <serviceType> IPAddress * Port * -listenpolicy <expression> [-listenpriority <positive_integer>]
```

Example

```
add cs vserver Vserver-CS-vlan1 ANY * *
    -listenpolicy "CLIENT.VLAN.ID.EQ(2)"  -listenpriority 10
```

To configure a wildcarded virtual server that listens to a specific VLAN by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, and configure a virtual server. Specify a listen policy that restricts it to processing traffic only on the specified VLAN.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

After you have created this virtual server, you bind it to one or more services as described in Binding Services to the Virtual Server.

Configuring the Microsoft SQL Server Version Setting

You can specify the version of Microsoft® SQL Server® for a content switching virtual server that is of type MSSQL. The version setting is recommended if you expect some clients to not be running the same version as your Microsoft SQL Server product. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version.

To set the Microsoft SQL Server version parameter by using the command line interface

At the command prompt, type the following commands to set the Microsoft SQL Server version parameter for a content switching virtual server and verify the configuration:

- set cs vserver <name> -mssqlServerVersion <mssqlServerVersion>
- show cs vserver <name>

Example

> set cs vserver myMSSQLcsvip -mssqlServerVersion 2008R2 Done > show cs vserver myMSSQLcsvip myMSSQLcsvip (192.0.2.13:1433) - MSSQL Type: CONTENT State: UP . . . . . MSsql Server Version: 2008R2 . . . . . Done >

To set the Microsoft SQL Server version parameter by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, configure a virtual server and specify the protocol as MYSQL.
2. In Advanced Settings, select MySQL, and specify the Server Version.

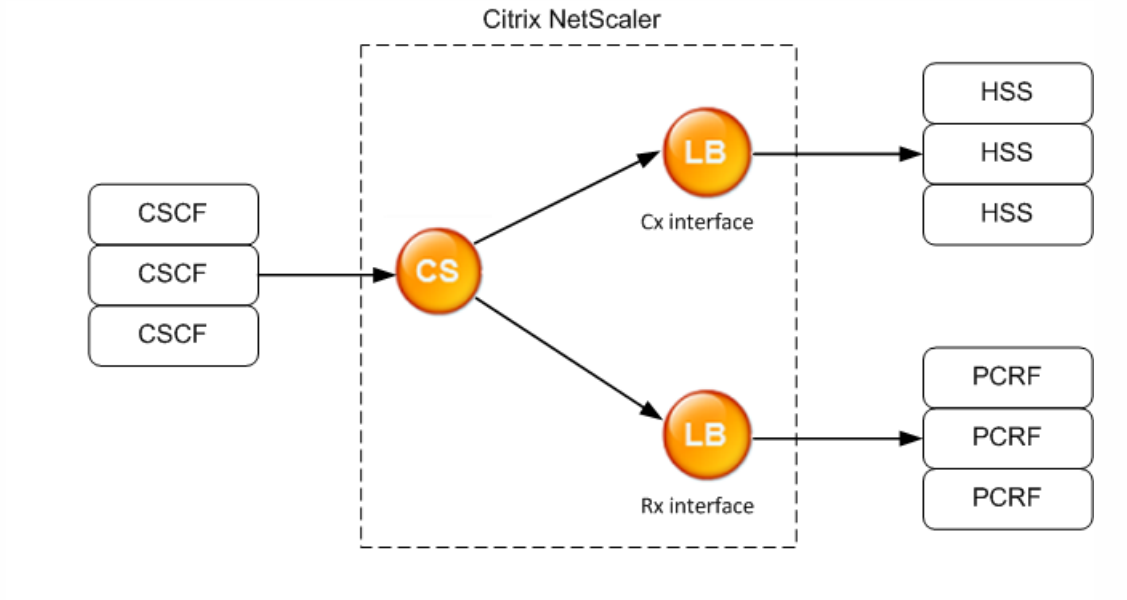
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Content Switching for Diameter Protocol

Jun 26, 2014

For Diameter-protocol traffic, you can configure the NetScaler appliance (or virtual appliance) to act as a relay agent that load balances and forwards a packet to the appropriate destination on the basis of the message content (AVP value in the message). Since the appliance does not perform any application-level processing, it provides relaying services for all diameter applications as specified by the configured content switching policies. Therefore, the appliance advertises the Relay Application ID in the capability exchange answer (CEA) message when the client establishes a diameter connection. You must configure a content switching virtual server, load balancing virtual servers, and services to represent the diameter nodes. When a request reaches the content switching virtual server, the virtual server applies the content switching policies associated with that type of request. After evaluating the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

A diameter interface provides a connection between the different diameter nodes. The following sample deployment uses Cx and Rx interfaces. A Cx interface provides a connection between a CSCF and an HSS. An Rx interface provides a connection between a CSCF and a PCRF. All the messages reach the NetScaler appliance. Depending on whether the message is for a Cx or an Rx interface, and on the content switching policies defined, the NetScaler selects an appropriate load balancing server pool.



CSCF=Call Session Control Function
HSS=Home Subscriber Server
PCRF=Policy and Charging Rules Function

Sample Configuration

1. For each entity, create a service, a load balancing server, and bind the service to the virtual server.

```
add service svc_pcrf[1-3] 1.1.1.1[1-3] DIAMETER 3868
add service svc_hss[1-3] 1.1.1.2[1-3] DIAMETER 3868
add lb vserver vs_rx DIAMETER -persistenceType DIAMETER –persistavpno 263
add lb vserver vs_cx DIAMETER -persistenceType DIAMETER –persistavpno 263
bind lb vserver vs_rx svc_pcrf[1-3]
bind lb vserver vs_cx svc_hss[1-3]
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Create a content switching virtual server and two actions (one for each load balancing virtual server). Create two content switching policies and bind these policies to the content switching virtual server, specifying a priority for each policy.

```
add cs vserver cs_diameter DIAMETER 10.1.1.10 3868
add cs action cx_action -targetLBVserver vs_cx
add cs action rx_action –targetLBvserver vs_rx
add cs policy cx_policy -rule "DIAMETER.REQ.AUTH_APPLICATION_ID.EQ(16777216)" -action cx_action
add cs policy rx_policy -rule "DIAMETER.REQ.AUTH_APPLICATION_ID.EQ(16777236)" -action rx_action
bind cs vserver cs_diameter -policyName rx_policy -priority 100
bind cs vserver cs_diameter -policyName cx_policy -priority 110
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Protecting the Content Switching Setup against Failure

Jun 08, 2015

Content switching may fail when the content switching virtual server goes DOWN or fails to handle excessive traffic, or for other reasons. To reduce the chances of failure, you can take the following measures to protect the content switching setup against failure:

- Configure a backup content switching virtual server
- Configure spillover for preventing the overloading of the primary and diverting excess traffic to the backup virtual server
- Specify a redirect URL, the URL to which the content is switched if both the primary and backup content switching virtual servers are DOWN
- Enable the State Update option for marking a content switching virtual server as DOWN when the load balancing virtual server is DOWN
- Flush the surge queues when the queues become too long

Configuring a Backup Virtual Server

If the primary content switching virtual server is marked DOWN or DISABLED, the NetScaler appliance can direct requests to a backup content switching virtual server. It can also send a notification message to the client regarding the site outage or maintenance. The backup content switching virtual server is a proxy and is transparent to the client.

When configuring the backup virtual server, you can specify the configuration parameter Disable Primary When Down to ensure that, when the primary virtual server comes back up, it remains the secondary until you manually force it to take over as the primary. This is useful if you want to ensure that any updates to the database on the server for the backup are preserved, enabling you to synchronize the databases before restoring the primary virtual server.

You can configure a backup content switching virtual server when you create a content switching virtual server or when you change the optional parameters of an existing content switching virtual server. You can also configure a backup content switching virtual server for an existing backup content switching virtual server, thus creating cascaded backup content switching virtual servers. The maximum depth of cascaded backup content switching virtual servers is 10. The appliance searches for a backup content switching virtual server that is up and accesses that content switching virtual server to deliver the content.

Note: If a content switching virtual server is configured with both a backup content switching virtual server and a redirect URL, the backup content switching virtual server takes precedence over the redirect URL. The redirect is used when the primary and backup virtual servers are down.

To set up a backup content switching virtual server by using the command line interface

At the command prompt, type:

```
set cs vserver <name> -backupVserver <string> -disablePrimaryOnDown (ON | OFF)
```

Example

```
set cs vserver Vserver-CS-1 -backupVserver Vserver-CS-2 -disablePrimaryOnDown ON
```

To set up a backup content switching virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, configure a virtual server and specify the protocol as MYSQL.
2. In Advanced Settings, select Protection, and specify a Backup Virtual Server.

Diverting Excess Traffic to a Backup Virtual Server

The spillover option diverts new connections arriving at a content switching virtual server to a backup content switching virtual server when the number of connections to the content switching virtual server exceeds the configured threshold value. The threshold value is dynamically calculated, or you can set the value. The number of established connections (in case of TCP) at the virtual server is

© 1999-2017 Citrix Systems, Inc. All rights reserved.

compared with the threshold value. When the number of connections reaches the threshold, new connections are diverted to the backup content switching virtual server.

If the backup content switching virtual servers reach the configured threshold and are unable to take the load, the primary content switching virtual server diverts all requests to the redirect URL. If a redirect URL is not configured on the primary content switching virtual server, subsequent requests are dropped.

To configure a content switching virtual server to divert new connections to a backup virtual server by using the command line interface

At the command prompt, type:

```
set cs vserver <name> -soMethod <methodType> -soThreshold <thresholdValue> -soPersistence <persistenceValue>
-soPersistenceTimeout <timeoutValue>
```

Example

```
set cs vserver Vserver-CS-1 -soMethod Connection -soThreshold 1000 -soPersistence enabled -soPersistenceTimeout 2
```

To set a content switching virtual server to divert new connections to a backup virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, configure a virtual server and specify the protocol as MYSQL.
2. In Advanced Settings, select Protection, and configure spillover.

Configuring a Redirection URL

You can configure a redirect URL to communicate the status of the NetScaler appliance in the event that a content switching virtual server of type HTTP or HTTPS is DOWN or DISABLED. This URL can be local or remote.

Redirect URLs can be absolute URLs or relative URLs. If the configured redirect URL contains an absolute URL, the HTTP redirect is sent to the configured location, regardless of the URL specified in the incoming HTTP request. If the configured redirect URL contains only the domain name (relative URL), the HTTP redirect is sent to a location after appending the incoming URL to the domain configured in the redirect URL.

Citrix recommends using an absolute URL. That is, a URL ending in /, for example www.example.com/ instead of a relative URL. A relative URL redirection might result in the vulnerability scanner reporting a false positive.

Note: If a content switching virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect URL is used when the primary and backup virtual servers are down.
When redirection is configured and the content switching virtual server is unavailable, the appliance issues an HTTP 302 redirect to the user's browser.

To configure a redirect URL for when the content switching virtual server is unavailable by using the command line interface

At the command prompt, type:

```
set cs vserver <name> -redirectURL <URLValue>
```

Example

```
set cs vserver Vserver-CS-1 -redirectURL http://www.newdomain.com/mysite/maintenance
```

To configure a redirect URL for when the content switching virtual server is unavailable by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, configure a virtual server and specify the protocol as MYSQL.
2. In Advanced Settings, select Protection, and specify a Redirect URL.

Configuring the State Update Option

The content switching feature enables the distribution of client requests across multiple servers on the basis of the specific content

© 1999-2017 Citrix Systems, Inc. All rights reserved.

presented to the users. For efficient content switching, the content switching virtual server distributes the traffic to the load balancing virtual servers according to the content type, and the load balancing virtual servers distribute the traffic to the physical servers according to the specified load balancing method.

For smooth traffic management, it is important for the content switching virtual server to know the status of the load balancing virtual servers. The state update option helps to mark the content switching virtual server DOWN if the load balancing virtual server bound to it is marked DOWN. A load balancing virtual server is marked DOWN if all the physical servers bound to it are marked DOWN.

When State Update is disabled:

The status of the content switching virtual server is marked as UP. It remains UP even if there is no bound load balancing virtual server that is UP.

When State Update is enabled:

When you add a new content switching virtual server, initially, its status is shown as DOWN. When you bind a load balancing virtual server whose status is UP, the status of the content switching virtual server becomes UP.

If more than one load balancing virtual server is bound and if one of them is specified as the default, the status of the content switching virtual server reflects the status of the default load balancing virtual server.

If more than one load balancing virtual server is bound without any of them being specified as the default, the status of the content switching virtual server is marked UP only if all the bound load balancing virtual servers are UP.

To configure the state update option by using the command line interface

At the command prompt, type:

add cs vserver <name> <protocol> <ipAddress> <port> -stateUpdate ENABLED
Example

add cs vserver csw_vserver HTTP 10.18.250.154 80 -stateupdate ENABLED -cltTimeout 180

To configure the state update option by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, configure a virtual server and specify the protocol as MYSQL.
2. In Advanced Settings, select Traffic Settings, and then select State Update.

Flushing the Surge Queue

When a physical server receives a surge of requests, it becomes slow to respond to the clients that are currently connected to it, which leaves users dissatisfied and disgruntled. Often, the overload also causes clients to receive error pages. To avoid such overloads, the NetScaler appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.
The appliance does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the appliance looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the appliance establishes a new connection with the server, and establishes a virtual link between client and the server. However, if the appliance cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the appliance cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The appliance maintains a different surge queue for each service bound to the virtual server.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The length of a surge queue increases whenever a request comes for which the appliance cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you may want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed; the other requests in the surge queue of the load balancing virtual server are not flushed.

Note: You cannot flush the surge queues of cache redirection, authentication, VPN or GSLB virtual servers or GSLB services.
Note: Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

To flush a surge queue by using the command line interface

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while executing the command, surge queue of the specified entity will be flushed. If more than one entity has the same name, the appliance flushes surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while executing the command, the appliance flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member (<serverName> and <port>) without specifying the name of the service group (<name>) and you cannot specify <port> without a <serverName>. Specify the <serverName> and <port> if you want to flush the surge queue for a specific service group member.
- If you execute the command without specifying any names, the appliance flushes the surge queues of all the entities present on the appliance.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the command prompt, type:

flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
Examples

1.
flush ns surgeQ –name SVC1ANZGB –serverName 10.10.10.1 80
The above command flushes the surge queue of the service or virtual server that is named SVC1ANZGB and has IP address as 10.10.10
2.
flush ns surgeQ
The above command flushes all the surge queues on the appliance.

To flush a surge queue by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, select a virtual server and, in the Action list, select Flush Surge Queue.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing a Content Switching Setup

Jun 08, 2015

After a content switching setup is configured, it may require periodic changes. When operating systems or software are updated, or hardware wears out and is replaced, you may need to take down your setup. Load on your setup may increase, requiring additional resources. You may also modify the configuration to improve performance.

These tasks may require unbinding policies from the content switching virtual server, or disabling or removing content switching virtual servers. After you have made changes to your setup, you may need to re-enable servers and rebind policies. You might also want to rename your virtual servers.

To manage a content switching setup, see the following sections:

- Unbinding Policies from the Content Switching Virtual Server
- Removing Content Switching Virtual Servers
- Disabling and Re-Enabling Content Switching Virtual Servers
- Renaming Content Switching Virtual Servers
- Managing Content Switching Policies
- Modifying a Content Switching Configuration by Using the Visualizer

Unbinding Policies from the Content Switching Virtual Server

When you unbind a content switching policy from its virtual server, the virtual server no longer includes that policy when determining where to direct requests.

To unbind a policy from a content switching virtual server by using the command line interface

At the command prompt, type:

unbind cs vserver <name> -policyname <string>

Example

```
unbind cs vserver Vserver-CS-1 -policyname Policy-CS-1
```

To unbind a policy from a content switching virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open the virtual server.
2. Click in the Policies section, select the policy, and click Unbind.

Removing Content Switching Virtual Servers

You normally remove a content switching virtual server only when you no longer require the virtual server. When you remove a content switching virtual server, the NetScaler appliance first unbinds all policies from the content switching virtual server, and then removes it.

To remove a content switching virtual server by using the command line interface

At the command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rm cs vserver <name>

Example

```
rm cs vserver Vserver-CS-1
```

To remove a content switching virtual server by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, select a virtual server, and click Delete.

Disabling and Re-Enabling Content Switching Virtual Servers

Content switching virtual servers are enabled by default when you create them. You can disable a content switching virtual server for maintenance. If you disable the content switching virtual server, the state of the content switching virtual server changes to Out of Service. While out of service, the content switching virtual server does not respond to requests.

To disable or re-enable a virtual server by using the command line interface

At the command prompt, type one of the following commands:

- disable cs vserver <name>
- enable cs vserver <name>

Example

```
disable cs vserver Vserver-CS-1

enable cs vserver Vserver-CS-1
```

To disable or re-enable a virtual server by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, select a virtual server and, in the Action list, select Enable or Disable.

Renaming Content Switching Virtual Servers

You can rename a content switching virtual server without unbinding it. The new name is propagated automatically to all affected parts of the NetScaler configuration.

To rename a virtual server by using the command line interface

At the command prompt, type:

rename cs vserver <name> <newName>

Example

```
rename cs vserver Vserver-CS-1 Vserver-CS-2
```

To rename a virtual server by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, select a virtual server and, in the Action list, select Rename.

Managing Content Switching Policies

You can modify an existing policy by configuring rules or changing the URL of the policy, or you can remove a policy. You can also rename an existing advanced

© 1999-2017 Citrix Systems, Inc. All rights reserved.

content switching policy. You can create different policies based on the URL. URL-based policies can be of different types, as described in the following table.

Table 1. Examples of URL-Based Policies

Type of URL-Based Policy	Specifies
Domain and Exact URL	Requests must match the configured domain name and configured URL (an exact prefix match if only the prefix is configured; or an exact match of the prefix and suffix if both the prefix and suffix are configured). **Example:** **add cs policy Policy-CS-1 -url /sports/tennis/index.html -domain "www.domainxyz.com"**
Domain and Wild Card URL	Requests must match the exact domain name and a partial prefix of the configured URL. **Example:** **add cs policy Policy-CS-1 -url /*.jsp -domain "www.domainxyz.com"**
Domain Only	Requests need match only the configured domain name. **Example:** **add cs policy Policy-CS-1 -domain "www.domainxyz.com"**
The Exact URL	The incoming URL must exactly match the URL specified by the policy. If only a URL prefix rule is configured, there must be an exact prefix match with the incoming URL. If a URL prefix and suffix-based rule is configured, there should be an exact match of the prefix and suffix with the incoming URL. **Example:** **add cs policy Policy-CS-1 -url /sports/tennis/index.html**
Prefix Only (Wild Card URL)	All the incoming URLs must start with the configured prefix. **Example:** **add cs policy Policy-CS-1 -url /sports*** sports/*" matches all URLs under /sports "/sports*" matches all URLs whose prefix match "/sports" starting from the beginning of a URL
Suffix Only (Wild Card URL)	All incoming URLs must end with the configured URL suffix. **Example:** **add cs policy Policy-CS-1 -url /*.jsp**
"/*.jsp" matches all URLs whose file extension is "jsp"	All incoming URLs must start with the configured prefix and end with the configured suffix. **Example:**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Type of URL-Based Policy Prefix and Suffix (Wild Card URL)	**add cs policy Policy-CS-1 -url /sports/*.jsp Specifies**

Note: You can configure rule-based content switching using classical policy expressions or advanced policy expressions.

To modify, remove, or rename a policy by using the command line interface

At the command prompt, type one of the following commands:

- set cs policy <policyName> [-domain <domainValue>] [-rule <ruleValue>] [-url <URLValue>]
- rm cs policy <policyName>
- rename cs policy <policyName> <newPolicyName>

Example

set cs policy Policy-CS-1 -domain "www.domainxyz.com"

set cs policy Policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(22).EQ(10.100.148.0)"

set cs policy Policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2010 Jun,GMT 2010 Jul)"

set cs policy Policy-CS-1 -url /sports/*

rename cs policy Policy-CS-1 Policy-CS-11

rm cs policy Policy-CS-1

To modify, remove, or rename a policy by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Policies.
2. Select the policy, and either delete it, edit it or, in the Action list, click Rename.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Client Connections

May 21, 2015

To ensure efficient management of client connections, you can configure the content switching virtual servers on the NetScaler appliance to use the following features:

- Redirecting client requests to a cache
- Enabling delayed cleanup of virtual server connections
- Rewriting ports and protocols for redirection
- Inserting the IP address and port of a virtual server in the request header
- Setting a time-out value for idle client connections
- Identifying Connections with the 4-tuple and Layer 2 Connection Parameters
- Configuring the ICMP Response. You can configure the NetScaler to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP VSERVER RESPONSE.
  The following settings can be made on a virtual server:
  - When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, NetScaler always responds.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds even if one virtual server is UP.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds even if one virtual server set to ACTIVE is UP.

Redirecting Client Requests to a Cache

The NetScaler cache redirection feature redirects HTTP requests to a cache. You can significantly reduce the burden of responding to HTTP requests and improve your Web site performance through proper implementation of the cache redirection feature.

A cache stores frequently requested HTTP content. When you configure cache redirection on a virtual server, the NetScaler appliance sends cacheable HTTP requests to the cache and non-cacheable HTTP requests to the origin Web server. For more information on cache redirection, see "Cache Redirection."

To configure cache redirection on a virtual server by using the command line interface

At the command prompt, type:

set cs vserver <name> -cacheable <Value>

Example

```
set cs vserver Vserver-CS-1 -cacheable yes
```

To configure cache redirection on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Cacheable.

Enabling Delayed Cleanup of Virtual Server Connections

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Under certain conditions, you can configure the down state flush setting to terminate existing connections when a service or a virtual server is marked DOWN. Terminating existing connections frees resources and in certain cases speeds recovery of overloaded load balancing setups.

To configure the down state flush setting on a virtual server by using the command line interface

At the command prompt, type:

set cs vserver <name> -downStateFlush <Value>

Example

```
set cs vserver Vserver-CS-1 -downStateFlush enabled
```

To configure the down state flush setting on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and then select Down State Flush.

Rewriting Ports and Protocols for Redirection

Virtual servers and the services that are bound to them may use different ports. When a service responds to an HTTP connection with a redirect, you may need to configure the NetScaler appliance to modify the port and the protocol to ensure that the redirection goes through successfully. You do this by enabling and configuring the redirectPortRewrite setting.

To configure HTTP redirection on a virtual server by using the command line interface

At the command prompt, type:

set cs vserver <name> -redirectPortRewrite <Value>

Example

```
set cs vserver Vserver-CS-1 -redirectPortRewrite enabled
```

To configure HTTP redirection on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Rewrite.

Inserting the IP Address and Port of a Virtual Server in the Request Header

If you have multiple virtual servers that communicate with different applications on the same service, you must configure the NetScaler appliance to add the IP address and port number of the appropriate virtual server to the HTTP requests that are sent to that service. This setting allows applications running on the service to identify the virtual server that sent the request.

If the primary virtual server is down and the backup virtual server is up, the configuration settings of the backup virtual server are added to the client requests. If you want the same header tag to be added, regardless of whether the requests are

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

from the primary virtual server or backup virtual server, you must configure the required header tag on both virtual servers.

Note: This option is not supported for wildcarded virtual servers or dummy virtual servers.

To insert the IP address and port of the virtual server in the client requests by using the command line interface

At the command prompt, type:

set cs vserver <name> -insertVserverIPPort <vServerIPPORT>

Example

```
set cs vserver Vserver-CS-1 -insertVserverIPPort 10.201.25.136:80
```

To insert the IP address and port of the virtual server in the client requests by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings and, in the Virtual Server IP Port Insertion list, select VIPADDR or V6TOV4MAPPING, and specify a port header in Virtual Server IP Port Insertion Value.

Setting a Time-out Value for Idle Client Connections

You can configure a virtual server to terminate any idle client connections after a configured time-out period elapses. When you configure this setting, the NetScaler appliance waits for the time you specify and, if the client is idle after that time, it closes the client connection.

To set a time-out value for idle client connections by using the command line interface

At the command prompt, type:

set cs vserver <name> -cltTimeout <Value>

Example

```
set cs vserver Vserver-CS-1 -cltTimeout 100
```

To set a time-out value for idle client connections by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and specify a Client Idle Time-Out value.

Identifying Connections with the 4-tuple and Layer 2 Connection Parameters

You can now set the L2Conn option for a content switching virtual server. With the L2Conn option set, connections to the content switching virtual server are identified by the combination of the 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) and Layer 2 connection parameters. The Layer 2 connection parameters are the MAC address, VLAN ID, and channel ID.

To set the L2Conn option for a content switching virtual server by using the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

command line interface

At the command line, type the following commands to configure the L2Conn parameter for a content switching virtual server and verify the configuration:

- set cs vserver <name> -l2Conn (**ON** | **OFF**)
- show cs vserver <name>

Example

```
> set cs vserver mycsvserver -l2Conn ON
Done
> show cs vserver mycsvserver
    mycsvserver (192.0.2.56:80) - HTTP   Type: CONTENT
    State: UP
        . . .
        . . .
    L2Conn: ON Case Sensitivity: ON
        . . .
        . . .
Done
>
```

To set the L2Conn option for a content switching virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and then select Layer 2 Parameters.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Jul 22, 2013

If the content switching feature does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for Troubleshooting Content Switching

For best results, use the following resources to troubleshoot a content switching issue on a NetScaler appliance:

- Configuration file
- Relevant newnslog file
- Trace files
- Network topology diagram for the network setup of the customer
- Citrix documentation, such as release notes, Knowledge Center articles, and eDocs

In addition to the above resources, the following tools expedite troubleshooting:

- The iehttpheaders or a similar utility
- The Wireshark application customized for the NetScaler trace files
- An SSH utility for command line access
- A HyperTerminal utility to access the console

Troubleshooting Content Switching Issues

The most common content switching issues involve the content switching feature not working at all, or working only intermittently, and Service Unavailable responses.

- **Issue**
  The content switching feature is not functioning.

  Resolution

  Check the configuration as follows:
  - Verify that the appliance is licensed for content switching.
  - Verify that the feature is enabled.
  - From the configuration file, verify that valid content switching policies are correctly bound to the load balancing virtual servers.
- **Issue**
  Client receives a 503 - Service Unavailable response.

  Resolution
  - Verify the URL and policy bindings. The client receives the 503 response when none of the policies you have configured is evaluated and no default load balancing virtual server is defined and bound to the content switching virtual server.
  - From the configuration, verify the policies and URL being accessed by the client.
  - Verify that for every type of request the respective policy is evaluated. If the policy is not evaluated, check the policy expression and update it if necessary.
  - Verify the URL and HTTP request and response headers. To do so, record an HTTPHeader trace and, if necessary, record the packet traces on the appliance and the client.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Issue**
  Intermittently, the content switching feature is not working as expected.

  Resolution
  - Study the network topology diagram, if available, of the setup to understand the various devices installed between the client and the server(s).
  - Verify the configuration and policy bindings. Make sure that the URL in the policy expression matches to the one in the client request.
  - Verify that appropriate priorities are assigned to the policies. An incorrect precedence or priority assigned to a policy can cause a problem.
  - Run the following commands to verify the bindings and the values of the policy hit counters in the output of the commands:
    show cs vserver <CS VServer>

    show cs policy <CS Policy>

    stat cs vserver <CS VServer>

  - Using iehttpheaders or a similar utility, determine whether the HTTP headers for the requests or responses provide any pointers to the issue.
  - Check the release notes and Knowledge Center articles.
  - If the issue is still not resolved, contact Citrix Technical Support with appropriate data for further investigation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

DataStream

May 25, 2015

The NetScaler DataStream feature provides an intelligent mechanism for request switching at the database layer by distributing requests based on the SQL query being sent.

When deployed in front of database servers, a NetScaler ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and on the basis of database names, usernames, character sets, and packet size.

You can either configure load balancing to switch requests based on load balancing algorithms or elaborate the switching criteria by configuring content switching to make a decision based on an SQL query parameters. You can further configure monitors to track the state of database servers.

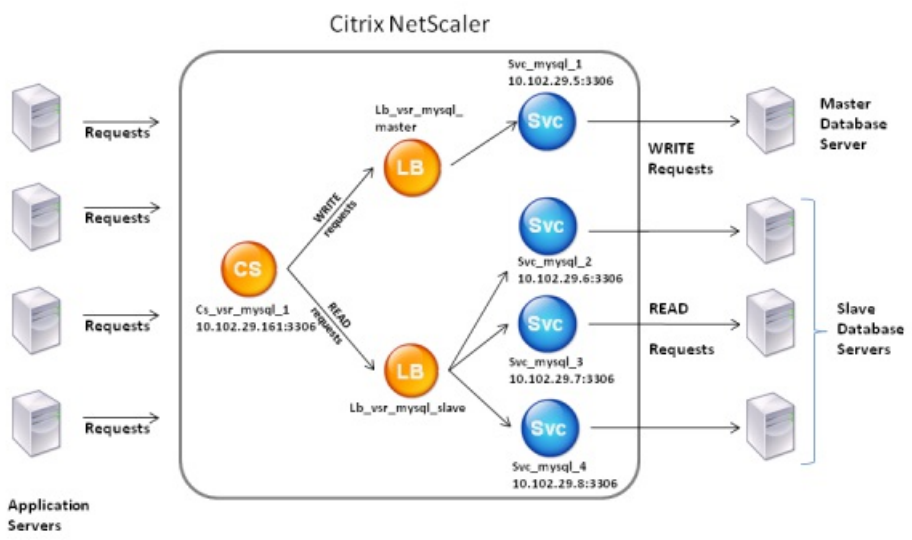
Note: NetScaler DataStream is supported only for MySQL and MS SQL databases. For information about the supported protocol version, character sets, special queries, and transactions, see DataStream Reference.

How NetScaler DataStream Works

In DataStream, the NetScaler is placed in-line between the application and/or Web servers and the database servers. On the NetScaler appliance, the database servers are represented by services.

A typical DataStream deployment consists of the entities described in the following diagram.

Figure 1. *DataStream Entity Model*



As shown in this figure, a DataStream configuration can consist of an optional content switching virtual server (CS), a load balancing setup consisting of load balancing virtual servers (LB1 and LB2) and services (Svc1, Svc2, Svc3, and Svc4), and content switching policies (optional).

The clients (application or Web servers) send requests to the IP address of a content switching virtual server (CS) configured

© 1999-2017 Citrix Systems, Inc. All rights reserved.

on the NetScaler appliance. The NetScaler, then, authenticates the clients using the database user credentials configured on the NetScaler appliance. The content switching virtual server (CS) applies the associated content switching policies to the requests. After evaluating the policies, the content switching virtual server (CS) routes the requests to the appropriate load balancing virtual server (LB1 or LB2), which, then, distributes the requests to the appropriate database servers (represented by services on the NetScaler) based on the load balancing algorithm. The NetScaler uses the same database user credentials to authenticate the connection with the database server.

If a content switching virtual server is not configured on the NetScaler, the clients (application or Web servers) send their requests to the IP address of a load balancing virtual server configured on the NetScaler appliance. The NetScaler authenticates the client by using the database user credentials configured on the NetScaler appliance, and then uses the same credentials to authenticate the connection with the database server. The load balancing virtual server distributes the requests to the database servers according to the load balancing algorithm. The most effective load balancing algorithm for database switching is the least connection method.

DataStream uses connection multiplexing to enable multiple client-side requests to be made over the same server-side connection. The following connection properties are considered:

- User name
- Database name
- Packet size
- Character set

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Database Users

Dec 28, 2016

In databases, a connection is always stateful, which means that as soon as a connection is established, it must be authenticated.

You need to configure your database user name and password on the NetScaler ADC. For example, if you have a user John configured on the database, you need to configure the user John on the ADC too. When you add the database user names and passwords on the ADC, these are added to the nsconfig file.
Note: Names are case sensitive.
The ADC uses these user credentials to authenticate the clients, and then authenticate the server connections with the database servers.

To add a database user by using the command line interface

At the command prompt, type

add db user <username> - password <password>
Example

```
> add db user nsdbuser -password dd260427edf
```

To add a database user by using the configuration utility

Navigate to System > User Administration > Database Users, and configure a database user.

If you have changed the password of the database user on the database server, you must reset the password of the corresponding user configured on the NetScaler.

To reset the password of a database user by using the command line interface

At the command prompt, type

set db user <username> -password <password>
Example

```
> set db user nsdbuser -password dd260538abs
```

To reset the password of database users by using the configuration utility

Navigate to System > User Administration > Database Users, select a user, and enter new values for the password.

If a database user no longer exists on the database server, you can remove the user from the NetScaler. However, if the user continues to exist on the database server and you remove the user from the NetScaler, any request from the client with this user name does not get authenticated, and therefore, does not get routed to the database server.

To remove a database user by using the command line interface

At the command prompt, type

rm db user <username>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

```
> rm db user nsdbuser
```

To remove a database user by using the configuration utility

Navigate to System > User Administration > Database Users, select a user, and click Delete.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Database Profile

Sep 03, 2014

A database profile is a named collection of parameters that is configured once but applied to multiple virtual servers that require those particular parameter settings. After creating a database profile, you bind it to load balancing or content switching virtual servers. You can create as many profiles as you need.

To create a database profile by using the command line interface

At the command line, type the following commands to create a database profile and verify the configuration:

- add db dbProfile <name> [-interpretQuery (YES | NO)] [-stickiness (YES | NO)] [-kcdAccount <string>]
- show db dbProfile

Example

```
> add dbProfile myDBProfile -interpretQuery YES -stickiness YES -kcdAccount mykcdaccnt
 Done
> show dbProfile myDBProfile
 Name: myDBProfile
 Interpret Query: YES
 Stickyness: YES
 KCD Account: mykcdaccnt
 Reference count: 0

 Done
>
```

To create a database profile by using the configuration utility

Navigate to System > Profiles and, on the Database Profiles tab, configure a database profile.

To bind a database profile to a load balancing or content switching virtual server by using the command line interface

At the command line, type:

set (lb | cs) vserver <name> -dbProfileName <string>

To bind a database profile to a load balancing or content switching virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers or Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Profiles and, in the DB Profile list, select a profile to bind to the virtual server. To create a new profile, click plus (+).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Load Balancing for DataStream

May 15, 2015

Before configuring a load balancing setup, you must enable the load balancing feature. Then, begin by creating at least one service for each database server in the load balancing group. With the services configured, you are ready to create a load balancing virtual server and bind the services to the virtual server.

Parameter values specific to DataStream

Protocol

Use the MYSQL protocol type for MySQL databases and MSSQL protocol type for MS SQL databases while configuring virtual servers and services. The MySQL and TDS protocols are used by the clients to communicate with the respective database servers by using SQL queries. For information about the MySQL protocol, see http://dev.mysql.com/doc/internals/en/client-server-protocol.html. For information about the TDS protocol, see http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx.

Port

Port on which the virtual server listens for client connections. Use port 3306 for MySQL database servers.

Method

It is recommended that you use the Least Connection method for better load balancing and lower server load. However, other methods, such as Round Robin, Least Response Time, Source IP Hash, Source IP Destination IP Hash, Least Bandwidth, Least Packets, and Source IP Source Port Hash, are also supported.

Note: URL Hash method is not supported for DataStream.

MS SQL Server Version

If you are using the Microsoft SQL Server, and you expect some clients to not be running the same version as your Microsoft SQL Server product, set the Server Version parameter for the load balancing virtual server. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version. For more information about setting the Server Version parameter, see Configuring the MySQL and Microsoft SQL Server Version Setting.

MySQL Server Version

If you are using the MySQL Server, and you expect some clients to not be running the same version as your MySQL Server product, set the Server Version parameter for the load balancing virtual server. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version. For more information about setting the Server Version parameter, see Configuring the MySQL and Microsoft SQL Server Version Setting.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Content Switching for DataStream

May 15, 2015

You can segment traffic according to information in the SQL query, on the basis of database names, user names, character sets, and packet size.

You can configure content switching policies with default syntax expressions to switch content based on connection properties, such as user name and database name, command parameters, and the SQL query to select the server.

The default syntax expressions evaluate traffic associated with MYSQL and MS SQL database servers. You can use request-based expressions in default syntax policies to make request switching decisions at the content switching virtual server bind point and response-based expressions (expressions that begin with MYSQL.RES) to evaluate server responses to user-configured health monitors.

Note: For information about default syntax expressions, see Default Syntax Expressions: DataStream.

Parameter values specific to DataStream

Protocol

Use the MYSQL protocol type for MySQL databases and MSSQL protocol type for MS SQL databases while configuring virtual servers and services. The MySQL and TDS protocols are used by the clients to communicate with the respective database servers by using SQL queries. For information about the MySQL protocol, see http://dev.mysql.com/doc/internals/en/client-server-protocol.html. For information about the TDS protocol, see http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx.

Port

Port on which the virtual server listens for client connections. Use port 3306 for MySQL database servers.

MS SQL Server Version

If you are using Microsoft SQL Server, and you expect some clients to not be running the same version as your Microsoft SQL Server product, set the Server Version parameter for the content switching virtual server. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version. For more information about setting the Server Version parameter, see Configuring the Microsoft SQL Server Version Setting.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Monitors for DataStream

Oct 10, 2016

To track the state of each load balanced database server in real time, you need to bind a monitor to each service. The monitor is configured to test the service by sending periodic probes to the service. (This is sometimes referred to as performing a health check.) If the monitor receives a timely response to its probes, it marks the service as UP. If it does not receive a timely response to the designated number of probes, it marks the service as DOWN.

For DataStream, you need to use the built-in monitors, MYSQL-ECV and MSSQL-ECV. This monitor provides the ability to send an SQL request and parse the response for a string.

Before configuring monitors for DataStream, you must add database user credentials to your NetScaler. For information about configuring monitors, see Monitors.

When you create a monitor, a TCP connection is established with the database server, and the connection is authenticated by using the user name provided while creating the monitor. You can then run an SQL query to the database server and evaluate the server response to check whether it matches the configured rule.

The following examples are for MYSQL servers.

Examples

In the following example, the value of the error message is evaluated to determine the state of the server.

```
add lb monitor lb_mon1 MYSQL-ECV -sqlQuery "select * from
table2;" -evalrule "mysql.res.error.message.contains(\"Invalid
User\")"-database "NS" -userName "user1"
```

In the following example, the number of rows in the response is evaluated to determine the state of the server.

```
add lb monitor lb_mon4 MYSQL-ECV -sqlQuery "select * from
table4;" -evalrule "mysql.res.atleast_rows_count(7)" -database "NS" -userName "user2"
```

In the following example, the value of a particular column is evaluated to determine the state of the server.

```
add lb monitor lb_mon3 MYSQL-ECV
-sqlQuery "select * from ABC;" -evalrule "mysql.res.row(1).double_elem(2) == 345.12"
-database "NS" -userName "user3"
```

The following examples are for MSSQL servers.

Examples

In the following example, the value of the error message is evaluated to determine the state of the server.

```
add lb monitor lb_mon1 MSSQL_ECV -sqlQuery "select * from
table2;" -evalrule "mssql.res.error.message.contains(\"Invalid
User\")"-database "NS" -userName "user1"
```

In the following example, the number of rows in the response is evaluated to determine the state of the server.

```
add lb monitor lb_mon4 MSSQL_ECV -sqlQuery "select * from
table4;" -evalrule "mssql.res.atleast_rows_count(7)" -database "NS" -userName "user2"
```

In the following example, the value of a particular column is evaluated to determine the state of the server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add lb monitor lb_mon3 MSSQL_ECV
-sqlQuery "select * from ABC;" -evalrule "mssql.res.row(1).double_elem(2) == 345.12"
-database "NS" -userName "user3"
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 1: Configuring DataStream for a Master/Slave Database Architecture

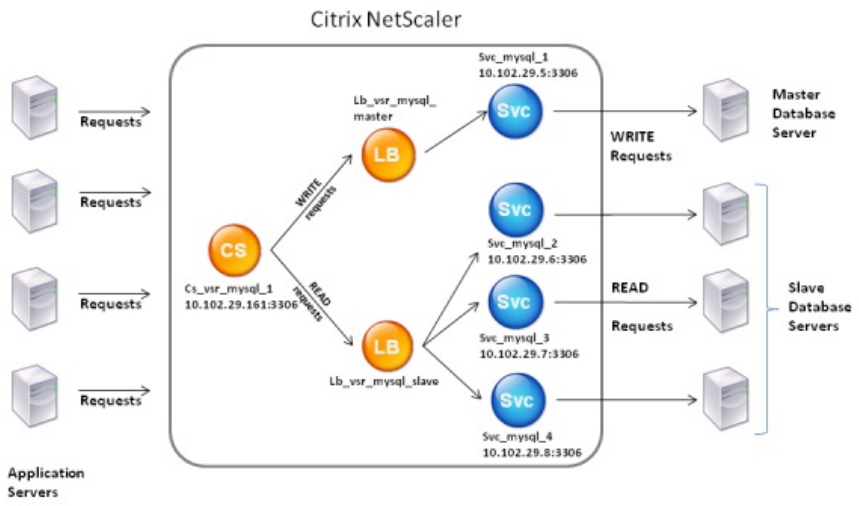
May 15, 2015

A commonly used deployment scenario is the master/slave database architecture where the master database replicates all information to the slave databases.

For master/slave database architecture, you may want all WRITE requests to be sent to the master database and all READ requests to the slave databases.

The following figure shows the entities and the values of the parameters you need to configure on the appliance.

Figure 1. *DataStream Entity Model for Master/Slave Database Setup*



In this example scenario, a service (Svc_mysql_1) is created to represent the master database and is bound to a load balancing virtual server (Lb_vsr_mysql_master). Three more services (Svc_mysql_2, Svc_mysql_3, and Svc_mysql_4) are created to represent the three slave databases, and they are bound to another load balancing virtual server (Lb_vsr_mysql_slave).

A content switching virtual server (Cs_vsr_mysql_1) is configured with associated policies to send all WRITE requests to the load balancing virtual server, Lb_vsr_mysql_master, and all READ requests to the load balancing virtual server, Lb_vsr_mysql_slave.

When a request reaches the content switching virtual server, the virtual server applies the associated content switching policies to that request. After evaluating the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

The following table lists the names and values of the entities and the policy configured on the NetScaler.

Table 1. ***Entity and Policy Names and Values***

Entity Type	Name	IP Address	Protocol	Port	Expression

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity Type	Name	IP Address	Protocol	Port	Expression
Services	Svc_mysql_1	10.102.29.5	MYSQL	3306	NA
	Svc_mysql_2	10.102.29.6	MYSQL	3306	NA
	Svc_mysql_3	10.102.29.7	MYSQL	3306	NA
	Svc_mysql_4	10.102.29.8	MYSQL	3306	NA
Load balancing virtual servers	Lb_vsr_mysql_master	10.102.29.201	MYSQL	3306	NA
	Lb_vsr_mysql_slave	10.102.29.202	MYSQL	3306	NA
Content switching virtual server	Cs_vsr_mysql_1	10.102.29.161	MYSQL	3306	NA
Content switching policy	Cs_select	NA	NA	NA	"MYSQL.REQ.QUERY.COMMAND.contains(\"select\")"

To configure DataStream for a master/slave database setup by using the command line interface

At the command prompt, type

- add service Svc_mysql_1 10.102.29.5 mysql 3306
- add service Svc_mysql_2 10.102.29.6 mysql 3306
- add service Svc_mysql_3 10.102.29.7 mysql 3306
- add service Svc_mysql_4 10.102.29.8 mysql 3306
- add lb vserver Lb_vsr_mysql_master mysql 10.102.29.201 3306
- add lb vserver Lb_vsr_mysql_slave mysql 10.102.29.202 3306
- bind lb vserver Lb_vsr_mysql_master svc_mysql_1
- bind lb vserver Lb_vsr_mysql_slave svc_mysql_2
- bind lb vserver Lb_vsr_mysql_slave svc_mysql_3
- bind lb vserver Lb_vsr_mysql_slave svc_mysql_4
- add cs vserver Cs_vsr_mysql_1 mysql 10.102.29.161 3306
- add cs policy Cs_select –rule "MYSQL.REQ.QUERY.COMMAND.contains(\"select\")"
- bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_master
- bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_slave –policy Cs_select –priority 10

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 2: Configuring the Token Method of Load Balancing for DataStream

May 15, 2015

You can configure the token method of load balancing for DataStream to base the selection of database servers on the value of the token extracted from the client (application or web server) requests. These tokens are defined by using SQL expressions. For subsequent requests with the same token, the NetScaler sends the requests to the same database server that handled the initial request. Requests with the same token are sent to the same database server until the maximum connection limit is reached or the session entry has aged out.

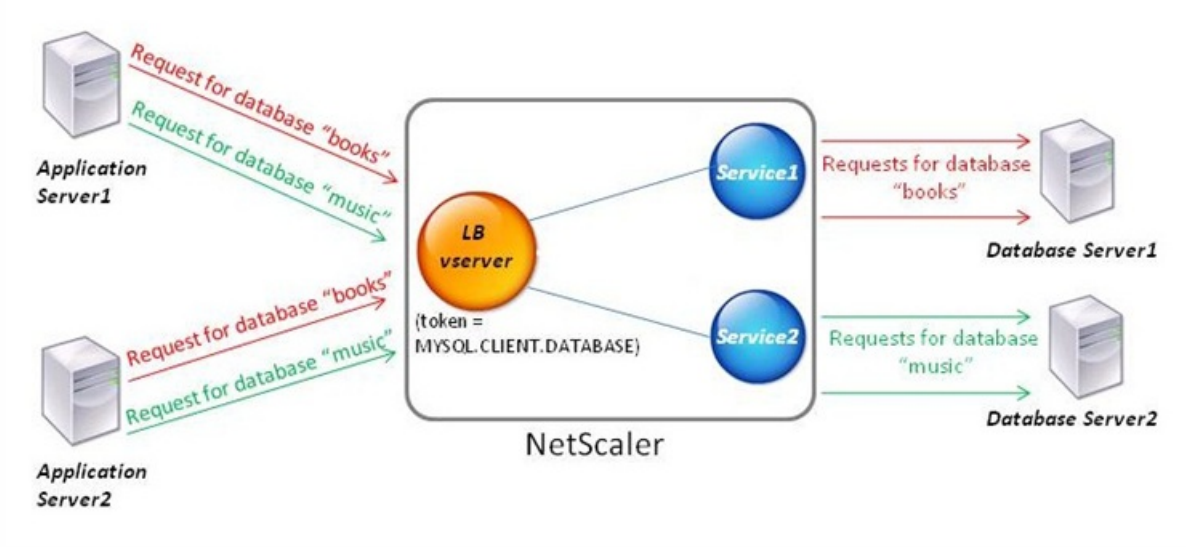
You can use the following sample SQL expressions to define tokens:

MySQL	**MS SQL**
MYSQL.REQ.QUERY.TEXT	MSSQL.REQ.QUERY.TEXT
MYSQL.REQ.QUERY.TEXT(n)	MSSQL. REQ.QUERY.TEXT(n)
MYSQL.REQ.QUERY.COMMAND	MSSQL.REQ.QUERY.COMMAND
MYSQL.CLIENT.USER	MSSQL.CLIENT.USER
MYSQL.CLIENT.DATABASE	MSSQL.CLIENT.DATABASE
MYSQL.CLIENT.CAPABILITIES	

The following example shows how the NetScaler DataStream feature works when you configure the token method of load balancing.

Figure 1. How DataStream Works with the Token Method of Load Balancing

© 1999-2017 Citrix Systems, Inc. All rights reserved.



In this example, the token is the name of the database. A request with token books is sent to Database Server1 and a request with token music is sent to Database Server2. All subsequent requests with token books are sent to Database Server1 and requests with token music are sent to Database Server2. This configuration provides pseudo persistence with the database servers.

To configure this example by using the command line interface

At the command prompt, type:

- add service Service1 192.0.2.9 MYSQL 3306
- add service Service2 192.0.2.11 MYSQL 3306
- add lb vserver token_lb_vserver MYSQL 192.0.2.15 3306 -lbmethod token -rule MYSQL.CLIENT.DATABASE
- bind lb vserver token_lb_vserver Service1
- bind lb vserver token_lb_vserver Service2

To configure this example by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, configure a virtual server and specify the protocol as MYSQL.
2. Click in the Service section, and configure two services specifying the protocol as MYSQL. Bind these services to the virtual server.
3. In Advanced Settings, click Method and, in the Load Balancing Method list, select TOKEN and specify the expression as MYSQL.CLIENT.DATABASE.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 3: Logging MSSQL Transactions in Transparent Mode

Jun 09, 2015

You can configure the NetScaler appliance to operate transparently between MSSQL clients and servers, and to only log or analyze details of all client-server transactions. Transparent mode is designed so that the NetScaler appliance only forwards MSSQL requests to the server, and then relays the server's responses to the clients. As the requests and responses pass through the appliance, the appliance logs information gathered from them, as specified by the audit logging or AppFlow configuration, or collects statistics, as specified by the Action Analytics configuration. You do not have to add database users to the appliance.

When operating in transparent mode, the NetScaler appliance does not perform load balancing, content switching, or connection multiplexing for the requests. However, it responds to a client's pre-login packet on behalf of the server so that it can prevent encryption from being agreed upon during the pre-login handshake. The login packet and subsequent packets are forwarded to the server.

This section includes the following details:

- Summary of Configuration Tasks
- Configuring Transparent Mode by Using a Wildcard Virtual Server
- Configuring Transparent Mode by Using MSSQL Services

Summary of Configuration Tasks

For logging or analyzing MSSQL requests in transparent mode, you have to do the following:

- Configure the NetScaler appliance as the default gateway for both clients and servers.
- Do one of the following on the NetScaler appliance:
  - **If you can configure the use source IP address (USIP) option globally,** create a load balancing virtual server with a wildcard IP address and the port number on which the MSSQL servers listen for requests (a port-specific wildcard virtual server). Then, enable the USIP option globally. If you configure a port-specific wildcard virtual server, you do not have to create MSSQL services on the appliance. The appliance discovers the services on the basis of the destination IP address in the client requests. For instructions, see Configuring Transparent Mode by Using a Wildcard Virtual Server.
  - **If you do not want to configure the USIP option globally,** create MSSQL services with the USIP option enabled on each of them. If you configure services, you do not have to create a port-specific wildcard virtual server. For instructions, see Configuring Transparent Mode by Using MSSQL Services.
- Configure audit logging, AppFlow, or Action Analytics to log or collect statistics about the requests. If you configure a virtual server, you can bind your policies either to the virtual server or to the global bind point. If you do not configure a virtual server, you can bind your policies to only the global bind point.

Configuring Transparent Mode by Using a Wildcard Virtual Server

You can configure transparent mode by configuring a port-specific wildcard virtual server and enabling Use Source IP (USIP) mode globally. When a client sends its default gateway (the NetScaler appliance) a request with the IP address of an MSSQL server in the destination IP address header, the appliance checks whether the destination IP address is available. If the IP address is available, the virtual server forwards the request to the server. Otherwise, it drops the request.

To create a wildcard virtual server by using the command line

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to create a wildcard virtual server and verify the configuration:

1. add lb vserver <name> <serviceType> <IPAddress> <port>
2. show lb vserver <name>

Example

```
> add lb vserver wildcardLbVs MSSQL * 1433
 Done
> show lb vserver wildcardLbVs
 wildcardLbVs (*:1433) - MSSQL Type: ADDRESS
 State: UP
 . . .

 Done
>
```

To create a wildcard virtual server by using the NetScaler configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers, and create a virtual server. Specify MSSQL as the protocol and * as the IP address.

To enable Use Source IP (USIP) mode globally by using the command line

At the command prompt, type the following commands to enable USIP mode globally and verify the configuration:

- enable ns mode USIP
- show ns mode

Example

```
> enable ns mode USIP
 Done
> show ns mode

 Mode                    Acronym                Status
 -------                 -------                ------
          . . .
 3) Use Source IP        USIP                   ON
          . . .
 Done
>
```

To enable USIP mode globally by using the NetScaler configuration utility

1. Navigate to System > Settings and, in Modes and Features, select Configure Modes.
2. Select Use Source IP.

Configuring Transparent Mode by Using MSSQL Services

You can configure transparent mode by configuring MSSQL services and enabling USIP on each service. When a client sends its default gateway (the NetScaler appliance) a request with the IP address of an MSSQL server in the destination IP address header, the appliance forwards the request to the destination server.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To create an MSSQL service and enable USIP mode on the service by using the command line interface

At the command prompt, type the following commands to create an MSSQL service, with USIP enabled, and verify the configuration:

- add service <name> (<IP> | <serverName>) <serviceType> <port> -usip YES
- show service <name>

Example

```
> add service myDBservice 192.0.2.0 MSSQL 1433 -usip YES
 Done
> show service myDBservice
 myDBservice (192.0.2.0:1433) - MSSQL
 State: UP
            . . .
 Use Source IP: YES  Use Proxy Port: YES
            . . .
 Done
>
```

To create an MSSQL service, with USIP enabled, by using the NetScaler configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and configure a service.
2. Specify protocol as MSSQL and, in Settings, select Use Source IP.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 4: Database Specific Load Balancing

Nov 28, 2016

A database server farm should be load balanced not only on the basis of the states of the servers, but also on the basis of the availability of the database on each server. A service might be up, and a load balancing device might show it as being in the UP state, but the requested database might be unavailable on that service. If a query is forwarded to a service on which the database is unavailable, the request is not served. Therefore, a load balancing device must be aware of the availability of a database on each service and, when making a load balancing decision, it must consider only those services on which the database is available.

As an example, consider that database servers server1, server2, and server3 host databases mydatabase1 and mydatabase2. If mydatabase1 becomes unavailable on server2, the load balancing device must be aware of that change in state, and it must load balance requests for mydatabase1 across only server1 and server3. After mydatabase1 becomes available on server2, the load balancing device must include server2 in load balancing decisions. Similarly, if mydatabase2 becomes unavailable on server3, the device must load balance requests for mydatabase2 across only server1 and server2, and it must include server3 in its load balancing decisions only when mydatabase2 becomes available. This load balancing behavior must be consistent across all the databases that are hosted on the server farm.

The Citrix NetScaler appliance implements this behavior by retrieving a list of all the databases that are active on a service. To retrieve the list of active databases, the appliance uses a monitor that is configured with an appropriate SQL query. If the requested database is unavailable on a service, the appliance excludes the service from load balancing decisions until it becomes available. This behavior ensures uninterrupted service to clients.

Note: Database specific load balancing is currently supported for only MSSQL and MySQL service types. This support is also available for Microsoft SQL Server 2012 high availability deployment.
To set up database specific load balancing, you must enable the load balancing feature, configure a load balancing virtual server of type MSSQL or MySQL, configure the services that host the database, and bind the services to the virtual server. The monitor needs valid user credentials to log on to the database server, so you must configure a database user account on each of the servers and then add the user account to the NetScaler appliance. Then, you configure an MSSQL-ECV or MYSQL-ECV monitor and bind the monitor to each service. Finally, you must test the configuration to ensure that it is working as intended. Before you perform these configuration tasks, make sure you understand how database specific load balancing works.

This section includes the following details:

- How Database Specific Load Balancing Works
- Enabling Load Balancing
- Configuring a Load Balancing Virtual Server for Database Specific Load Balancing
- Configuring Services
- Configuring Database Users
- Configuring a Monitor to Retrieve the Names of Active Databases
- Availability Groups Deployment Support for MSSQL

How Database Specific Load Balancing Works

For database specific load balancing, you configure a monitor that periodically queries each database server for the names of all the active databases on it. The Citrix NetScaler appliance stores the results, and regularly updates the records on the basis of the information retrieved through monitoring. When a client queries a particular database, the appliance uses the configured load balancing method to select a service, and then checks its records to determine whether the database is

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

available on that service. If the records indicate that the database is not available, it uses the configured load balancing method to select the next available service, and then repeats the check. The appliance forwards the query to the first available service on which the database is active.

Enabling Load Balancing

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

- enable ns feature LB
- show ns feature

Example

```
> enable ns feature LoadBalancing
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging            WL               OFF
 2)   Surge Protection       SP               ON
 3)   Load Balancing         LB               ON
 .
 .
 .
 24)  NetScaler Push         push             OFF
 Done
```

To enable load balancing by using the configuration utility

Navigate to System > Settings and, in Configure Basic Features, select Load Balancing.

Configuring a Load Balancing Virtual Server for Database Specific Load Balancing

To configure a virtual server to load balance databases on the basis of availability, you enable the database specific load balancing parameter on the virtual server. Enabling the parameter modifies the load balancing logic so that the NetScaler appliance refers the results of the monitoring probe sent to the selected service, before forwarding the query to that service.

> ### Note
>
> For configuration examples related to MSSQL or MySQL, refer to the following topics:
>
> - Configuration examples for MSSQL virtual server
> - Configuration examples for MySQL virtual server

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure a load balancing virtual server for database specific load balancing

At the command prompt, type the following command to configure a load balancing virtual server for database specific load balancing and verify the configuration:

- add lb vserver <name> <serviceType> <ipAddress> <port> -dbsLb ENABLED
- show lb vserver <name>

Configuring Services

After you enable the load balancing feature, you must create at least one service for each application server that is to be included in your load balancing setup. The services that you configure provide the connections between the NetScaler appliance and the load balanced servers. Each service has a name and specifies an IP address, a port, and the type of data that is served.

If you create a service without first creating a server object, the IP address of the service is also the name of the server that hosts the service. If you prefer to identify servers by name rather than IP address, you can create server objects and then specify a server's name instead of its IP address when you create a service.

Configuring Database Users

In databases, a connection is always stateful, which means that as soon as a connection is established, it must be authenticated.

You need to configure your database user name and password on the NetScaler ADC. For example, if you have a user John configured on the database, you need to configure the user John on the ADC too. When you add the database user names and passwords on the ADC, these are added to the nsconfig file.
Note: Names are case sensitive.
The ADC uses these user credentials to authenticate the clients, and then authenticate the server connections with the database servers.

To add a database user by using the command line interface

At the command prompt, type

add db user <username> - password <password>
Example

```
> add db user nsdbuser -password dd260427edf
```

To add a database user by using the configuration utility

Navigate to **System** > **User Administration** > **Database Users**, and configure a database user.

If you have changed the password of the database user on the database server, you must reset the password of the corresponding user configured on the NetScaler.

To reset the password of a database user by using the command line interface

At the command prompt, type

© 1999-2017 Citrix Systems, Inc. All rights reserved.

set db user <username> -password <password>
Example

```
> set db user nsdbuser -password dd260538abs
```

To reset the password of database users by using the configuration utility

Navigate to System > User Administration > Database Users, select a user, and enter new values for the password.

If a database user no longer exists on the database server, you can remove the user from the NetScaler. However, if the user continues to exist on the database server and you remove the user from the NetScaler, any request from the client with this user name does not get authenticated, and therefore, does not get routed to the database server.

To remove a database user by using the command line interface

At the command prompt, type

rm db user <username>
Example

```
> rm db user nsdbuser
```

To remove a database user by using the configuration utility

Navigate to System > User Administration > Database Users, select a user, and click Delete.

Configuring a Monitor to Retrieve the Names of Active Databases

To retrieve a list of all the active databases on a database instance, you create a monitor that logs on to the database server by using a valid user credentials and runs an appropriate SQL query. The SQL query you need to use depends on your SQL server deployment. For example, in an MSSQL database mirroring setup, you can use the following query to retrieve a list of active databases available on a server instance.

select name from sys.databases where state=0
In a MySQL database setup you can use the following queries to retrieve a list of active databases available on a server instance.

show databases

You also configure the monitor to evaluate the response for an error condition, and to store the results if there is no error. If the response contains an error, the monitor marks the service as DOWN, and the appliance excludes the service from load balancing decisions until an error is no longer returned.

Note: The database specific load balancing feature is supported only for the MSSQL and MySQL service types. Therefore, the monitor type must be MSSQL-ECV or MYSQL-ECV.

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the command line

At the command prompt, type the following commands to retrieve the names of all the active databases hosted on a service and verify the configuration:

- **add lb monitor** <monitorName> <type> -userName <string> -sqlQuery <text> -evalRule <expression> -storedb **ENABLED**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

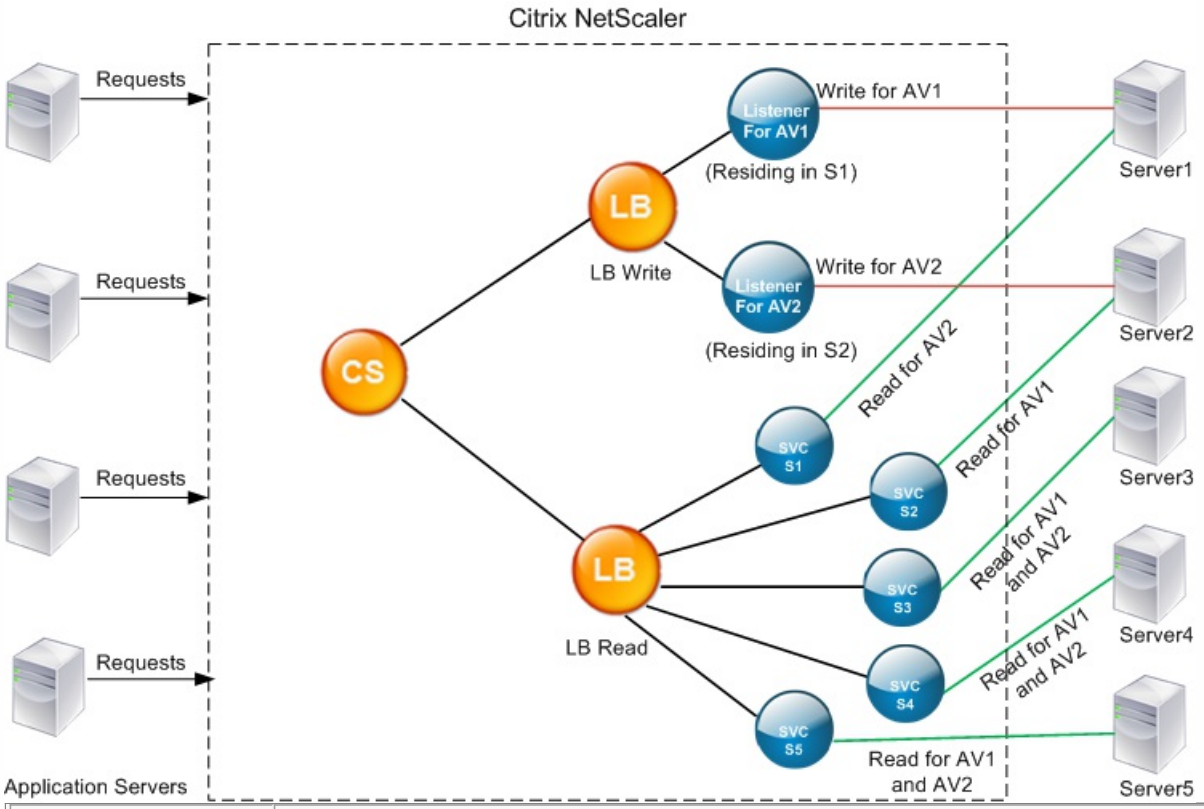
- **show lb monitor** <monitorName>

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Monitors** and configure a monitor of type MSSQL-ECV or MYSQL-ECV.
2. In **Special Parameters**, specify a user name, query, and a rule.
   For example, for MSSQL-ECV, the query should be "select name from sys.databases where state=0", and a rule should be MSSQL.RES.TYPE.NE(ERROR).
   for MYSQL-ECV, the query should be "show databases" and a rule should be MYSQL.RES.TYPE.NE(ERROR).

Availability Groups Deployment Support for MSSQL

Consider the following scenario in which database specific load balancing is configured in a high availability group deployment. S1 through S5 are the services on the NetScaler. DB1 through DB4 are the databases on the servers represented by the services S1 through S5. AV1 and AV2 are the availability groups. Each availability group contains up to one primary database server instance and up to four secondary database server instances. A service, representing the servers in the availability group, can be primary for one availability group and secondary for another availability group. Each availability group contains different databases and one listener, which is a service. All requests arrive on the listener service that resides on the primary database. AVI contains databases DB1 and DB2. AV2 contains databases DB3 and DB4. L1 and L2 are the listeners on AV1 and AV2 respectively. S1 is the primary service for AV1 and S2 is the primary service for AV2.



Service	List of Active Databases on the Service
S1	DB1, DB2, DB3, DB4
S2	DB3, DB4

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Service	List of Active Databases on the Service
S3	DB3, DB4
S4	DB1, DB2
S5	DB1, DB2

Availability Group	Databases	Services representing the Servers in Availability Group
AV1	DB1, DB2	S1, S4, S5
AV2	DB3, DB4	S1, S2, S3

Queries flow as follows:

1. A READ query for AV1 is load balanced between S4 and S5. S1 is the primary for AV1.
2. A WRITE query for AV1 is directed to L1.
3. A READ query for AV2 is load balanced between S1 and S3. S2 is the primary for AV2.
4. A WRITE query for AV1 is directed to L2.

Sample Configuration

1. Configure load balancing and content switching virtual servers.
   - add lb vserver lbwrite -dbslb enabled
   - add lbvserver lbread MSSQL -dbslb enabled
   - add csvserver csv MSSQL 1.1.1.10 1433
2. Configure two listener services, one for each availability group, and five services S1 through S5 representing databases DB1 through DB4.
   - add service L1 1.1.1.11 MSSQL 1433
   - add service L2 1.1.1.12 MSSQL 1433
   - add service s1 1.1.1.13 MSSQL 1433
   - add service s2 1.1.1.14 MSSQL 1433
   - add service s3 1.1.1.15 MSSQL 1433
   - add service s4 1.1.1.16 MSSQL 1433
   - add service s5 1.1.1.17 MSSQL 1433
3. Bind the services to the load balancing virtual servers.
   - bind lbvserver lbwrite L1
   - bind lbvserver lbwrite L2
   - bind lbvserver lbread s1
   - bind lbvserver lbread s2
   - bind lbvserver lbread s3
   - bind lbvserver lbread s4
   - bind lbvserver lbread s5
4. Configure database users.
   - add db user nsdbuser1 -password dd260427edf
   - add db user nsdbuser2 -password ccd1234xyzw

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Configure two monitors, monitor_L1 and monitor_L2 for each listener service, to retrieve the list of active databases in that availability group. Add a monitor, monitor1 to retrieve the list of databases for the secondary database server instance.
   - add lb monitor monitor_L1 MSSQL-ECV -userName user1 -sqlQuery "SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_states b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_listeners c on b.group_id = c.group_id INNER JOIN sys.availability_group_listener_ip_addresses d on c.listener_id = d.listener_id WHERE b.role = 1 and d.ip_address like '1.1.1.11'" -evalRule "MSSQL.RES.TYPE.NE(ERROR)" –storedb ENABLED
   - add lb monitor monitor_L2 MSSQL-ECV -userNameuser1 -sqlQuery "SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replicca_states b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_listeners c on b.group_id = c.group_id INNER JOIN sys.availability_group_listener_ip_addresses d on c.listener_id = d.listener_id WHERE b.role = 1 and d.ip_address like '1.1.1.12'" -evalRule "MSSQL.RES.TYPE.NE(ERROR)" -storedb ENABLED
   - add lb monitor monitor1 MSSQL-ECV -userNameuser1 -sqlQuery "SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_states b ON a.replica_id=b.replica_id WHERE b.role = 2" -evalRule "MSSQL.RES.TYPE.NE(ERROR)" -storedb ENABLED
6. Configure read and write policies.
   - add cs policy pol_write -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS(\"insert\")"
   - add cs policy pol_read -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS(\"select\")"
7. Bind the policies to the content switching virtual server.
   - bind csvserver csv -targetLBVserver lbwrite -policyName pol_write -priority 11
   - bind csvserver csv -targetLBVserver lbread -policyName pol_read -priority 12
8. Bind monitors to the services. Bind monitors to services L1 and L2 to get the list of active databases for the availability group for which it is the listener. Bind monitors to all the services that are bound to the read-only virtual server.
   - bind service L1 -monitorName monitor_L1
   - bind service L2 -monitorName monitor_L2
   - bind service s1 -monitorName monitor1
   - bind service s2 -monitorName monitor1
   - bind service s3 -monitorName monitor1
   - bind service s4 -monitorName monitor1
   - bind service s5 -monitorName monitor1

Configuration examples for MSSQL virtual server

To configure a load balancing virtual server for database specific load balancing

> **add lb vserver** DBSpecificLB1 MSSQL 192.0.2.10 1433 -dbsLb **ENABLED**

Done
> **show lb vserver** DBSpecificLB1
DBSpecificLB1 (192.0.2.10:1433) - MSSQL Type: ADDRESS
. . .
DBS_LB: ENABLED
Done
>

To configure services

add service msservice1 5.5.5.5 MSSQL 1433

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

command line

> **add lb monitor** mssql-monitor1 MSSQL-ECV -userName user1 -sqlQuery "select name from sys.databases where state=0" -evalRule "MSSQL.RES.TYPE.NE(ERROR)" -storedb EN

Done

> **show lb monitor** mssql-monitor1

1) Name.....: mssql-monitor1 Type.....: MSSQL-ECV

...

Special parameters: Database.....:""

User name.....:"user1"

Query..:select name from sys.databases where state=0 EvalRule...:MSSQL.RES.TYPE.NE(ERROR)

Version...:70 STORE_DB...:ENABLED

Done

>

Configuration examples for MySQL virtual server

To configure a load balancing virtual server for database specific load balancing

> **add lb vserver** DBSpecificLB1 MYSQL 192.0.2.10 3306 -dbsLb **ENABLED**

Done

> **show lb vserver** DBSpecificLB1

DBSpecificLB1 (192.0.2.10:3306) - MYSQL Type: ADDRESS

. . .

DBS_LB: ENABLED

Done

>

To configure services

add service msservice1 5.5.5.5 MYSQL 3306

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the command line

> **add lb monitor** mysql-monitor1 MYSQL-ECV -userName user1 -sqlQuery "show databases" -evalRule "MYSQL.RES.TYPE.NE(ERROR)" -storedb ENABLED

Done

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

> **show lb monitor** mysql-monitor1

1) Name.....: mysql-monitor1 Type.....: MYSQL-ECV State.....: ENABLED

...

Special parameters: Database.....:""

User name.....:"user1" Query...:show databases

EvalRule...:MYSQL.RES.TYPE.NE(ERROR) STORE_DB...:ENABLED

Done

>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

DataStream Reference

Jun 15, 2017

This reference describes the MySQL and TDS protocols, the database versions, the authentication methods, and the character sets supported by the DataStream feature. It also describes how NetScaler handles transaction requests and special queries that modify the state of a connection.

You can also configure the NetScaler appliance to generate audit log messages for the DataStream feature.

Supported Database Versions, Protocols, and Authentication Methods

	MySQL Database	MS SQL Database
Database Versions	MySQL database versions 4.1, 5.0, 5.1, 5.4, 5.5, and 5.6.	MS SQL database versions 2000, 2000SP1, 2005, 2008, 2008R2, and 2012.
Protocols	MySQL protocol version 10. For information about the MySQL protocol, see http://dev.mysql.com/doc/internals/en/client-server-protocol.html	Tabular Data Stream (TDS) protocol version 7.1 and higher. For information about the TDS protocol, see http://msdn.microsoft.com/en/us/library/dd304523(v=prot.13).
Authentication Methods	MySQL native authentication is supported.	SQL server authentication and Windows Authentication (Kerberos/NTLM) are supported.

Character Sets

The DataStream feature supports only the UTF-8 charset.

The character set used by the client while sending a request may be different from the character set used in the database server responses. Although the charset parameter is set during the connection establishment, it can be changed at any time by sending an SQL query. The character set is associated with a connection, and therefore, requests on connections with one character set cannot be multiplexed onto a connection with a different character set.

NetScaler parses the queries sent by the client and the responses sent by the database server.

The character set associated with a connection can be changed after the initial handshake by using the following two queries:

- SET NAMES <charset> COLLATION <collation>
- SET CHARACTER SET <charset>

Transactions

In MySQL, transactions are identified by using the connection parameter AUTOCOMMIT or the BEGIN:COMMIT queries. The AUTOCOMMIT parameter can be set during the initial handshake, or after the connection is established by using the query SET AUTOCOMMIT.

NetScaler explicitly parses each and every query to determine the beginning and end of a transaction.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

In MySQL protocol, the response contains two flags to indicate whether the connection is a transaction, the TRANSACTION and AUTOCOMMIT flags.

If the connection is a transaction, the TRANSACTION flag is set. Or, if the AutoCommit mode is OFF, the AUTOCOMMIT flag is not set. NetScaler parses the response, and if either the TRANSACTION flag is set or the AUTOCOMMIT flag is not set, it does not do connection multiplexing. When these conditions are no longer true, the NetScaler begins connection multiplexing.

> **Note**
>
> Transactions are also supported for MS SQL.

Special Queries

There are special queries, such as SET and PREPARE, that modify the state of the connection and may break request switching, and therefore, these need to be handled differently.

On receiving a request with special queries, NetScaler sends an OK response to the client and additionally, stores the request in the connection.

When a non-special query, such as INSERT and SELECT, is received along with a stored query, the NetScaler first, looks for the server-side connection on which the stored query has already been sent to the database server. If no such connections exist, NetScaler creates a new connection, and sends the stored query first, and then, sends the request with the non-special query.

In case of SET, USE db, and INIT_DB special queries, the appliance modifies a field in the server side connection corresponding to the special query. This results in better reuse of the server side connection.

Only 16 queries are stored in each connection.

The following is a list of the special queries for which NetScaler has a modified behavior.

SET query

The SET SQL queries define variables that are associated with the connection. These queries are also used to define global variables, but as of now, NetScaler is unable to differentiate between local and global variables. For this query, the NetScaler uses the 'store and forward' mechanism.

USE <db> query

Using this query, the user can change the database associated with a connection. In this case, NetScaler parses the <db> value sent and modifies a field in the server side connection to reflect the new database to be used.

INIT_DB command

Using this query, the user can change the database associated with a connection. In this case, NetScaler parses the <init_db> value sent and modifies a field in the server side connection to reflect the new database to be used.

COM_PREPARE

NetScaler stops request switching on receiving this command.

PREPARE query

This query is used to create prepared statements that are associated with a connection. For this query, the NetScaler uses the 'store and forward' mechanism.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Audit Log Message Support

You can now configure the NetScaler appliance to generate audit log messages for the DataStream feature. Audit log messages are generated when client-side and server-side connections are established, closed, or dropped. The categories of messages that you can log and view are ERROR and INFO. Error messages for client-side connections begin with "CS" and error messages for server-side connections begin with "SS." Additional information is provided where necessary. For example, log messages for closed connections (CS_CONN_CLOSED) include only the connection ID. However, log messages for established connections (CS_CONN_ESTD) include information such as the user name, database name, and the client IP address in addition to the connection ID.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Domain Name System

May 26, 2015

You can configure the Citrix NetScaler appliance to function as an authoritative domain name server (ADNS server) for a domain. You can add the DNS resource records that belong to the domain for which the appliance is authoritative and configure resource record parameters. You can also configure the NetScaler appliance as a proxy DNS server that load balances a farm of DNS name servers that are either within your network or outside your network. You can configure the appliance as an end resolver and forwarder. You can configure DNS suffixes that enable name resolution when fully qualified domain names are not configured. The appliance also supports the DNS ANY query that retrieves all the records that belong to a domain.

You can configure the NetScaler appliance to concurrently function as an authoritative DNS server for one domain and a DNS proxy server for another domain. When you configure the NetScaler as the authoritative DNS server or DNS proxy server for a zone, you can enable the appliance to use the Transmission Control Protocol (TCP) for response sizes that exceed the size limit specified for the User Datagram Protocol (UDP).

How DNS Works on the NetScaler

You can configure the NetScaler appliance to function as an ADNS server, DNS proxy server, end resolver, and forwarder. You can add DNS resource records on the NetScaler, including service (SRV) records, IPv6 (AAAA) records, address (A) records, mail exchange (MX) records, canonical name (CNAME) records, pointer (PTR) records, start of authority (SOA) records, and text (TXT) records. Also, you can configure the NetScaler to load balance external DNS name servers.

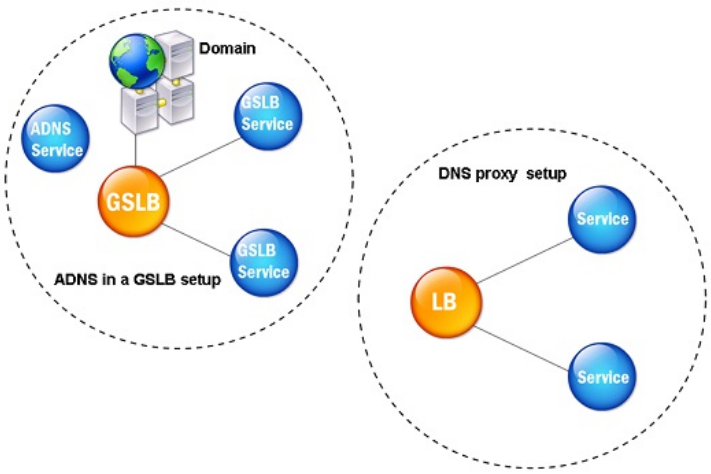
The NetScaler can be configured as the authority for a domain. To do this, you add valid SOA and NS records for the domain.

An ADNS server is a DNS server that contains complete information about a zone.

To configure the NetScaler as an ADNS server for a zone, you must add an ADNS service, and then configure the zone. To do so, you add valid SOA and NS records for the domain. When a client sends a DNS request, the NetScaler appliance searches the configured resource records for the domain name. You can configure the ADNS service to be used with the NetScaler Global Server Load Balancing (GSLB) feature.

You can delegate a subdomain, by adding NS records for the subdomain to the zone of the parent domain. You can then make the NetScaler authoritative for the subdomain, by adding a "glue record" for each of the subdomain name servers. If GSLB is configured, the NetScaler makes a GSLB load balancing decision based on its configuration and replies with the IP address of the selected virtual server. The following figure shows the entities in an ADNS GSLB setup and a DNS proxy setup.

Figure 1. DNS Proxy Entity Model



The NetScaler appliance can function as a DNS proxy. Caching of DNS records, which is an important function of a DNS proxy, is enabled by

© 1999-2017 Citrix Systems, Inc. All rights reserved.

default on the NetScaler appliance. This enables the NetScaler to provide quick responses for repeated translations. You must also create a load balancing DNS virtual server, and DNS services, and then bind these services to the virtual server.

The NetScaler provides two options, minimum time to live (TTL) and maximum TTL for configuring the lifetime of the cached data. The cached data times out as specified by your settings for these two options. The NetScaler checks the TTL of the DNS record coming from the server. If the TTL is less than the configured minimum TTL, it is replaced with the configured minimum TTL. If the TTL is greater than the configured maximum TTL, it is replaced with the configured maximum TTL.

The NetScaler also allows caching of negative responses for a domain. A negative response indicates that information about a requested domain does not exist, or that the server cannot provide an answer for the query. The storage of this information is called *negative caching*. Negative caching helps speed up responses to queries on a domain, and can optionally provide the record type.

A negative response can be one of the following:

- NXDOMAIN error message - If a negative response is present in the local cache, the NetScaler returns an error message (NXDOMAIN). If the response is not in the local cache, the query is forwarded to the server, and the server returns an NXDOMAIN error to the NetScaler. The NetScaler caches the response locally, then returns the error message to the client.
- NODATA error message - The NetScaler sends a NODATA error message, if the domain name in query is valid but records of the given type are not available.

The NetScaler supports recursive resolution of DNS requests. In recursive resolution, the resolver (DNS client) sends a recursive query to a name server for a domain name. If the queried name server is authoritative for the domain, it responds with the requested domain name. Otherwise, the NetScaler queries the name servers recursively until the requested domain name is found.

Before you can apply the recursive query option, you must first enable it. You can also set the number of times the DNS resolver must send a resolution request (DNS retries) if a DNS lookup fails.

You can configure the NetScaler as a DNS forwarder. A forwarder passes DNS requests to external name servers. The NetScaler allows you to add external name servers and provides name resolution for domains outside the network. The NetScaler also allows you to set the name lookup priority to DNS or Windows Internet Name Service (WINS).

Round Robin DNS

When a client sends a DNS request to find the DNS resource record, it receives a list of IP addresses resolving to the name in the DNS request. The client then uses one of the IP addresses in the list, generally, the first record or IP address. Hence, a single server is used for the total TTL of the cache and is overloaded when a large number of requests arrive.

When the NetScaler receives a DNS request, it responds by changing the order of the list of DNS resource records in a round robin method. This feature is called *round robin DNS*. Round robin distributes the traffic equally between data centers. The NetScaler performs this function automatically. You do not have to configure this behavior.

Functional Overview

If the NetScaler is configured as an ADNS server, it returns the DNS records in the order in which the records are configured. If the NetScaler is configured as a DNS proxy, it returns the DNS records in the order in which it receives the records from the server. The order of the records present in the cache matches the order in which records are received from the server.

The NetScaler then changes the order in which records are sent in the DNS response in a round robin method. The first response contains the first record in sequence, the second response contains the second record in sequence, the third response contains the third record in sequence, and the order continues in the same sequence. Thus, clients requesting the same name can connect to different IP addresses.

Round Robin DNS Example

As an example of round robin DNS, consider DNS records that have been added as follows:

```
add dns addRec ns1 1.1.1.1  add dns addRec ns1 1.1.1.2  add dns addRec ns1 1.1.1.3  add dns addRec ns1 1.1.1.4
```

The domain, abc.com is linked to an NS record as follows:

```
add dns nsrec abc.com. ns1
```

When the NetScaler receives a query for the A record of ns1, the Address records are served in a round robin method as follows. In the first DNS response, 1.1.1.1 is served as the first record:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
ns1.        1H IN A    1.1.1.1  ns1.        1H IN A    1.1.1.2  ns1.        1H IN A    1.1.1.3  ns1.        1H IN A    1.1.1.4
```

In the second DNS response, the second IP address, 1.1.1.2 is served as the first record:

```
ns1.        1H IN A    1.1.1.2  ns1.        1H IN A    1.1.1.3  ns1.        1H IN A    1.1.1.4  ns1.        1H IN A    1.1.1.1
```

In the third DNS response, the third IP address, 1.1.1.2 is served as the first record:

```
ns1.        1H IN A    1.1.1.3  ns1.        1H IN A    1.1.1.4  ns1.        1H IN A    1.1.1.1  ns1.        1H IN A    1.1.1.2
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DNS Resource Records

Oct 04, 2016

You configure resource records on the Citrix® NetScaler® appliance when you configure the appliance as an ADNS server for a zone. You can also configure resource records on the appliance if the resource records belong to a zone for which the appliance is a DNS proxy server. On the appliance, you can configure the following record types:

- Service records
- AAAA records
- Address records
- Mail Exchange records
- Name Server records
- Canonical records
- Pointer records
- NAPTR records
- Start of Authority records
- Text records

The following table lists the record types and the number of records (per record type) that you can configure for a domain name record on the NetScaler. For example, you can configure a maximum of 25 IP addresses for one record.

Table 1. Record Type and Number Configurable

Record Type	Number of Records
Address (A)	25
IPv6 (AAAA)	5
Mail exchange (MX)	12
Name server (NS)	16
Service (SRV)	8
Pointer (PTR)	20
Canonical name (CNAME)	1
Start of Authority (SOA)	1
Text (TXT)	20

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Record Type	Number of Records
Naming Authority Pointer (NAPTR)	20

Note

The maximum number of IP addresses for a specific hostname is 25. However, the number of different address records can be more than 25.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating SRV Records for a Service

Nov 21, 2014

The SRV record provides information about the services available on the NetScaler appliance. An SRV record contains the following information: name of the service and the protocol, domain name, TTL, DNS class, priority of the target, weight of records with the same priority, port of the service, and host name of the service. The NetScaler chooses the SRV record that has the lowest priority setting first. If a service has multiple SRV records with the same priority, clients use the weight field to determine which host to use.

To add an SRV record by using the command line interface

At the command prompt, type the following commands to add an SRV record and verify the configuration:

- add dns srvRec <domain> <target> -priority <positive_integer> -weight <positive_integer> -port <positive_integer> [-TTL <secs>]
- sh dns srvRec <domain>

Example

```
> add dns srvRec _http._tcp.example.com nameserver1.com -priority 1 -weight 1 -port 80
 Done
> show dns srvRec _http._tcp.example.com
1)     Domain Name : _http._tcp.example.com
       Target Host : nameserver1.com
       Priority : 1    Weight : 1
       Port : 80       TTL : 3600 secs
 Done
>
```

To modify or remove an SRV record by using the command line interface

- To modify an SRV record, type the set dns srvRec command, the name of the domain for which the SRV record is configured, the name of the target host that hosts the associated service, and the parameters to be changed, with their new values.
- To remove an SRV record, type the rm dns srvRec command, the name of the domain for which the SRV record is configured, and the name of the target host that hosts the associated service.

To configure an SRV record by using the configuration utility

Navigate to Traffic Management > DNS > Records > SRV Records and create an SRV record.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating AAAA Records for a Domain Name

Nov 21, 2014

An AAAA resource record stores a single IPv6 address.

To add an AAAA record by using the command line interface

At the command prompt, type the following commands to add an AAAA record and verify the configuration:

- add dns aaaaRec <hostName> <IPv6Address> ... [-TTL <secs>]
- show dns aaaaRec <hostName>

Example

```
> add dns aaaaRec www.example.com 2001:0db8:0000:0000:0000:0000:1428:57ab
 Done
> show dns aaaaRec www.example.com
1)     Host Name : www.example.com
       Record Type : ADNS            TTL : 5 secs
       IPV6 Address : 2001:db8::1428:57ab
 Done
>
```

To remove an AAAA record and all of the IPv6 addresses associated with the domain name, type the rm dns aaaaRec command and the domain name for which the AAAA record is configured. To remove only a subset of the IPv6 addresses associated with the domain name in an AAAA record, type the rm dns aaaaRec command, the domain name for which the AAAA record is configured, and the IPv6 addresses that you want to remove.

To add an AAAA record by using the configuration utility

Navigate to Traffic Management > DNS > Records > AAAA Records and create an AAAA record.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating Address Records for a Domain Name

Nov 21, 2014

Address (A) records are DNS records that map a domain name to an IPv4 address.

You cannot delete Address records for a host participating in global server load balancing (GSLB). However, the NetScaler deletes Address records added for GSLB domains when you unbind the domain from a GSLB virtual server. Only user-configured records can be deleted manually. You cannot delete a record for a host referenced by records such as NS, MX, or CNAME.

To add an Address record by using the command line interface

At the command prompt, type the following commands to add an Address record and verify the configuration:

- add dns addRec <hostName> <IPAddress> [-TTL <secs>]
- show dns addRec <hostName>

Example

```
> add dns addRec ns.example.com 192.0.2.0
 Done
> show dns addRec ns.example.com
1)     Host Name : ns.example.com
       Record Type : ADNS                    TTL : 5 secs
       IP Address : 192.0.2.0
 Done
>
```

To remove an Address record and all of the IP addresses associated with the domain name, type the rm dns addRec command and the domain name for which the Address record is configured. To remove only a subset of the IP addresses associated with the domain name in an Address record, type the rm dns addRec command, the domain name for which the Address record is configured, and the IP addresses that you want to remove.

To add an Address record by using the configuration utility

Navigate to Traffic Management > DNS > Records > Address Records and create an Address record.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating MX Records for a Mail Exchange Server

Nov 21, 2014

Mail Exchange (MX) records are used to direct email messages across the Internet. An MX record contains an MX preference that specifies the MX server to be used. The MX preference values range from 0 through 65536. An MX record contains a unique MX preference number. You can set the MX preference and the TTL values for an MX record.

When an email message is sent through the Internet, a mail transfer agent sends a DNS query requesting the MX record for the domain name. This query returns a list of host names of mail exchange servers for the domain, along with a preference number. If there are no MX records, the request is made for the Address record of that domain. A single domain can have multiple mail exchange servers.

To add an MX record by using the command line interface

At the command prompt, type the following commands to add an MX record and verify the configuration:

- add dns mxRec <domain> -mx <string> -pref <positive_integer> [-TTL <secs>]
- show dns mxRec <domain>

Example

```
> add dns mxRec example.com -mx mail.example.com -pref 1
 Done
> show dns mxRec example.com
1)     Domain : example.com    MX Name : mail.example.com
       Preference :   1        TTL : 5 secs
 Done
>
```

To modify or remove an MX record by using the command line interface

- To modify an MX record, type the set dns mxRec command, the name of the domain for which the MX record is configured, the name of the MX record, and the parameters to be changed, with their new values.
- To set the TTL parameter to its default value, type the unset dns mxRec command, the name of the domain for which the MX record is configured, the name of the MX record, and -TTL without any TTL value. You can use the unset dns mxRec command to unset only the TTL parameter.
- To remove an MX record, type the rm dns mxRec command, the name of the domain for which the MX record is configured, and the name of the MX record.

To add an MX record by using the configuration utility

Navigate to Traffic Management > DNS > Records > Mail Exchange Records and create an MX record.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating NS Records for an Authoritative Server

Nov 21, 2014

Name Server (NS) records specify the authoritative server for a domain. You can configure a maximum of 16 NS records. You can use an NS record to delegate the control of a subdomain to a DNS server.

To create an NS record by using the command line interface

At the command prompt, type the following commands to create an NS record and verify the configuration:

- add dns nsRec <domain> <nameServer> [-TTL <secs>]
- show dns nsRec <domain>

Example

```
> add dns nsRec example.com nameserver1.example.com
 Done
> show dns nsRec example.com
1)     Domain : example.com    NameServer : nameserver1.example.com
       TTL : 5 sec
 Done
>
```

To remove an NS record, type the rm dns nsRec command, the name of the domain to which the NS record belongs, and the name of the name server.

To create an NS record by using the configuration utility

Navigate to Traffic Management > DNS > Records > Name Server Records and create an NS record.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating CNAME Records for a Subdomain

May 26, 2015

A canonical name record (CNAME record) is an alias for a DNS name. These records are useful when multiple services query the DNS server. The host that has an address (A) record cannot have a CNAME record.

In some cases, a NetScaler appliance in proxy mode requests an address record from the cache instead of the server.

To add a CNAME record by using the command line interface

At the command prompt, type the following commands to create a CNAME record and verify the configuration:

- add dns cnameRec <aliasName> <canonicalName> [-TTL <secs>]
- show dns cnameRec <aliasName>

Example

```
> add dns cnameRec www.example.com www.examplenw.com
 Done
> show dns cnameRec www.example.com
      Alias Name     Canonical Name  TTL
1)    www.example.com       www.examplenw.com      5 secs
 Done
>
```

To remove a CNAME record for a given domain, type the rm dns cnameRec command and the alias of the domain name.

To add a CNAME record by using the configuration utility

Navigate to Traffic Management > DNS > Records > Canonical Records and create a CNAME record.

Caching of CNAME Records

Updated: 2015-05-26

NetScaler ADC when deployed in a proxy mode does not always send the query for an address record to the back-end server. This happens when for a answer to a query for an address record, a partial CNAME chain is present in the cache. There are few conditions in which the ADC caches the partial CNAME record and serves the query from the cache. Following are the conditions:

- NetScaler should be deployed in a proxy mode
- The response from the back-end server should have a CNAME chain, for which the record type of last entry in the answer section must be a CNAME and the question type not a CNAME
- The response from the back-end server cannot be a No-data or NX-Domain
- The response from the back-end server has to be a authoritative response

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating NAPTR Records for Telecommunications Domain

Oct 29, 2014

NAPTR (Naming Address Pointer) is one of the most commonly used DNS record in telecommunications domain. NAPTR records map the Internet telephony address space to the Internet address space. They therefore enable a mobile device to send a request to the correct server. The combination of NAPTR records with Service Records (SRV) allows the chaining of multiple records to form complex rewrite rules that produce new domain labels or uniform resource identifiers (URIs). The DNS code for NAPTR is 35.

NetScaler ADCs support NAPTR in two modes: ADNS mode and proxy mode. In proxy mode, the ADC caches the response from the servers and uses the cached records to server future queries. A maximum of 20 NAPTR records can be added for a particular domain in NetScaler. NetScaler caches the reply to a DNS NAPTR record query. Any subsequent requests for the NAPTR record is served from the cache.

To create a NAPTR record by using command line interface

At the command prompt, type the following commands to add a NAPTR record and verify the configuration:

add dns naptrRec <order> <preference>[flags<string>][services<string>](regexp<expressions>|-replacement<string>) [-TTL<secs>]

To remove a NAPTR record by using command line interface

rm dns naptrRec<domain> (<order> <preference> [-flags <string>] [-services <string>] (-regexp <expression> | -replacement <string>)) | -recordId <positive_integer>@)

To configure a NAPTR record using configuration utility

Navigate to Traffic Management > DNS > Records > NAPTR Records and create an NAPTR record.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating PTR Records for IPv4 and IPv6 Addresses

May 26, 2015

A pointer (PTR) record translates an IP address to its domain name. IPv4 PTR records are represented by the octets of an IP address in reverse order with the string "in-addr.arpa." appended at the end. For example, the PTR record for the IP address 1.2.3.4 is 4.3.2.1.in-addr.arpa.

IPv6 addresses are reverse mapped under the domain IP6.ARPA. IPv6 reverse-maps use a sequence of nibbles separated by dots with the suffix ".IP6.ARPA" as defined in RFC 3596. For example, the reverse lookup domain name corresponding to the address, 4321:0:1:2:3:4:567:89ab would be b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.ARPA.

To add a PTR record by using the command line interface

At the command prompt, type the following commands to add a PTR record and verify the configuration:

- add dns ptrRec <reverseDomain> <domain> [-TTL <secs>]
- show dns ptrRec <reverseDomain>

Example

```
> add dns ptrRec 0.2.0.192.in-addr.arpa example.com
 Done
> show dns ptrRec 0.2.0.192.in-addr.arpa
1)     Reverse Domain Name : 0.2.0.192.in-addr.arpa
       Domain Name : example.com            TTL : 3600 secs
 Done
>
```

To remove a PTR record, type the rm dns ptrRec command and the reverse domain name associated with the PTR record

To add a PTR record by using the configuration utility

Navigate to Traffic Management > DNS > Records > PTR Records and create a PTR record.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating SOA Records for Authoritative Information

Nov 21, 2014

A Start of Authority (SOA) record is created only at the zone apex and contains information about the zone. The record includes, among other parameters, the primary name server, contact information (e-mail), and default (minimum) time-to-live (TTL) values for records.

To create an SOA record by using the command line interface

At the command prompt, type the following commands to add an SOA record and verify the configuration:

- add dns soaRec <domain> -originServer <originServerName> -contact <contactName>
- sh dns soaRec <do main>

Example

```
> add dns soaRec example.com -originServer nameserver1.example.com -contact admin.example.com
 Done
> show dns soaRec example.com
1)     Domain Name : example.com
       Origin Server : nameserver1.example.com
       Contact : admin.example.com
       Serial No. : 100       Refresh : 3600 secs     Retry : 3 secs
       Expire : 3600 secs     Minimum : 5 secs        TTL : 3600 secs
 Done
>
```

To modify or remove an SOA record by using the command line interface

- To modify an SOA record, type the set dns soaRec command, the name of the domain for which the record is configured, and the parameters to be changed, with their new values.
- To remove an SOA record, type the rm dns soaRec command and the name of the domain for which the record is configured.

To configure an SOA record by using the configuration utility

Navigate to Traffic Management > DNS > Records > SOA Records and create an SOA record.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating TXT Records for Holding Descriptive Text

Nov 21, 2014

Domain hosts store TXT records for informative purposes. A TXT record's RDATA component, which consists of one or more character strings of variable length, can store practically any information that a recipient might need to know about the domain, including information about the service provider, contact person, email addresses, and associated details. SPF (Sender Policy Framework) protection has been the most prominent use case for the TXT record.

All configuration types (authoritative DNS, DNS proxy, end resolver, and forwarder configurations) on the NetScaler appliance support TXT records. You can add a maximum of 20 TXT resource records to a domain. Each resource record is stored with a unique, internally generated record ID. You can view the ID of a record and use it to delete the record. However, you cannot modify a TXT resource record.

To create a TXT resource record by using the command line interface

At the command prompt, type the following commands to create a TXT resource record and verify the configuration:

- add dns txtRec <domain> <string> ... [-TTL <secs>]
- show dns txtRec [<domain> | -type <type>]

Example

```
> add dns txtRec www.example.com "Contact: Mark" "Email: mark@example.com" -TTL 36000
 Done
> show dns txtRec www.example.com
1) Domain : www.example.com Record id: 13783    TTL : 36000 secs Record Type : ADNS
  "Contact: Mark"
  "Email: mark@example.com"
 Done
```

To remove a TXT resource record by using the command line interface

At the command prompt, type the following commands to remove a TXT resource record and verify the configuration:

- rm dns txtRec <domain> (<string> ... | -recordId <positive_integer>)
- show dns txtRec [<domain> | -type <type>]

Example

You can use the show dns txtRec command first to view the record ID of the TXT resource record that you want to remove, as shown:

```
> show dns txtRec www.example.com
1) Domain : www.example.com Record id: 36865    TTL : 36000 secs Record Type : ADNS
  "Contact: Evan"
  "Email: evan@example.com"
2) Domain : www.example.com Record id: 14373    TTL : 36000 secs Record Type : ADNS
  "Contact: Mark"
  "Email: mark1@example.com"
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

The simpler method of deleting a TXT record is to use the record ID. If you want to provide the strings, enter them in the order in which they are stored in the record. In the following example, the TXT record is deleted by using its record ID.

```
>rm dns txtRec www.example.com -recordID 36865
 Done
> show dns txtRec www.example.com
1) Domain : www.example.com Record id: 14373    TTL : 36000 secs Record Type : ADNS
  "Contact: Mark"
  "Email: mark1@example.com"
 Done
```

To configure a TXT record by using the configuration utility

Navigate to Traffic Management > DNS > Records > TXT Records and create a TXT record.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing DNS Statistics

Aug 27, 2013

You can view the DNS statistics generated by the Citrix® NetScaler® appliance. The DNS statistics include runtime, configuration, and error statistics.

To view DNS records statistics by using the command line interface

At the command prompt, type:

stat dns

Example

```
> stat dns
DNS Statistics

Runtime Statistics
Dns queries                    21
NS queries                     8
SOA queries                    18
.
.
.
Configuration Statistics
AAAA records                   17
A records                      36
MX records                     9
.
.
.
Error Statistics
Nonexistent domain             17
No AAAA records                0
No A records                   13
.
.
.
Done
>
```

To view DNS records statistics by using the configuration utility

1. Navigate to Traffic Management > DNS.
2. In the details pane, click Statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a DNS Zone

Oct 29, 2014

A DNS zone entity on the Citrix® NetScaler® appliance facilitates the ownership of a domain on the appliance. A zone on the appliance also enables you to implement DNS Security Extensions (DNSSEC) for the zone, or to offload the zone's DNSSEC operations from the DNS servers to the appliance. DNSSEC sign operations are performed on all the resource records in a DNS zone. Therefore, if you want to sign a zone, or if you want to offload DNSSEC operations for a zone, you must first create the zone on the NetScaler appliance.

You must create a DNS zone on the appliance in the following scenarios:

- The NetScaler appliance owns all the records in a zone, that is, the appliance is operating as the authoritative DNS server for the zone. The zone must be created with the proxyMode parameter set to NO.
- The NetScaler appliance owns only a subset of the records in a zone, and all the other resource records in the zone are hosted on a set of back-end name servers for which the appliance is configured as a DNS proxy server. A typical configuration where the NetScaler appliance owns only a subset of the resource records in the zone is a global server load balancing (GSLB) configuration. Only the GSLB domain names are owned by the NetScaler appliance, while all the other records are owned by the back-end name servers. The zone must be created with the proxyMode parameter set to YES.
- You want to offload DNSSEC operations for a zone from your authoritative DNS servers to the appliance. The zone must be created with the proxyMode parameter set to YES. You might need to configure additional settings for the zone.

The current topic describes how to create a zone for the first two scenarios. For more information about how to configure a zone for offloading DNSSEC operations to the appliance, see Offloading DNSSEC Operations to the NetScaler Appliance.

Note: If the NetScaler is operating as the authoritative DNS server for a zone, you must create Start of Authority (SOA) and name server (NS) records for the zone before you create the zone. If the NetScaler is operating as the DNS proxy server for a zone, SOA and NS records must not be created on the NetScaler appliance. For more information about creating SOA and NS records, see Configuring DNS Resource Records.

When you create a zone, all existing domain names and resource records that end with the name of the zone are automatically treated as a part of the zone. Additionally, any new resource records created with a suffix that matches the name of the zone are implicitly included in the zone.

To create a DNS zone on the NetScaler appliance by using the command line interface

At the command prompt, type the following command to add a DNS zone to the NetScaler appliance and verify the configuration:

- add dns zone <zoneName> -proxyMode (YES | NO)
- show dns zone [<zoneName> | -type <type>]

Example

```
> add dns zone example.com -proxyMode Yes
 Done
> show dns zone example.com
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Zone Name : example.com
    Proxy Mode : YES
Done
>
```

To modify or remove a DNS zone by using the command line interface

- To modify a DNS zone, type the set dns zone command, the name of the DNS zone, and the parameters to be changed, with their new values.
- To remove a DNS zone, type the rm dns zone command and the name of the dns zone.

To configure a DNS zone by using the configuration utility

Navigate to Traffic Management > DNS > Zones and create a DNS zone.

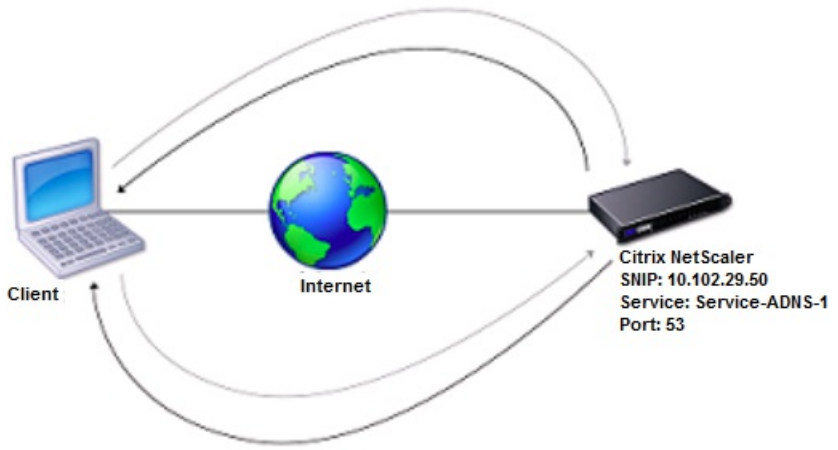
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NetScaler as an ADNS Server

Aug 29, 2017

You can configure the Citrix® NetScaler® appliance to function as an authoritative domain name server (ADNS) for a domain. As an ADNS server for a domain, the NetScaler resolves DNS requests for all types of DNS records that belong to the domain. To configure the NetScaler to function as an ADNS server for a domain, you must create an ADNS service and configure NS and Address records for the domain on the NetScaler. The ADNS service can be configured using the subnet IP address (SNIP) or a separate IP address. The following topology diagram shows a sample configuration and the flow of requests and responses.

Figure 1. NetScaler as an ADNS



The following table shows the parameters that are configured for the ADNS service illustrated in the preceding topology diagram.

Table 1. Example of ADNS Service Configuration

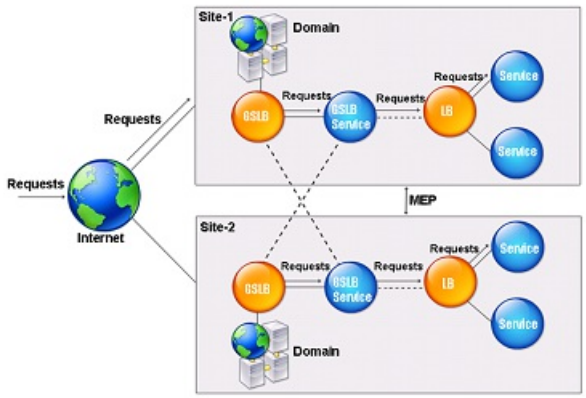
Entity type	Name	IP address	Type	Port
ADNS Service	Service-ADNS-1	10.102.29.51	ADNS	53

To configure an ADNS setup, you must configure the ADNS service. For instructions on configuring the ADNS service, see "Load Balancing".

During DNS resolution, the ADNS server directs the DNS proxy or local DNS server to query the NetScaler for the IP address of the domain. Because the NetScaler is authoritative for the domain, it sends the IP address to the DNS proxy or local DNS server. The following diagram describes the placement and role of the ADNS server in a GSLB configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Figure 2. GSLB Entity Model



Note: In ADNS mode, if you remove SOA and ADNS records, the following do not function for the domain hosted by the NetScaler: ANY query (for more information about the ANY query, see DNS ANY Query), and negative responses, such as NODATA and NXDOMAIN.
This document includes the following information:

- Creating an ADNS Service
- Configuring the ADNS Setup to Use TCP
- Adding DNS Resource Records
- Removing ADNS Services
- Configuring Domain Delegation

Creating an ADNS Service

Updated: 2014-11-14

An ADNS service is used for global service load balancing. For more information about creating a GSLB setup, see "Global Server Load Balancing". You can add, modify, enable, disable, and remove an ADNS service. For instructions on creating an ADNS service, see Configuring Services.

Note: You can configure the ADNS service to use SNIP or any new IP address.
When you create an ADNS service, the NetScaler responds to DNS queries on the configured ADNS service IP and port.

You can verify the configuration by viewing the properties of the ADNS service You can view properties such as name, state, IP address, port, protocol, and maximum client connections.

Configuring the ADNS Setup to Use TCP

Updated: 2013-08-26

By default, some clients use the User Datagram Protocol (UDP) for DNS, which specifies a limit of 512 bytes for the payload length of UDP packets. To handle payloads that exceed 512 bytes in size, the client must use the Transmission Control Protocol (TCP). To enable DNS communications over TCP, you must configure the NetScaler appliance to use the TCP protocol for DNS. The NetScaler then sets the truncation bit in the DNS response packets. The truncation bit specifies

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

that the response is too large for UDP and that the client must send the request over a TCP connection. The client then uses the TCP protocol on port 53 and opens a new connection to the NetScaler. The NetScaler listens on port 53 with the IP address of the ADNS service to accept the new TCP connections from the client.

To configure the NetScaler to use the TCP protocol, you must configure an ADNS_TCP service. For instructions on creating an ADNS_TCP service, see "Load Balancing".
Important: To configure the NetScaler to use UDP for DNS and use TCP only when the payload length of UDP exceeds 512 bytes, you need to configure the ADNS and ADNS_TCP services. The IP address of the ADNS_TCP service must be same as the IP address of the ADNS service.

Adding DNS Resource Records

Updated: 2013-08-26

After you create an ADNS service, you can add DNS records. For instructions on adding DNS records, see Configuring DNS Resource Records.

Removing ADNS Services

Updated: 2013-08-27

For instructions on removing services, see Load Balancing.

Configuring Domain Delegation

Domain delegation is the process of assigning responsibility for a part of the domain space to another name server. Therefore, during domain delegation, the responsibility for responding to the query is delegated to another DNS server. Delegation uses NS records.

In the following example, sub1.abc.com is the subdomain for abc.com. The procedure describes the steps to delegate the subdomain to the name server ns2.sub1.abc.com and add an Address record for ns2.sub1.abc.com.

To configure domain delegation, you need to perform the following tasks, which are described in the sections that follow:

1. Create an SOA record for a domain.
2. Create an NS record to add a name server for the domain.
3. Create an Address record for the name server.
4. Create an NS record to delegate the subdomain.
5. Create a glue record for the name server.

Creating an SOA Record

For instructions on configuring SOA records, see Creating SOA Records for Authoritative Information.

Creating an NS Record for a Name Server

For instructions on configuring an NS record, see Creating NS Records for an Authoritative Server. In the Name Server drop-down list, select the primary authoritative name server, for example, ns1.abc.com.

Creating an Address Record

For instructions on configuring Address records, see Creating Address Records for a Domain Name. In the Host Name and IP address text boxes, type the domain name for the DNS Address record and the IP address, for example, ns1.abc.com and

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

10.102.11.135, respectively.

Creating an NS Record for Domain Delegation

For instructions on configuring NS records, see Creating NS Records for an Authoritative Server. In the Name Server drop-down list, select the primary authoritative name server, for example, ns2.sub1.abc.com.

Creating a Glue Record

NS records are usually defined immediately after the SOA record (but this is not a restriction.) A domain must have at least two NS records. If an NS record is defined within a domain, it must have a matching Address record. This Address record is referred to as a glue record. Glue records speed up DNS queries.

For instructions on adding glue records for a subdomain, see the procedure for adding an Address (A) record, Configuring DNS Resource Records.

For instructions on configuring Address records, seeCreating Address Records for a Domain Name. In Host Name and IP address text boxes, type the domain name for the DNS Address record and the IP address, for example, ns2.sub1.abc.com and 10.102.12.135, respectively.

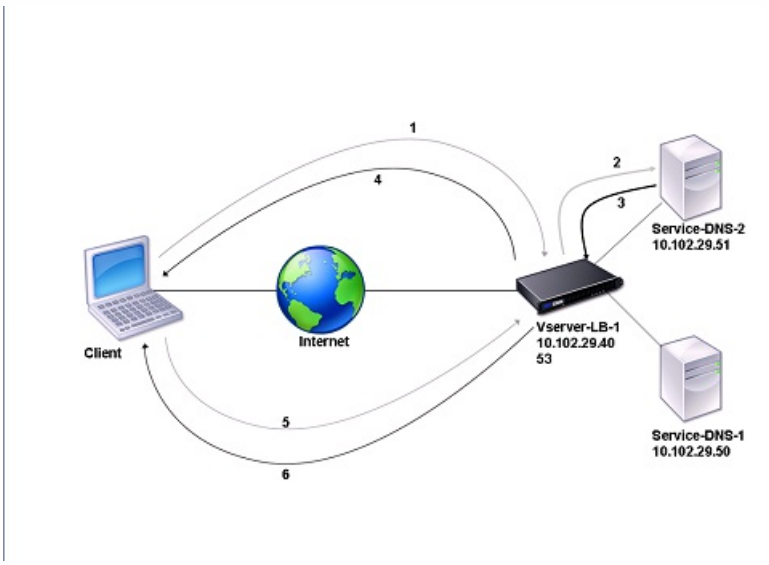
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NetScaler as a DNS Proxy Server

Jul 16, 2017

As a DNS proxy server, the Citrix® NetScaler® appliance can function as a proxy for either a single DNS server or a group of DNS servers. The flow of requests and responses is illustrated in the following sample topology diagram.

Figure 1. NetScaler as DNS proxy



By default, the NetScaler appliance caches responses from DNS name servers. When the appliance receives a DNS query, it checks for the queried domain in its cache. If the address for the queried domain is present in its cache, the NetScaler returns the corresponding address to the client. Otherwise, it forwards the query to a DNS name server that checks for the availability of the address and returns it to the NetScaler. The NetScaler then returns the address to the client.

For requests for a domain that has been cached earlier, the NetScaler serves the Address record of the domain from the cache without querying the configured DNS server.

The NetScaler discards a record stored in its cache when the time-to-live (TTL) value of the record reaches the configured value. A client that requests an expired record has to wait until the NetScaler retrieves the record from the server and updates its cache. To avoid this delay, the NetScaler proactively updates the cache by retrieving the record from the server before the record expires.

The following table lists sample names and the values of the entities that need to be configured on the NetScaler.

Table 1. Example of DNS Proxy Entity Configuration

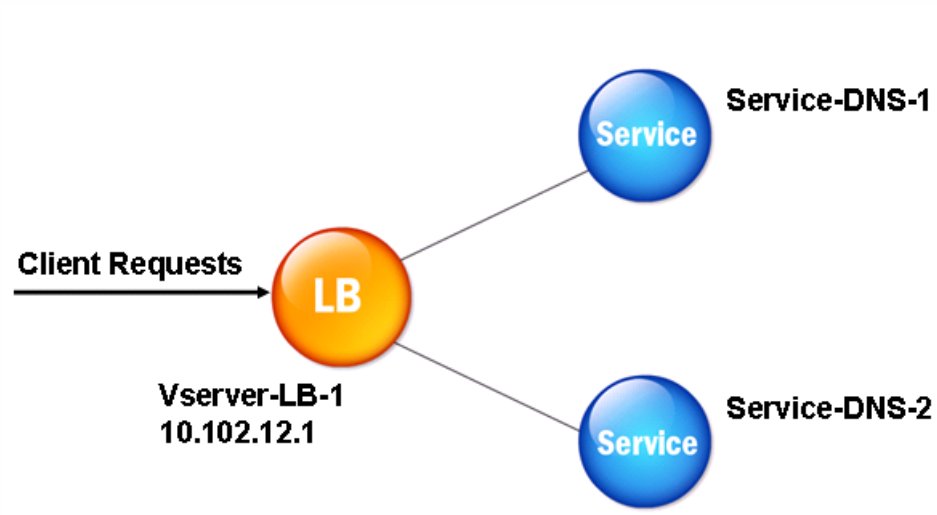
Entity type	Name	IP address	Type	Port
LB virtual server	Vserver-DNS-1	10.102.29.40	DNS	53
Services	Service-DNS-1	10.102.29.50	DNS	53
	Service-DNS-2	10.102.29.51	DNS	53

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity type	Name	IP address	Type	Port

The following diagram shows the entities of a DNS Proxy and the values of the parameters to be configured on the NetScaler.

Figure 2. DNS Proxy Entity Model



Note: To configure DNS proxy, you need to know how to configure load balancing services and virtual servers. For information about configuring load balancing services and virtual servers, see "Load Balancing", and then configure DNS proxy setup.

This document includes the following information:

- Creating a Load Balancing Virtual Server
- Creating DNS Services
- Binding a Load Balancing Virtual Server to DNS Services
- Configuring the DNS Proxy Setup to Use TCP
- Adding DNS Resource Records
- Removing a Load Balancing DNS Virtual Server
- Limiting the Number of Concurrent DNS Requests on a Client Connection

Creating a Load Balancing Virtual Server

Updated: 2014-12-29

To configure a DNS Proxy on the NetScaler ADC, configure a load balancing virtual server of type DNS. To configure a DNS virtual server to load balance a set of DNS servers that support recursive queries, you must set the Recursion Available option. With this option, the RA bit is set to ON in the DNS replies from the DNS virtual server.

For instructions on creating a load balancing virtual server, see "Load Balancing".

Creating DNS Services

Updated: 2013-08-26

© 1999-2017 Citrix Systems, Inc. All rights reserved.

After creating a load balancing virtual server of type DNS, you must create DNS services. You can add, modify, enable, disable, and remove a DNS service. For instructions on creating a DNS service, see "Load Balancing".

Binding a Load Balancing Virtual Server to DNS Services

Updated: 2013-09-13

To complete the DNS Proxy configuration, you must bind the DNS services to the load balancing virtual server. For instructions on binding a service to a load balancing virtual server, see "Load Balancing".

Configuring the DNS Proxy Setup to Use TCP

Updated: 2013-08-26

Some clients use the User Datagram Protocol (UDP) for DNS communications. However, UDP specifies a maximum packet size of 512 bytes. When payload lengths exceed 512 bytes, the client must use the Transmission Control Protocol (TCP). When a client sends the Citrix® NetScaler® appliance a DNS query, the appliance forwards the query to one of the name servers. If the response is too large for a UDP packet, the name server sets the truncation bit in its response to the NetScaler. The truncation bit indicates that the response is too large for UDP and that the client must send the query over a TCP connection. The NetScaler relays the response to the client with the truncation bit intact and waits for the client to initiate a TCP connection with the IP address of the DNS load balancing virtual server, on port 53. The client sends the request over a TCP connection. The NetScaler appliance then forwards the request to the name server and relays the response to the client.

To configure the NetScaler to use the TCP protocol for DNS, you must configure a load balancing virtual server and services, both of type DNS_TCP. You can configure monitors of type DNS_TCP to check the state of the services. For instructions on creating DNS_TCP virtual servers, services, and monitors, see "Load Balancing."

For updating the records proactively, the NetScaler uses a TCP connection to the server to retrieve the records.

Important: To configure the NetScaler to use UDP for DNS and use TCP only when the payload length of UDP exceeds 512 bytes, you need to configure DNS and DNS_TCP services. The IP address of the DNS_TCP service must be same as that of the DNS service.

Configuring Time-to-Live Values for DNS Entries

The TTL is the same for all DNS records with the same domain name and record type. If the TTL value is changed for one of the records, the new value is reflected in all records of the same domain name and type. The default TTL value is 3600 seconds. The minimum is 0, and the maximum is 604800. If a DNS entry has a TTL value less than the minimum or greater than the maximum, it is saved as the minimum or maximum TTL value, respectively.

To specify the minimum and/or maximum TTL by using the command line interface

At the NetScaler command prompt, type the following commands to specify the minimum and maximum TTL and verify the configuration:

- set dns parameter [-minTTL <secs>] [-maxTTL <secs>]
- show dns parameter

Example

```
> set dns parameter -minTTL 1200 -maxTTL 1800
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 Done
> show dns parameter
      DNS parameters:
      DNS retries: 5
      Minimum TTL: 1200            Maximum TTL: 1800
  .
  .
  .
 Done
>
```

To specify the minimum and/or maximum TTL by using the configuration utility

1. Navigate to Traffic Management > DNS.
2. In the details pane, under Settings, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, in TTL, in the Minimum and Maximum text boxes, type the minimum and maximum time to live (in seconds), respectively, and then click OK.

Note: When the TTL expires, the record is deleted from the cache. The NetScaler proactively contacts the servers and obtains the DNS record just before the DNS record expires.

Flushing DNS Records

You can delete all DNS records present in the cache. For example, you might want to flush DNS records when a server is restarted after modifications are made.

To delete all proxy records by using the command line interface

At the NetScaler command prompt, type:

```
flush dns proxyRecords
```

To delete all proxy records by using the configuration utility

1. Navigate to Traffic Management > DNS > Records.
2. In the details pane, click Flush Proxy Records.

Adding DNS Resource Records

Updated: 2013-08-26

You can add DNS records to a domain for which the Citrix® NetScaler® appliance is configured as a DNS proxy server. For information about adding DNS records, see Configuring DNS Resource Records.

Removing a Load Balancing DNS Virtual Server

Updated: 2013-08-27

For information about removing a load balancing virtual server, see Load Balancing.

Limiting the Number of Concurrent DNS Requests on a Client Connection

Updated: 2013-09-10

You can limit the number of concurrent DNS requests on a single client connection, which is identified by the <clientip:port>-<vserver ip:port> tuple. Concurrent DNS requests are those requests that the NetScaler appliance has forwarded to the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

name servers and for which the appliance is awaiting responses. Limiting the number of concurrent requests on a client connection enables you to protect the name servers when a hostile client attempts a Distributed Denial of Service (DDoS) attack by sending a flood of DNS requests. When the limit for a client connection is reached, subsequent DNS requests on the connection are dropped till the outstanding request count goes below the limit. This limit does not apply to the requests that the NetScaler appliance serves out of its cache.

The default value for this parameter is 255. This default value is sufficient in most scenarios. If the name servers serve a large number of concurrent DNS requests under normal operating conditions, you can specify either a large value or a value of zero (0). A value of 0 disables this feature and specifies that there is no limit to the number of DNS requests that are allowed on a single client connection. This is a global parameter and applies to all the DNS virtual servers that are configured on the NetScaler appliance.

To specify the maximum number of concurrent DNS requests allowed on a single client connection by using the command line interface

At the command prompt, type the following commands to specify the maximum number of concurrent DNS requests allowed on a single client connection and verify the configuration:

- set dns parameter -maxPipeline <positive_integer>
- show dns parameter

Example

```
> set dns parameter -maxPipeline 1000
 Done
> show dns parameter
      DNS parameters:
      DNS retries: 5
      .
      .
      .
      Max DNS Pipeline Requests: 1000
 Done
>
```

To specify the maximum number of concurrent DNS requests allowed on a single client connection by using the configuration utility

1. Navigate to Traffic Management > DNS.
2. In the details pane, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, specify a value for Max DNS Pipeline Requests.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NetScaler as an End Resolver

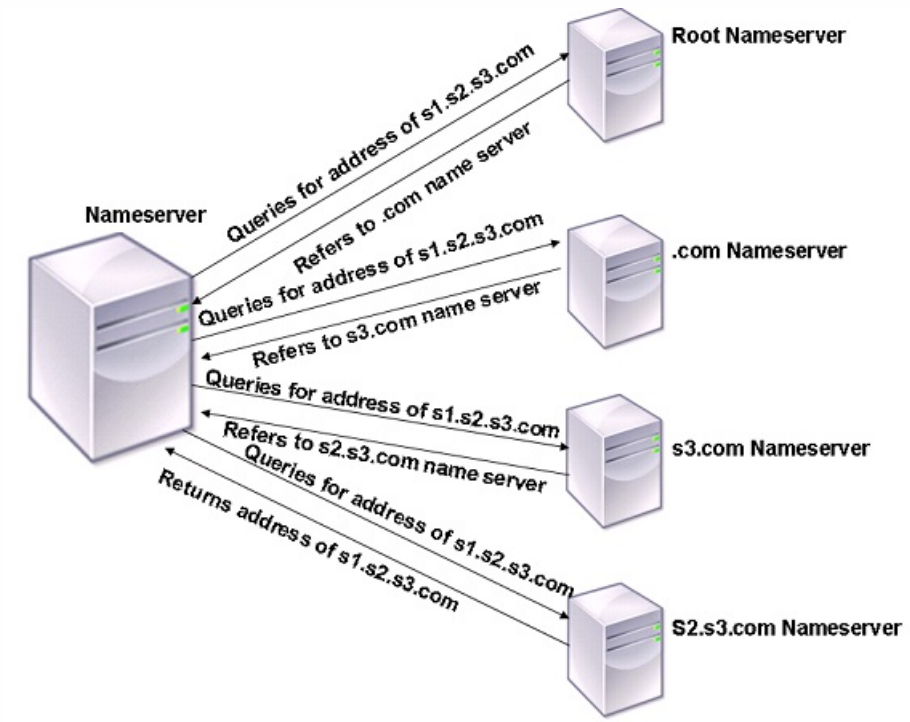
May 26, 2015

A resolver is a procedure that is invoked by an application program that translates a domain/host name to its resource record. The resolver interacts with the LDNS, which looks up the domain name to obtain its IP address. The NetScaler can provide end-to-end resolution for DNS queries.

In recursive resolution, the NetScaler appliance queries different name servers recursively to access the IP address of a domain. When the NetScaler receives a DNS request, it checks its cache for the DNS record. If the record is not present in the cache, it queries the root servers configured in the ns.conf file. The root name server reports back with the address of a DNS server that has detailed information about the second-level domain. The process is repeated until the required record is found.

When you start the NetScaler appliance for the first time, 13 root name servers are added to the ns.conf file. The NS and Address records for the 13 root servers are also added. You can modify the ns.conf file, but the NetScaler does not allow you to delete all 13 records; at least one name server entry is required for the appliance to perform name resolution. The following diagram illustrates the process of name resolution.

Figure 1. Recursive Resolution



In the process shown in the diagram, when the name server receives a query for the address of s1.s2.s3.com, it first checks the root name servers for s1.s2.s3.com. A root name server reports back with the address of the .com name server. If the address of s1.s2.s3.com is found in the name server, it responds with a suitable IP address. Otherwise, it queries other name servers for s3.com, then for s2.s3.com to retrieve the address of s1.s2.s3.com. In this way, resolution always starts from root name servers and ends with the domain's authoritative name server.

Note: For recursive resolution functionality, caching should be enabled.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

This document includes the following information:

- Enabling Recursive Resolution
- Setting the Number of Retries

Enabling Recursive Resolution

Updated: 2013-08-27

To configure the NetScaler appliance to function as an end resolver, you must enable recursive resolution on the appliance.

To enable recursive resolution by using the command line interface

At the command prompt, type the following commands to enable recursive resolution and verify the configuration:

- set dns parameter -recursion ENABLED
- show dns parameter

Example

```
> set dns parameter -recursion ENABLED
 Done
> show dns parameter
      DNS parameters:
   .
   .
   .
      Recursive Resolution : ENABLED
   .
   .
   .
 Done
>
```

To enable recursive resolution by using the configuration utility

1. Navigate to Traffic Management > DNS.
2. In the details pane, under Settings, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, select the Enable recursion check box, and then click OK.

Setting the Number of Retries

Updated: 2013-08-27

The NetScaler appliance can be configured to make a preconfigured number of attempts (called DNS retries) when it does not receive a response from the server to which it sends a query. By default, the number of DNS retries is set to 5.

To set the number of DNS retries by using the command line interface

At the command prompt, type the following commands to set the number of retries and verify the configuration:

- set dns parameter -retries <positive_integer>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- show dns parameter

Example

```
> set DNS parameter -retries 3
 Done
> show dns parameter
      DNS parameters:
      DNS retries: 3
   .
   .
   .
 Done
>
```

To set the number of retries by using the configuration utility

1. Navigate to Traffic Management > DNS.
2. In the details pane, under Settings, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, in the DNS Retries text box, type the DNS resolver request retry count, and then click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the NetScaler as a Forwarder

Mar 15, 2012

A forwarder is a server that forwards DNS queries to DNS servers that are outside the forwarder server's network. Queries that cannot be resolved locally are forwarded to other DNS servers. A forwarder accumulates external DNS information in its cache as it resolves DNS queries. To configure the NetScaler as a forwarder, you must add an external name server (a name server other than the Citrix NetScaler appliance).

The NetScaler appliance allows you to add external name servers to which it can forward the name resolution queries that cannot be resolved locally. To configure the NetScaler appliance as a forwarder, you must add the name servers to which it should forward name resolution queries. You can specify the lookup priority to specify the name service that the NetScaler appliance must use for name resolution.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding a Name Server

Oct 26, 2017

You can create a name server by specifying its IP address or by configuring an existing virtual server as the name server.

While adding name servers, you can provide an IP address or a virtual IP address (VIP). If you add an IP address, the NetScaler load balances requests to the configured name servers in round robin method. If you add a VIP, you can configure any load balancing method.

Note: To verify the configuration, you can also use the sh dns <recordtype> <domain> command. If the queried records are not present in the cache, the resource records are fetched from the configured external name servers.

To add a name server (when the NetScaler appliance acts as a forwarder) by using the command line interface

At the command prompt, type;

add dns nameServer ((<IP>) | <dnsVserverName>)

Example:

- add dns nameServer 10.102.9.20
- add dns nameServer dnsVirtualNS

To add a name server (when the NetScaler appliance acts as a resolver) by using the command line interface

At the command prompt, type:

add dns nameServer ((<IP> [-local]) | <dnsVserverName>)

Example

add dns nameServer 10.102.9.19 -local

Note

- To remove a name server, at the NetScaler command prompt, type the **rm dns nameServer** command followed by the IP address of the name server.
- To view the details of the DNS nameserver, at the NetScaler command prompt, type **show dns nameServer** command followed by the IP address of the name server.

To add a name server by using the configuration utility

Navigate to **Traffic Management** > **DNS** > **Name Servers** and create a name server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Presently, the NetScaler appliance supports name servers over UDP only and thus the DNS response size is limited to 512 bytes. If the DNS response size exceeds 512 bytes, the truncated bit is set in the DNS response. The NetScaler appliance ignores such messages.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting DNS Lookup Priority

Aug 27, 2013

You can set the lookup priority to either DNS or WINS. This option is used in the SSL VPN mode of operation.

To set the lookup priority to DNS by using the command line interface

At the command prompt, type the following commands to set the lookup priority to DNS and verify the configuration:

- set dns parameter -nameLookupPriority (DNS | WINS)
- show dns parameter

Example

```
> set dns parameter -nameLookupPriority DNS
 Done
> show dns parameter
   .
   .
   .
      Name lookup priority : DNS
   .
   .
   .
 Done
>
```

To set lookup priority to DNS by using the configuration utility

1. Navigate to Traffic Management > DNS.
2. In the details pane, under Settings, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, under Name Lookup Priority, select DNS or WINS, and then click OK.

Note: If the DNS virtual server that you have configured is DOWN and if you set the -nameLookupPriority to DNS, the NetScaler does not attempt WINS lookup. Therefore, if a DNS virtual server is not configured or is disabled, set the -nameLookupPriority to WINS.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Disabling and Enabling Name Servers

Aug 27, 2013

The following procedure describes the steps to enable or disable an existing name server.

To enable or disable a name server by using the command line interface

At the command prompt, type the following commands to enable or disable a name server and verify the configuration:

- (enable | disable) dns nameServer <IPAddress>
- show dns nameServer <IPAddress>

Example

```
> disable dns nameServer 10.102.9.19
 Done
> show dns nameServer 10.102.9.19
1)      10.102.9.19: LOCAL  -  State: OUT OF SERVICE
 Done
>
```

To enable or disable a name server by using the configuration utility

1. Navigate to Traffic Management > DNS > Name Servers.
2. In the details pane, select the name server that you want to enable or disable.
3. Click Enable or Disable. If a name server is enabled, the Disable option is available. If a name server is disabled, the Enable option is available.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DNS Logging

Jun 29, 2015

You can configure the NetScaler appliance to log the DNS requests and responses that it handles. The appliance logs the DNS requests and responses in SYSLOG format. You can choose to log either DNS requests or DNS responses, or both, and send the syslog messages to a remote log server. The log messages can be used to:

- Audit the DNS responses to the client
- Audit DNS clients
- Detect and prevent DNS attacks
- Troubleshoot

A NetScaler appliance can log the following sections in the DNS request or response, on the basis of your configuration:

- Header Section
- Questions Section
- Answer Section
- Authority Section
- Additional Section

DNS Profiles

You can use a DNS profile to configure various DNS parameters that you want the DNS endpoint to apply to the DNS traffic. In the profile, you can enable logging, caching and negative caching.

Important: From NetScaler 11.0 release, enabling DNS caching using global DNS parameters has been deprecated. You can enable or disable DNS caching using DNS profiles. You can now enable DNS caching for an individual virtual server by enabling DNS caching in a DNS profile and setting the DNS profile to the individual virtual server.
DNS profiles support the following types of DNS logging:

- DNS Query Logging
- DNS Answer Section Logging
- DNS Extended Logging
- DNS Error Logging

DNS Query Logging

You can configure a NetScaler appliance to log only the DNS queries that are received by the DNS endpoints on the appliance.

Note: If errors occur during processing of a query, they are logged if this option is set in the DNS profile.

Following is an example of a query log message:

```
DNS DNS_QUERY 143 0 : U:10.102.27.70#61297:10.102.27.73#53/22142/Q/
(RD)/NO/1/0/0/0#test.com./1#
```

DNS Answer Section Logging

You can configure a NetScaler appliance to log all the Answer sections in the DNS responses that appliance sends to the client. DNS Answer Section logging is very useful when NetScaler is configured as a DNS resolver, or in GLSB use cases.
Following is an example of a DNS answer section log:

```
DNS DNS_RESPONSE 6678 0 : U:100.100.100.210#32776:100.100.100.10#
53/61373/Q/(RD,AA,RA,R)/NO/1/1/2/4#n1.citrix.com1./
28#ANS#AAAA/120/1111:2345:6789:ffab:abcd:effa:1234:3212##
```

DNS Extended logging

To configure a NetScaler appliance to log Authority and Additional sections in the DNS responses, enable Extended logging with Answer Section logging.

Note: If errors occur during processing of either queries or responses, the errors are logged if this option is set in the DNS profile.

Following is an example of a message logged when the cache lookup is completed and the response is embedded in the packet:

```
DNS DNS_RESPONSE 2252 0 : T:100.100.100.118#21411:100.100.100.10
#53/48537/Q/(RD,AA,CD,RA,R)/NO/1/1/2/6#a1.citrix.com1./1#ANS#A/
120/1.1.1.1##AUTH#citrix.com1/NS/120/n2.citrix.com1#n1.citrix.com1##ADD#n1.citrix.com1
/A/120/1.1.1.1#1.1.1.2##n1.citrix.com1/AAAA/120/
1111:2345:6789:ffab:abcd:effa:1234:3212##n2.citrix.com1/A/120/2.1.1.2
##n2.citrix.com1/AAAA/120/2222:faff:3212:8976:123:1241:64:ff9b##OPT/0/1280/DO##
```

DNS Error Logging

You can configure a NetScaler appliance to log the errors or failures that occur when it processes a DNS query or response. For these errors, the appliance logs the DNS header, Question sections and OPT records.

Following is an example of a message logged when an error occurs during processing of a DNS request or response:

```
DNS DNS_ERROR 149 0 : U:10.102.27.70#27832:10.102.27.73#53/61153/Q/
(RD)/NO/1/0/0/0#test.com./1140#Packet Dropped
```

Policy Based Logging

You can configure custom logging based on DNS expressions by configuring the logAction on DNS policies, Rewrite, or Responder policies. You can specify that logging occurs only when a particular DNS policy evaluates to true. For more information, see Configuring Policy Based Logging for DNS.

Understanding the NetScaler Syslog Log Message Format

NetScaler appliance log DNS requests and responses in the following Syslog format:

```
<transport> :<client IP>#<client ephemeral port>:<DNS endpoint IP>#<port>
 : <query id> /opcode/header flags/rcode/question section count/answer section count
/ auth section count / additional section count #<queried domain name>
/<queried type>#...
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following table describes the fields in the Syslog message format:

Field	Values
<transport>	• T = TCP • U = UDP
<client IP>#< client ephemeral port >	DNS client IP address and port number
<DNS endpoint IP>#<port>	NetScaler DNS endpoint IP address and port number
<query id>	Query ID
<opcode>	Operation code. Supported Values: • Q : query • I : inverse query • S: status • X0: unassigned • N : notify • U : update • X1-10: unassigned values
<header flags>	Flags. Supported Values: • RD : recursion desired • TC : truncated • AA : authoritative response • CD : check disabled • AD : authenticated data • Z : unassigned • RA : recursion available • R : response
<rcode>	Response Code. Supported Values: • NO : no error • F format error • S : server failure • NX : non-existent domain • NI : not implemented • R: query refused • YX : Name Exists when it should not • YXR : RR Set Exists when it should not • NXR: RR Set that should exist does not • NAS : Server Not Authoritative for zone • NA : Not Authorized • NZ : Name not contained in zone • X1-5: unassigned
/question section count/answer section count/auth section count/additional section count	Question section, Authority section count, and Additional section count in the DNS request
<queried domain name>/<queried type>	Queried domain and queried type in the DNS request
#ANS#<record type>/<ttl>/.. #AUTH#<domain name>/<record type>/<ttl>.. #ADD# <domain name>/<record type>/<ttl>...	In case of DNS responses: • Answer Section is logged if answer section logging is enabled in the DNS profile. • Authority and Additional sections are logged if extended logging is enabled in the DNS profile.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Field	Values
	The log format would differ depending on the type of record. For more information see Understanding the Record Logging Format. • ANS: answer section • AUTH: authority • ADD: Additional section
OPT/<edns version>/UDP max payload size/DO	OPT record format in the DNS log

Understanding the Record Logging Format

Following is an example of the record logging format in a Syslog message:

<domainname>/<record type>/ <record ttl> / <resource record data>#<resource record data>#.....##

where:

Record Type	Sample Format
Address (A) record	A/5/1.1.1.1#1.1.1.2#1.1.1.3##
AAAA record	AAAA/5/1::1#1::2#1::3##
SOA record	SOA/3600/ns1.dnslogging.test./root.dnslogging.test./100/3600/3/3600/5##
NS record	NS/5/ns1.dnslogging.test
MX record	#MX/5/10/host1.dnslogging.test.#11/host2.dnslogging.test.##
CNAME record logging	CNAME/5/host1.dnslogging.test.##
SRV record	SRV/5/1/2/3/host1.dnslogging.test.#4/5/6/host2.dnslogging.test.##
TXT record	TXT/5/dns+logging##
NAPTR record	NAPTR/5/10/11////dnslogging#20/21/R/SIP//sip.dnslogging.test##
DNSKEY record	DNSKEY/5/1/3/5/AwEAAanP0K+i5bfv5SU478L760EjDjnPqI2Ccx6JZgiDBZhSONP29GfO2bkP056xp7+9Wz8X2oo5sANaDwSzUVR0YtZdPw23gAaktH6pFvnwcIHa/PTFw5VcXyiUaDc+Ana
PTR record	PTR/3600/test.com.#test4.com.##

Limitations of DNS Logging

DNS Logging has the following limitations:

- If response logging is enabled, only the following record types are logged:
  - Address (A) record
  - AAAA record
  - SOA record
  - NS record
  - MX record
  - CNAME record
  - SRV record
  - TXT record
  - NAPTR record
  - DNSKEY record
  - PTR record

  For all other record types, only L3/L4 parameters, DNS Header, and Question section are logged.
- RRSIG records are not logged even if response logging is enabled.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- DNS64 is not supported.
- DNS proactive update requests or responses are logged according to the settings in the default profile.
- On the virtual server, if sessionless option and response logging is enabled, L3/L4 parameters, DNS Header, and DNS Question section are logged instead of the response.
- The maximum size of the syslog message is 1024 bytes.
- If you have set DNS profile for a DNS policy with action type Rewrite Response, NetScaler appliance does not log the query or the manipulated responses. To log the required information you need to use audit message action in the DNS policy.
- DNS transactions that are due to DNS monitoring traffic are not logged.

Configuring DNS Logging

Following is an overview of configuring DNS logging:

1. Create a Syslog action and enable DNS in the action.
2. Create a Syslog policy and specify the Syslog action in the policy.
3. Globally bind the Syslog policy to enable logging of all NetScaler system events. Or, bind the Syslog policy to a specific load balancing virtual server.
4. Create a DNS profile and define any of the following type of logging that you want to enable:
   - DNS Query Logging
   - DNS Answer Section Logging
   - DNS Extended Logging
   - DNS Error Logging
5. Configure any of the following, based on your requirement:
   - DNS service and virtual server for DNS
   - ADNS service
   - NetScaler as a forwarder
   - NetScaler as a resolver
6. Set the created DNS profile to one of the DNS entities.

To configure DNS logging for NetScaler configured as DNS Proxy by using the command line interface

1. Add a syslog action and enable DNS in the action. At the command prompt, type:
   add **audit syslogAction** <name> (<serverIP> | -**lbVserverName** <string>) [-**serverPort** <port>] -**logLevel** <logLevel> ... [-**dateFormat** <dateFormat>] [-**logFacility** <logFacility>] [-**tcp** (**NONE** | **ALL**)] [-**acl** (**ENABLED** | **DISABLED**)] [-**timeZone** (**GMT_TIME** | **LOCAL_TIME**)] [-**userDefinedAuditlog** (**YES** | **NO**)] [-**appflowExport** (**ENABLED** | **DISABLED**)] [-**lsn** (**ENABLED** | **DISABLED**)] [-**alg** (**ENABLED** | **DISABLED**)] [-**transport** (**TCP** | **UDP**)] [-**tcpProfileName** <string>] [-**maxLogDataSizeToHold** <positive_integer>] [-**dns** (**ENABLED** | **DISABLED**)]

   Example:

   add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone LOCAL_TIME -dns ENABLED

2. Create a syslog policy and specify the created syslog action in the policy. At the command prompt, type:
   add **audit syslogPolicy** <name> <rule> <action>

   Example:

   add audit syslogPolicy syslogpol1 ns_true nssyslogact1

3. Bind the syslog policy globally. At the command prompt, type:
   bind **system global** [<policyName> [-**priority** <positive_integer>]]

   Example:

   bind system global syslogpol1

4. Create a DNS profile and enable any of the following type of logs that you want to configure:
   - DNS Query Logging
   - DNS Answer Section Logging
   - DNS Extended Logging
   - DNS Error Logging

   At the command prompt, type:

   add **dns profil**e <dnsProfileName> [-**dnsQueryLogging** (**ENABLED** | **DISABLED**)] [-**dnsAnswerSecLogging** (**ENABLED** | **DISABLED**)] [-**dnsExtendedLogging** (**ENABLED** | **DISABLED**)] [-**dnsErrorLogging** (**ENABLED** | **DISABLED**)] [-**cacheRecords** (**ENABLED** | **DISABLED**)] [-**cacheNegativeResponses** (**ENABLED** | **DISABLED**)]

   Example:

   add dns profile dnsprofile1 -dnsQueryLogging ENABLED

5. Configure service of type DNS. At the command prompt, type:
   add **service** <name> <serverName> <serviceType> <port>

   Example:

   add service svc1 10.102.84.140 dns 53

6. Configure a load balancing virtual server of service type DNS.
   add **lb vserver** <name> <serviceType> <ip> <port>

   Example:

   add lb vserver lb1 dns 100.100.100.10 53

7. Bind the service to the virtual server. At the command prompt, type:
   bind **lb vserver** <name> <serviceName>

   Example:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

bind lb vserver lb1 svc1

8. Set the created DNS profile to the virtual server. At the command prompt, type:
   set **lb vserver** <name> [- **dnsProfileName** <string>]

   Example:

   set lb vserver lb1 –dnsProfileName dnsprofile1

Sample DNS Logging Configuration for NetScaler Appliance Configured as DNS Proxy

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel
CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
LOCAL_TIME -dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add lb vserver lb1 dns 100.100.100.10 53 –dnsProfileName dnsprofile1
Done
> add service svc1 10.102.84.140 dns 53
Done
> bind lb vserver lb1 svc1
Done
```

Sample DNS Logging Configuration for NetScaler Appliance Configured as ADNS

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add lb vserver lb1 dns 100.100.100.10 53 –dnsProfileName dnsprofile1
Done
> add service svc1 10.102.84.140 dns 53
Done
> bind lb vserver lb1 svc1
Done
```

Sample DNS Logging Configuration for NetScaler Appliance Configured as a Forwarder

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> Add dns nameserver 8.8.8.8 –dnsProfileName dnsprofile1
Done
```

Sample DNS Logging Configuration for NetScaler Appliance Configured as a Resolver

```
> add audit syslogAction nssyslogact1 10.102.151.136
-logLevel CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4
-timeZone LOCAL_TIME -dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> set dns parameter -recursion enABLED
Done
> add nameserver 1.1.1.100 -local dnsProfileName dnsprofile1
Done
```

Configuring Policy Based Logging for DNS

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy based logging enables you to specify a format for log messages. The contents of a log message are defined by using a default syntax expression. When the message action specified in the policy is performed, the NetScaler appliance constructs the log message from the expression and writes the message to the log file. You can configure the appliance to log only when a particular DNS policy evaluates to True.

Note: If you have set a DNS policy with a DNS profile for the request side, NetScaler appliance logs only the query.

To configure policy based logging for a DNS policy, you must first configure an audit message action. For more information about configuring an audit message action, see Configuring Policy-Based Logging. After configuring the audit message action, specify the message action in a DNS policy.

To configure policy based logging for a DNS policy by using the command line interface

At the command prompt, type the following commands to configure policy based logging for a DNS policy and verify the configuration:

- add **dns action** <actionName> <actionType> [-**IPAddress** <ip_addr|ipv6_addr> ... | -**viewName** <string> | -**preferredLocList** <string> ...] [-**TTL** <secs>] [-**dnsProfileName** <string>]
- set **dns policy** <name> [<rule>] [-**actionName** <string>] [-**logAction** <string>]
- show **dns policy** [<name>]

Example 1:

In a GSLB deployment, if you want to respond with different IP addresses to the client requests coming from a particular subnet, instead of responding with IP addresses used for general purposes (such as the IP addresses of internal users), you can configure a DNS policy with the action type as DNS view. In this case, you can configure DNS logging on the specified DNS action such that you can log the specific responses.

For example:

```
> add dns profile dns_prof1 -dnsqueryLogging enABLED -dnsanswerSecLogging enABLED
 Done
> add dns view dns_view1
 Done
> add dns action dns_act1 viewName -view dns_view1 –dnsprofilename dns_prof1
 Done
> add dns policy dns_pol1 "CLIENT.IP.SRC.APPLY_MASK(255.255.255.0).EQ(100.100.100.0)”
dns_act1
 Done
> bind dns global dns_pol1 100 -gotoPriorityExpression END -type REQ_DEFAULT
 Done
> bind gslb service site_1_svc -viewName dns_view1 123.1.1.1
 Done
> bind gslb service site_5_svc -view dns_view1 132.1.1.1
 Done
```

Note: In the above configuration, if you query for the domain configured on a GSLB virtual server, for example, *sampletest.com*, all the internal users of subnet 100.100.100.0/24 are served with the DNS view IP addresses, and the responses are logged. Client requests for other subnets are not logged.

Example 2:

If you want to log only the queries for the domain *example.com*, you can create a DNS profile with query logging enabled and set the DNS profile to a DNS action with the action type **NOOP**, and then create a DNS policy and set the DNS action. For example:

```
>add dns profile query_logging -dnsqueryLogging ENABLED
Done
>add dns action dns_act1 NOOP -dnsprofileName query_logging
Done
>add dns policy dns_pol1 DNS.REQ.QUESTION.DOMAIN.EQ("example.com")  dns_act1
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DNS Suffixes

Oct 29, 2014

You can configure DNS suffixes that enable the NetScaler appliance to complete non-fully qualified domain names (non-FQDNs) during name resolution. For example, during the process of resolving the domain name abc (which is not fully qualified), if a DNS suffix example.com is configured, the appliance appends the suffix to the domain name (abc.example.com) and resolves it. If DNS suffixes are not configured, the appliance appends a period to the non-FQDNs and resolves the domain name.

Creating DNS Suffixes

DNS suffixes have significance and are valid only when the NetScaler is configured as an end resolver or forwarder. You can specify a suffix of up to 127 characters.

To create DNS suffixes by using the command line interface

At the command prompt, type the following commands to create a DNS suffix and verify the configuration:

- add dns suffix <dnsSuffix>
- show dns suffix <dnsSuffix>

Example

```
> add dns suffix example.com
 Done
> show dns suffix example.com
1)     Suffix: example.com
 Done
>
```

To remove a DNS suffix by using the NetScaler command line, at the NetScaler command prompt, type the rm dns suffix command and the name of the DNS suffix.

To create DNS suffixes by using the configuration utility

Navigate to Traffic Management > DNS > DNS Suffix and create DNS suffixes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

DNS ANY Query

Mar 15, 2012

An ANY query is a type of DNS query that retrieves all records available for a domain name. The ANY query must be sent to a name server that is authoritative for a domain.

Behavior in ADNS Mode

In the ADNS mode, the NetScaler appliance returns the records held in its local cache. If there are no records in the cache, the appliance returns the NXDOMAIN (negative) response.

If the NetScaler can match the domain delegation records, it returns the NS records. Otherwise, it returns the NS records of the root domain.

Behavior in DNS Proxy Mode

In proxy mode, the NetScaler appliance checks its local cache. If there are no records in the cache, the appliance passes the query to the server.

Behavior for GSLB Domains

Updated: 2013-08-26

If a GSLB domain is configured on the NetScaler appliance and an ANY query is sent for the GSLB domain (or GSLB site domain), the appliance returns the IP address of the GSLB service that it selects through the Load Balancing decision. If the multiple IP response (MIR) option is enabled, the IP addresses of all GSLB services are sent.

For the NetScaler to return these records when it responds to the ANY query, all records corresponding to a GSLB domain must be configured on the NetScaler.

Note: If records for a domain are distributed between the NetScaler and a server, only records configured on the NetScaler are returned.
The NetScaler provides the option to configure DNS views and DNS policies. These are used for performing global server load balancing. For more information, see Global Server Load Balancing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configure Negative Caching of DNS Records

Aug 12, 2016

The NetScaler appliance supports caching of negative responses for a domain. A negative response indicates that information about a requested domain does not exist, or that the server cannot provide an answer for the query. The storage of this information is called negative caching. Negative caching helps speed up responses to queries about a domain.

> **Note**
>
> Negative caching is supported only when the back-end server is configured as an authoritative DNS (ADNS) server for the queried domain.

A negative response can be one of the following:

- NXDOMAIN error message—If a negative response is present in the local cache, the NetScaler returns an error message (NXDOMAIN). If the response is not in the local cache, the query is forwarded to the server, and the server returns an NXDOMAIN error to the NetScaler appliance. The appliance caches the response locally, then returns the error message to the client.
- NODATA error message—If the domain name in query is valid but records of the given type are not available, the appliance sends a NODATA error message.

When negative caching is enabled, the appliance caches the negative response from the DNS server and serves the future requests from the cache only. This helps speed up responses to queries and also to reduce the DNS traffic. Negative caching can be used in all deployments, that is, when a NetScaler appliance is serving as a proxy, as an end resolver, or as a forwarder.

You can enable or disable negative caching using DNS profile, for more information see, DNS Profiles. By default, negative caching is enabled in the default DNS profile (*default-dns-profile*) that are bound by default to a DNS virtual server or in the newly created DNS profile.

To enable or disable negative caching by using the command line interface

At the command prompt, type the following commands to enable or disable negative caching and verify the configuration:

- add dns profile <dnsProfileName> [-cacheRecords (ENABLED | DISABLED)] [-cacheNegativeResponses (ENABLED | DISABLED)]
- show dns profile [<dnsProfileName>]

Example of a default DNS profile

```
> sh dns profile default-dns-profile
1)  default-dns-profile
 Query logging : DISABLED  Answer section logging : DISABLED
 Extended logging : DISABLED  Error logging : DISABLED
 Cache Records : ENABLED  Cache Negative Responses: ENABLED
Done
```

Example of a newly created DNS profile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> add dnsprofile dns_profile1 -cacheRecords ENABLED -cacheNegativeResponses ENABLED
Done
> show dns profile dns_profile1
1)  dns_profile1
 Query logging : DISABLED  Answer section logging : DISABLED
 Extended logging : DISABLED  Error logging : DISABLED
 Cache Records : ENABLED  Cache Negative Responses: ENABLED
 Done
```

To specify service or virtual server level DNS parameters by using the command line interface

At the command prompt, perform the following:

1. Configure the DNS profile.
   add dns profile <dnsProfileName> [-cacheRecords (ENABLED | DISABLED)] [-cacheNegativeResponses (ENABLED | DISABLED)]

2. Bind the DNS profile to the service or virtual server.
   To bind the DNS profile to the service:

   set service <name> [-dnsProfileName <string>]

   Example
   >set service service1 -dnsProfileName dns_profile1
   Done
   To bind the DNS profile to the virtual server:

   set lb vserver <name> [-dnsProfileName <string>]

   Example
   >set lb vserver lbvserver1 -dnsProfileName dns_profile1
   Done

To specify service or virtual server level DNS parameters by using the configuration utility

1. Configure the HTTP profile.
   Navigate to **System** > **Profiles**> **DNS Profile**, and create the DNS profile.

   .
2. Bind the HTTP profile to the service or virtual server.
   Navigate to **Traffic Management** > **Load Balancing**> **Services/Virtual Servers**, and create the DNS profile, which should be bound to the service/virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Domain Name System Security Extensions

Aug 27, 2014

DNS Security Extensions (DNSSEC) is an Internet Engineering Task Force (IETF) standard that aims to provide data integrity and data origin authentication in communications between name servers and clients while still transmitting User Datagram Protocol (UDP) responses in clear text. DNSSEC specifies a mechanism that uses asymmetric key cryptography and a set of new resource records that are specific to its implementation.

The DNSSEC specification is described in RFC 4033, “DNS Security Introduction and Requirements,” RFC 4034, “Resource Records for the DNS Security Extensions,” and RFC 4035, “Protocol Modifications for the DNS Security Extensions.” The operational aspects of implementing DNSSEC within DNS are discussed in RFC 4641, “DNSSEC Operational Practices.”

You can configure DNSSEC on the Citrix® NetScaler® ADC. You can generate and import keys for signing DNS zones. You can configure DNSSEC for zones for which the NetScaler ADC is authoritative. You can configure the ADC as a DNS proxy server for signed zones hosted on a farm of backend name servers. If the ADC is authoritative for a subset of the records belonging to a zone for which the ADC is configured as a DNS proxy server, you can include the subset of records in the DNSSEC implementation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DNSSEC

May 26, 2015

Configuring DNSSEC involves enabling DNSSEC on the Citrix® NetScaler® appliance, creating a Zone Signing Key and a Key Signing Key for the zone, adding the two keys to the zone, and then signing the zone with the keys.

The NetScaler ADC does not act as a DNSSEC resolver. DNSSEC on the ADC is supported only in the following deployment scenarios:

1. ADNS—NetScaler is the ADNS and generates the signatures itself.
2. Proxy—NetScaler acts as a DNSSEC proxy. It is assumed that the NetScaler is placed in front of the ADNS/LDNS servers in a trusted mode. The ADC acts only as a proxy caching entity and does not validate any signatures.

This document includes the following information:

- Enabling and Disabling DNSSEC
- Creating DNS Keys for a Zone
- Publishing a DNS Key in a Zone
- Configuring a DNS Key
- Signing and Unsigning a DNS Zone
- Viewing the NSEC Records for a Given Record in a Zone
- Removing a DNS Key

Enabling and Disabling DNSSEC

Updated: 2014-08-27

You must enable DNSSEC on the NetScaler ADC for the ADC to respond to DNSSEC-aware clients. By default, DNSSEC is enabled.

You can disable the DNSSEC feature if you do not want the NetScaler ADC to respond to clients with DNSSEC-specific information.

To enable or disable DNSSEC by using the command line interface

At the command prompt, type the following commands to enable or disable DNSSEC and verify the configuration:

- set dns parameter -dnssec (ENABLED | DISABLED)
- show dns parameter

Example

```
> set dns parameter -dnssec ENABLED
 Done
> show dns parameter
      DNS parameters:
      DNS retries: 5
   .
   .
   .
   DNSEC Extension: ENABLED
      Max DNS Pipeline Requests: 255
 Done
>
```

To enable or disable DNSSEC by using the configuration utility

1. Navigate to Traffic Management > DNS.
2. In the details pane, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, select or clear the Enable DNSSEC Extension check box.

Creating DNS Keys for a Zone

Updated: 2014-10-29

For each DNS zone that you want to sign, you must create two pairs of asymmetric keys. One pair, called the Zone Signing Key, is used to sign all the resource record sets in the zone. The second pair is called the Key Signing Key and is used to sign only the DNSKEY resource records in the zone.

When the Zone Signing Key and Key Signing Key are created, the suffix .key is automatically appended to the names of the public components of the keys and the suffix .private is automatically appended to the names of their private components.

Additionally, the NetScaler ADC also creates a Delegation Signer (DS) record and appends the suffix .ds to the name of the record. If the parent zone is a signed zone, you must publish the DS record in the parent zone to establish the chain of trust.

When you create a key, the key is stored in the /nsconfig/dns/ directory, but it is not automatically published in the zone. After you create a key by using the create dns key command,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

you must explicitly publish the key in the zone by using the add dns key command. The process of generating a key has been separated from the process of publishing the key in a zone to enable you to use alternative means to generate keys. For example, you can import keys generated by other key-generation programs (such as bind-keygen) by using Secure File Transfer Protocol (SFTP) and then publish the keys in the zone. For more information about publishing a key in a zone, see Publishing a DNS Key in a Zone.

Perform the steps described in this topic to create a Zone Signing Key and then repeat the steps to create a Key Signing Key. The example that follows the command syntax first creates a Zone Signing Key pair for the zone example.com. The example then uses the command to create a Key Signing Key pair for the zone.

To create a DNS key by using the command line interface

At the NetScaler command prompt, type the following command to create a DNS key:

create dns key -zoneName <string> -keyType <keyType> -algorithm RSASHA1 -keySize <positive_integer> -fileNamePrefix <string>

Example

```
> create dns key -zoneName example.com -keyType zsk -algorithm RSASHA1 -keySize 1024 -fileNamePrefix example.com.zsk.rsasha1.1024
File Name: /nsconfig/dns/example.com.zsk.rsasha1.1024.key (public); /nsconfig/dns/example.com.zsk.rsasha1.1024.private (private); /nsconfig/dns/example.com.zsk.rsasha1.1024.ds (ds)
This operation may take some time, Please wait...
 Done
> create dns key -zoneName example.com -keyType ksk -algorithm RSASHA1 -keySize 4096 -fileNamePrefix example.com.ksk.rsasha1.4096
File Name: /nsconfig/dns/example.com.ksk.rsasha1.4096.key (public); /nsconfig/dns/example.com.ksk.rsasha1.4096.private (private); /nsconfig/dns/example.com.ksk.rsasha1.4096.ds (ds)
This operation may take some time, Please wait...
 Done
>
```

To create a DNS key by using the configuration utility

1. Navigate to Traffic Management > DNS.
2. In the details area, click **Create DNS Key** and create a DNS key.
   Note: For File Name Prefix, if you want to modify the file name prefix of an existing key, click the arrow next to the Browse button, click either Local or Appliance (depending on whether the existing key is stored on your local computer or in the /nsconfig/dns/ directory on the appliance), browse to the location of the key, and then double-click the key. The File Name Prefix box is populated with only the prefix of the existing key. Modify the prefix accordingly.

Publishing a DNS Key in a Zone

Updated: 2014-10-29

A key (Zone Signing Key or Key Signing Key) is published in a zone by adding the key to the NetScaler ADC. A key must be published in a zone before you sign the zone.

Before you publish a key in a zone, the key must be available in the /nsconfig/dns/ directory. Therefore, if you used other means to generate the key—means other than the create dns key command on the NetScaler ADC (for example, by using the bind-keygen program on another computer)—make sure that the key is added to the /nsconfig/dns/ directory before you publish the key in the zone.

If the key has been generated by another program, you can import the key to your local computer and use the NetScaler configuration utility to add the key to the /nsconfig/dns/ directory. Or, you can use other means to import the key to the directory, such as the Secure File Transfer Protocol (SFTP).

You must use the add dns key command for each public-private key pair that you want to publish in a given zone. If you created a Zone Signing Key pair and a Key Signing Key pair for a zone, use the add dns key command to first publish one of the key pairs in the zone and then repeat the command to publish the other key pair. For each key that you publish in a zone, a DNSKEY resource record is created in the zone.

The example that follows the command syntax first publishes the Zone Signing Key pair (that was created for the example.com zone) in the zone. The example then uses the command to publish the Key Signing Key pair in the zone.

To publish a key in a zone by using the command line interface

At the command prompt, type the following command to publish a key in a zone and verify the configuration:

- add dns key <keyName> <publickey> <privatekey> [-expires <positive_integer> [<units>]] [-notificationPeriod <positive_integer> [<units>]] [-TTL <secs>]
- show dns zone [<zoneName> | -type <type>]

Example

```
> add dns key example.com.zsk example.com.zsk.rsasha1.1024.key example.com.zsk.rsasha1.1024.private
 Done
> add dns key example.com.ksk example.com.ksk.rsasha1.4096.key example.com.ksk.rsasha1.4096.private
 Done
> show dns zone example.com
    Zone Name : example.com
    Proxy Mode : NO
    Domain Name : example.com
        Record Types : NS SOA DNSKEY
    Domain Name : ns1.example.com
        Record Types : A
    Domain Name : ns2.example.com
        Record Types : A
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 Done
>
```

To publish a key in a DNS zone by using the NetScaler configuration utility

Navigate to Traffic Management > DNS > Keys.
Note: For Public Key and Private Key, to add a key that is stored on your local computer, click the arrow next to the Browse button, click Local, browse to the location of the key, and then double-click the key.

Configuring a DNS Key

Updated: 2014-08-27

You can configure the parameters of a key that has been published in a zone. You can modify the key's expiry time period, notification period, and time-to-live (TTL) parameters. If you change the expiry time period of a key, the NetScaler ADC automatically re-signs all the resource records in the zone with the key, provided that the zone is currently signed with the particular key.

To configure a key by using the command line interface

At the command prompt, type the following command to configure a key and verify the configuration:

- set dns key <keyName> [-expires <positive_integer> [<units>]] [-notificationPeriod <positive_integer> [<units>]] [-TTL <secs>]
- show dns key [<keyName>]

Example

```
> set dns key example.com.ksk -expires 30 DAYS -notificationPeriod 3 DAYS -TTL 3600
 Done
> show dns key example.com.ksk
1)     Key Name: example.com.ksk
       Expires: 30 DAYS       Notification: 3 DAYS    TTL: 3600
       Public Key File: example.com.ksk.rsasha1.4096.key
       Private Key File: example.com.ksk.rsasha1.4096.private
 Done
>
```

To configure a key by using the configuration utility

1. Navigate to Traffic Management > DNS > Keys.
2. In the details pane, click the key that you want to configure, and then click Open.
3. In the Configure DNS Key dialog box, modify the values of the following parameters as shown:

   - Expires—expires
   - Notification Period—notificationPeriod
   - TTL—TTL
4. Click OK.

Signing and Unsigning a DNS Zone

Updated: 2014-08-27

To secure a DNS zone, you must sign the zone with the keys that have been published in the zone. When you sign a zone, the NetScaler ADC creates a Next Secure (NSEC) resource record for each owner name. Then, it uses the Key Signing Key to sign the DNSKEY resource record set. Finally, it uses the Zone Signing Key to sign all the resource record sets in the zone, including the DNSKEY resource record sets and NSEC resource record sets. Each sign operation results in a signature for the resource record sets in the zone. The signature is captured in a new resource record called the RRSIG resource record.

After you sign a zone, you must save the configuration.

To sign a zone by using the command line interface

At the command prompt, type the following command to sign a zone and verify the configuration:

- sign dns zone <zoneName> [-keyName <string> ...]
- show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
- save config

Example

```
> sign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
       Zone Name : example.com
       Proxy Mode : NO
       Domain Name : example.com
             Record Types : NS SOA DNSKEY RRSIG NSEC
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Domain Name : ns1.example.com
        Record Types : A RRSIG NSEC
    Domain Name : ns2.example.com
        Record Types : A RRSIG
    Domain Name : ns2.example.com
        Record Types : RRSIG NSEC
Done
> save config
 Done
>
save config
```

To unsign a zone by using the command line interface

At the command prompt, type the following command to unsign a zone and verify the configuration:

- unsign dns zone <zoneName> [-keyName <string> ...]
- show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]

Example

```
> unsign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
    Zone Name : example.com
    Proxy Mode : NO
    Domain Name : example.com
        Record Types : NS SOA DNSKEY
    Domain Name : ns1.example.com
        Record Types : A
    Domain Name : ns2.example.com
        Record Types : A
 Done
>
```

To sign or unsign a zone by using the configuration utility

1. Navigate to Traffic Management > DNS > Zones.
2. In the details pane, click the zone that you want to sign, and then click Sign/Unsign.
3. In the Sign/Unsign DNS Zone dialog box, do one of the following:
   - To sign the zone, select the check boxes for the keys (Zone Signing Key and Key Signing Key) with which you want to sign the zone.
     You can sign the zone with more than one Zone Signing Key or Key Signing Key pair.
   - To unsign the zone, clear the check boxes for the keys (Zone Signing Key and Key Signing Key) with which you want to unsign the zone.
     You can unsign the zone with more than one Zone Signing Key or Key Signing Key pair.
4. Click OK.

Viewing the NSEC Records for a Given Record in a Zone

Updated: 2014-08-27

You can view the NSEC records that the NetScaler ADC automatically creates for each owner name in the zone.

To view the NSEC record for a given record in a zone by using the command line interface

At the command prompt, type the following command to view the NSEC record for a given record in a zone:

show dns nsecRec [<hostName> | -type (ADNS | PROXY | ALL)]

Example

```
> show dns nsecRec example.com
1)    Domain Name : example.com
    Next Nsec Name: ns1.example.com
    Record Types : NS SOA DNSKEY RRSIG NSEC
 Done
>
```

To view the NSEC record for given record in a zone by using the configuration utility

1. Navigate to Traffic Management > DNS > Records > Next Secure Records.
2. In the details pane, click the name of the record for which you want to view the NSEC record. The NSEC record for the record you select is displayed in the Details area.

Removing a DNS Key

Updated: 2014-08-27

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You remove a key from the zone in which it is published when the key has expired or if the key has been compromised. When you remove a key from the zone, the zone is automatically unsigned with the key. Removing the key with this command does not remove the key files present in the /nsconfig/dns/ directory. If the key files are no longer needed, they have to be explicitly removed from the directory.

To remove a key from the NetScaler ADC by using the command line interface

At the command prompt, type the following command to remove a key and verify the configuration:

- rm dns key <keyName>
- show dns key <keyName>

Example

```
> rm dns key example.com.zsk
 Done
> show dns key example.com.zsk
ERROR: No such resource [keyName, example.com.zsk]
>
```

To remove a key from the NetScaler ADC by using the configuration utility

1. Navigate to Traffic Management > DNS > Keys.
2. In the details pane, click the name of the key that you want to remove from the ADC, and then click Remove.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DNSSEC When the NetScaler ADC is Authoritative for a Zone

Aug 27, 2014

When the Citrix® NetScaler® ADC is authoritative for a given zone, all the resource records in the zone are configured on the ADC. To sign the authoritative zone, you must create keys (the Zone Signing Key and the Key Signing Key) for the zone, add the keys to the ADC, and then sign the zone, as described in Creating DNS Keys for a Zone, Publishing a DNS Key in a Zone, and Signing and Unsigning a DNS Zone, respectively.

If any global server load balancing (GSLB) domains configured on the ADC belong to the zone being signed, the GSLB domain names are signed along with the other records that belong to the zone.

After you sign a zone, responses to requests from DNSSEC-aware clients include the RRSIG resource records along with the requested resource records. DNSSEC must be enabled on the ADC. For more information about enabling DNSSEC, see Enabling and Disabling DNSSEC.

Finally, after you configure DNSSEC for the authoritative zone, you must save the NetScaler configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DNSSEC for a Zone for Which the NetScaler ADC Is a DNS Proxy Server

May 26, 2015

The procedure for signing a zone for which the Citrix® NetScaler® ADC is configured as a DNS proxy server depends on whether or not the ADC owns a subset of the zone information owned by the backend name servers. If it does, the configuration is considered a *partial zone ownership configuration*. If the ADC does not own a subset of the zone information, the NetScaler configuration for managing the backend servers is considered a *zone-less DNS proxy server configuration*. The basic DNSSEC configuration tasks for both NetScaler configurations are the same. However, signing the partial zone on the NetScaler ADC requires some additional configuration steps.

Note: The terms *zone-less proxy server configuration* and *partial zone* are used only in the context of the NetScaler appliance.
Important: When configured in proxy mode, the ADC does not perform signature verification on DNSSEC responses before updating the cache.
If you configure the ADC as a DNS proxy to load balance DNSSEC aware resolvers (servers), you must set the Recursion Available option while configuring the DNS virtual server. If a DNSSEC query arrives with Checking Disabled (CD) bit set, the query is passed on to the server with the CD bit retained, and the response from the server is not cached. In releases prior to 10.5.e build xx.x, the ADC unset the CD bit before passing it to the server and also cached the server response.

This document includes the following information:

- Configuring DNSSEC for a Zone-Less DNS Proxy Server Configuration
- Configuring DNSSEC for a Partial Zone Ownership Configuration

Configuring DNSSEC for a Zone-Less DNS Proxy Server Configuration

Updated: 2014-08-27

For a zone-less DNS proxy server configuration, zone signing must be performed on the backend name servers. On the NetScaler ADC, you configure the ADC as a DNS proxy server for the zone. You create a load balancing virtual server of protocol type DNS, configure services on the ADC to represent the name servers, and then bind the services to the load balancing virtual server. For more information about these configuration tasks, see Configuring the NetScaler as a DNS Proxy Server.

When a client sends the ADC a DNS request with the DNSSEC OK (DO) bit set, the ADC checks its cache for the requested information. If the resource records are not available in its cache, the ADC forwards the request to one of the DNS name servers, and then relays the response from the name server to the client. Additionally, the ADC caches the RRSIG resource records along with the response from the name server. Subsequent requests from DNSSEC-aware clients are served from the cache (including the RRSIG resource records), subject to the time-to-live (TTL) parameter. If a client sends a DNS request without setting the DO bit, the ADC responds with only the requested resource records, and does not include the RRSIG resource records that are specific to DNSSEC.

Configuring DNSSEC for a Partial Zone Ownership Configuration

Updated: 2014-08-27

© 1999-2017 Citrix Systems, Inc. All rights reserved.

In some NetScaler configurations, even though the authority for a zone lies with the backend name servers, a subset of the resource records that belong to the zone might be configured on the NetScaler ADC. The ADC owns (or is authoritative for) only this subset of records. Such a subset of records can be considered to constitute a *partial zone* on the ADC. The ADC owns the partial zone. All other records are owned by the backend name servers.

A typical partial zone configuration on the NetScaler ADC is seen when global server load balancing (GSLB) domains are configured on the ADC, and the GSLB domains are a part of a zone for which the backend name servers are authoritative.

Signing a zone that includes only a partial zone on the ADC involves including the partial zone information in the backend name server zone files, signing the zone on the backend name servers, and then signing the partial zone on the ADC. The same key set must be used to sign the zone on the name servers and the partial zone on the ADC.

To sign the zone on the backend name servers

1. Include the resource records that are contained in the partial zone, in the zone files of the name servers.
2. Create keys and use the keys to sign the zone on the backend name servers.

To sign the partial zone on the NetScaler ADC

1. Create a zone with the name of the zone that is owned by the backend name servers. When configuring the partial zone, set the proxyMode parameter to YES. This zone is the partial zone that contains the resource records owned by the ADC.

   For example, if the name of the zone that is configured on the backend name servers is example.com, you must create a zone named example.com on the ADC, with the proxyMode parameter set to YES. For more information about adding a zone, see Configuring a DNS Zone.

   Note: Do not add SOA and NS records for the zone. These records should not exist on the ADC for a zone for which the ADC is not authoritative.

2. Import the keys (from one of the backend name servers) to the ADC and then add them to the /nsconfig/dns/ directory. For more information about how you can import a key and add it to the ADC, see Publishing a DNS Key in a Zone.
3. Sign the partial zone with the imported keys. When you sign the partial zone with the keys, the ADC generates RRSIG and NSEC records for the resource record sets and individual resource records in the partial zone, respectively. For more information about signing a zone, see Signing and Unsigning a DNS Zone.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DNSSEC for GSLB Domain Names

Aug 27, 2014

If global server load balancing (GSLB) is configured on the Citrix® NetScaler® ADC and the ADC is authoritative for the zone to which the GSLB domain names belong, all GLSB domain names are signed when the zone is signed. For more information about signing a zone for which the ADC is authoritative, see Configuring DNSSEC When the NetScaler Appliance Is Authoritative for a Zone.

If the GSLB domains belong to a zone for which the backend name servers are authoritative, you must first sign the zone on the name servers, and then sign the partial zone on the ADC to complete the DNSSEC configuration for the zone. For more information, see Configuring DNSSEC for a Partial Zone Ownership Configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Zone Maintenance

May 26, 2015

From a DNSSEC perspective, zone maintenance involves rolling over Zone Signing Keys and Key Signing Keys when key expiry is imminent. These zone maintenance tasks have to be performed manually. The process of re-signing a zone is performed automatically and does not require manual intervention.

This document includes the following information:

- Re-Signing an Updated Zone
- Rolling Over DNSSEC Keys

Re-Signing an Updated Zone

Updated: 2014-08-27

When a zone is updated, that is, when new records are added to the zone or existing records are changed, the process of re-signing the new (or modified) record is performed automatically by the Citrix® NetScaler® ADC. If a zone contains multiple Zone Signing Keys, the ADC re-signs the new (or modified) record with the key with which the zone is signed at the point in time when the re-signing is to be performed.

Rolling Over DNSSEC Keys

Updated: 2014-08-27

On the NetScaler ADC, you can use the pre-publish and double signature methods to perform a rollover of the Zone Signing Key and Key Signing Key. More information about these two rollover methods is available in RFC 4641, “DNSSEC Operational Practices.”

The following topics map commands on the ADC to the steps in the rollover procedures discussed in RFC 4641.

The key expiry notification is sent through an SNMP trap called dnskeyExpiry. Three MIB variables, dnskeyName, dnskeyTimeToExpire, and dnskeyUnitsOfExpiry are sent along with the dnskeyExpiry SNMP trap. For more information, see *Citrix NetScaler SNMP OID Reference* at .

Pre-Publish Key Rollover

RFC 4641, “DNSSEC Operational Practices” defines four stages for the pre-publish key rollover method: initial, new DNSKEY, new RRSIGs, and DNSKEY removal. Each stage is associated with a set of tasks that you must perform on the ADC. Following are the descriptions of each stage and the tasks that you must perform. The rollover procedure described here can be used for both Key Signing Keys and Zone Signing Keys.

- **Stage 1: Initial.** The zone contains only those key sets with which the zone has currently been signed. The state of the zone in the initial stage is the state of the zone just before you begin the key rollover process.
  Example

  Consider the key, example.com.zsk1, with which the zone example.com is currently signed. The zone contains only those RRSIGs that were generated by the example.com.zsk1 key, which is due for expiry. The Key Signing Key is example.com.ksk1.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Stage 2: New DNSKEY.** A new key is created and published in the zone (that is, the key is added to the ADC), but the zone is not signed with the new key until the pre-roll phase is complete. In this stage, the zone contains the old key, the new key, and the RRSIGs generated by the old key. Publishing the new key for the complete duration of the pre-roll phase gives the DNSKEY resource record (that corresponds to the new key) enough time to propagate to the secondary name servers.
  Example

  A new key example.com.zsk2 is added to the example.com zone. The zone is not signed with example.com.zsk2 until the pre-roll phase is complete. The example.com zone contains DNSKEY resource records for both example.com.zsk1 and example.com.zsk2.

  NetScaler commands

  Perform the following tasks on the ADC:
  - Create a new DNS key by using the create dns key command.
    For more information about creating a DNS key, including an example, see Creating DNS Keys for a Zone.
  - Publish the new DNS key in the zone by using the add dns key command.
    For more information about publishing the key in the zone, including an example, see Publishing a DNS Key in a Zone.
- **Stage 3: New RRSIGs.** The zone is signed with the new DNS key and then unsigned with the old DNS key. The old DNS key is not removed from the zone and remains published until the RRSIGs that were generated by the old key expire.
  Example

  The zone is signed with example.com.zsk2 and then unsigned with example.com.zsk1. The zone continues to publish example.com.zsk1 until the RRSIGs that were generated by example.com.zsk1 expire.

  NetScaler commands

  Perform the following tasks on the ADC:
  - Sign the zone with the new DNS key by using the sign dns zone command.
  - Unsign the zone with the old DNS key by using the unsign dns zone command.

  For more information about signing and unsigning a zone, including examples, see Signing and Unsigning a DNS Zone.
- **Stage 4: DNSKEY Removal.** When the RRSIGs that were generated by the old DNS key expire, the old DNS key is removed from the zone.
  Example

  The old DNS key example.com.zsk1 is removed from the example.com zone.

  NetScaler commands

  On the ADC, you remove the old DNS key by using the rm dns key command. For more information about removing a key from a zone, including an example, see Removing a DNS Key.

Double Signature Key Rollover

RFC 4641, “DNSSEC Operational Practices” defines three stages for double signature key rollover: initial, new DNSKEY, and DNSKEY removal. Each stage is associated with a set of tasks that you must perform on the ADC. Following are the descriptions of each stage and the tasks that you must perform. The rollover procedure described here can be used for

© 1999-2017 Citrix Systems, Inc. All rights reserved.

both Key Signing Keys and Zone Signing Keys.

- **Stage 1: Initial.** The zone contains only those key sets with which the zone has currently been signed. The state of the zone in the initial stage is the state of the zone just before you begin the key rollover process.
  Example

  Consider the key, example.com.zsk1, with which the zone example.com is currently signed. The zone contains only those RRSIGs that were generated by the example.com.zsk1 key, which is due for expiry. The Key Signing Key is example.com.ksk1.

- **Stage 2: New DNSKEY.** The new key is published in the zone and the zone is signed with the new key. The zone contains the RRSIGs that are generated by the old and the new keys. The minimum duration for which the zone must contain both sets of RRSIGs is the time required for all the RRSIGs to expire.
  Example

  A new key example.com.zsk2 is added to the example.com zone. The zone is signed with example.com.zsk2. The example.com zone now contains the RRSIGs generated from both keys.

  NetScaler commands

  Perform the following tasks on the ADC:
  - Create a new DNS key by using the create dns key command.
    For more information about creating a DNS key, including an example, see [Creating DNS Keys for a Zone].
  - Publish the new key in the zone by using the add dns key command.
    For more information about publishing the key in the zone, including an example, see [Publishing a DNS Key in a Zone].
  - Sign the zone with the new key by using the sign dns zone command.
    For more information about signing a zone, including examples, see [Signing and Unsigning a DNS Zone].

- **Stage 3: DNSKEY Removal.** When the RRSIGs that were generated by the old DNS key expire, the old DNS key is removed from the zone.
  Example

  The old DNS key example.com.zsk1 is removed from the example.com zone.

  NetScaler commands

  On the ADC, you remove the old DNS key by using the rm dns key command.

  For more information about removing a key from a zone, including an example, see [Removing a DNS Key].

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Offloading DNSSEC Operations to the NetScaler ADC

Nov 21, 2014

For DNS zones for which your DNS servers are authoritative, you can offload DNSSEC operations to the NetScaler ADC. In a DNSSEC offloading deployment, a DNS server sends unsigned responses. The ADC signs the response on the fly before relaying it to the client. The ADC also caches the signed response. Apart from reducing the load on the DNS servers, offloading DNSSEC operations to the ADC gives you the following benefits:

- You can sign records that the DNS servers generate programmatically. Such records cannot be signed by routine zone signing operations performed on the DNS servers.
- You can serve signed responses to clients even if you have not implemented DNSSEC on your servers.

For setting up DNSSEC offloading, you must configure a DNS load balancing virtual server, configure services that represent the DNS servers, and then bind the services to the virtual server. For information about configuring a DNS load balancing virtual server, configuring services, and binding the services to the virtual server, see Configuring a DNS Zone.

You must create a zone entity on the ADC for each DNS zone whose DNSSEC operations you want to offload. For each DNS zone, you must enable the Proxy Mode and DNSSEC Offload parameters. You can optionally configure NSEC record generation for an offloaded zone. To create a DNS zone entity for DNSSEC offloading, follow the instructions in this topic.

To complete the configuration, you must generate DNS keys for the zone, add the keys to the zone, and then sign the zone with the keys. This process is the same as for normal DNSSEC. For information about creating keys, adding keys to a zone, and signing the zone, see Domain Name System Security Extensions.

After you configure DNS offloading, you must flush the DNS cache on the ADC. Flushing the DNS cache ensures that any unsigned records in the cache are removed and subsequently replaced by signed records. For information about flushing the DNS cache, see Flushing DNS Records.

Note: DNSSEC offloading is supported on all NetScaler MPX platforms, except the NetScaler MPX 9700/10500/12500/15500 FIPS platform. The feature is also supported on NetScaler virtual appliances hosted on NetScaler SDX platforms.

To enable DNSSEC offloading for a zone by using the command line interface

At the command line, type the following commands to enable DNSSEC offloading for a zone and verify the configuration:

- add dns zone <zoneName> -proxyMode YES -dnssecOffload ENABLED [-nsec (ENABLED | DISABLED)
- show dns zone

Example

```
> add dns zone example.com -proxyMode YES -dnssecOffload ENABLED nsec ENABLED
 Done
> show dns zone example.com
  Zone Name : example.com
  Proxy Mode : YES
  DNSSEC Offload: ENABLED NSEC: ENABLED
 Done
>
```

To enable DNSSEC offloading for a zone by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Traffic Management > DNS > Zones.
2. In the details pane, do one of the following:
   - To create a zone on the ADC, click Add.
   - To configure DNSSEC offloading for an existing zone, double-click the zone.
3. In the Create DNS Zone or Configure DNS Zone dialog box, select the Proxy Mode and DNSSEC Offload check boxes.
4. Optionally, if you want the ADC to generate NSEC records for the zone, select the NSEC check box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Firewall Load Balancing

Jun 29, 2012

Firewall load balancing distributes traffic across multiple firewalls, providing fault tolerance and increased throughput. Firewall load balancing protects your network by:

- Dividing the load between the firewalls, which eliminates a single point of failure and allows the network to scale.
- Increasing high availability.

Configuring a NetScaler appliance for firewall load balancing is similar to configuring load balancing, with the exception that the recommended service type is ANY, recommended monitor type is PING, and the load balancing virtual server mode is set to MAC.

You can set up firewall load balancing in a sandwich, an enterprise, or multiple-firewall environment configuration. The sandwich environment is used for load balancing traffic entering the network from outside and traffic leaving the network to the internet and involves configuring two NetScaler appliances, one on each side of a set of firewalls. You configure an enterprise environment for load balancing traffic leaving the network to the internet. The enterprise environment involves configuring a single NetScaler appliance between the internal network and the firewalls that provide access to the Internet. The multiple-firewall environment is used for load balance traffic coming from another firewall. Having firewall load balancing enabled on both the sides of NetScaler improves the traffic flow in both the egress and ingress direction and ensures faster processing of the traffic. The multiple-firewall environment involves configuring a NetScaler appliance sandwiched between two firewalls.

Important: If you configure static routes on the NetScaler for the destination IP address and enable L3 mode, the NetScaler uses its routing table to route the traffic instead of sending the traffic to the load balancing vserver.
Note: For FTP to work, an additional virtual server or service should be configured on the NetScaler with IP address and port as * and 21 respectively, and the service type specified as FTP. In this case, the NetScaler manages the FTP protocol by accepting the FTP control connection, modifying the payload, and managing the data connection, all through the same firewall.
Firewall Load Balancing supports only some of the load balancing methods supported on the NetScaler. Also, you can configure only a few types of persistence and monitors.

Firewall Load Balancing Methods

The following load balancing methods are supported for firewall load balancing.

- Least Connections
- Round Robin
- Least Packets
- Least Bandwidth
- Source IP Hash
- Destination IP Hash
- Source IP Destination IP Hash
- Source IP Source Port hash
- Least Response Time Method (LRTM)
- Custom Load

Firewall Persistence

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Only SOURCEIP, DESTIP, and SOURCEIPDESTIP based persistence are supported for firewall load balancing.

Firewall Server Monitoring

Only PING and transparent monitors are supported in firewall load balancing. You can bind a PING monitor (default) to the backend service that represents the firewall. If a firewall is configured not to respond to ping packets, you can configure transparent monitors to monitor hosts on the trusted side through individual firewalls.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Sandwich Environment

Apr 08, 2013

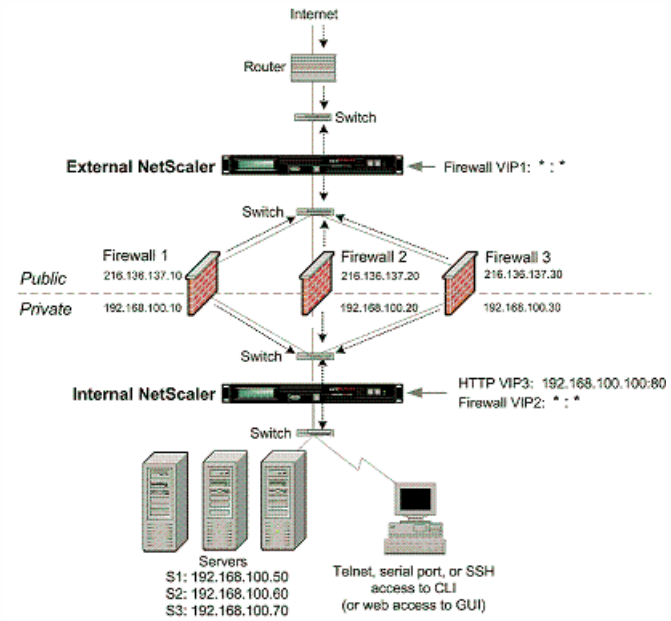
A NetScaler deployment in a sandwich mode is capable of load balancing network traffic through firewalls in both directions: ingress (traffic entering the network from the outside, such as the internet) and egress (traffic leaving the network to the internet).

In this setup, a NetScaler is located on each side of a set of firewalls. The NetScaler placed between the firewalls and the Internet, called the external NetScaler that handles ingress traffic selects the best firewall, based on the configured method. The NetScaler between the firewalls and the private network, called the internal NetScaler tracks the firewall from which the initial packet for a session is received. It then makes sure that all subsequent packets for that session are sent to the same firewall.

The internal NetScaler can be configured as a regular traffic manager to load balance traffic across the private network servers. This configuration also allows traffic originating from the private network (egress) to be load balanced across the firewalls.

The following diagram shows the sandwich firewall load balancing environment.

Figure 1. Firewall Load Balancing (Sandwich)



The service type ANY configures the NetScaler to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and virtual server with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the External NetScaler in a Sandwich Environment

Updated: 2015-05-22

Perform the following tasks to configure the external NetScaler in a sandwich environment

- Enable the load balancing feature.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.
- Configure a wildcard virtual server for traffic coming from the Internet.
- Configure the virtual server in MAC rewrite mode.
- Bind services to the wildcard virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

- enable ns feature LB
- show ns feature

Example

```
> enable ns feature LoadBalancing
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging            WL               OFF
 2)   Surge Protection       SP               ON
 3)   Load Balancing         LB               ON
 .
 .
 .
 24)  NetScaler Push         push             OFF
 Done
```

To enable load balancing by using the configuration utility

Navigate to System > Settings and, in **Configure Basic Features**, select **Load Balancing**.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the command line interface

At the command prompt, type:

add service <name> <serverName> ANY *

Example

```
add service Service-HTTP-1 10.102.29.5 ANY *
```

To configure a wildcard service for each firewall by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services and add a service. Specify **ANY** in the **Protocol** field and ***** in the Port field.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-transparent (YES | NO)]
- bind lb monitor <monitorName> <serviceName>

Example

add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
To bind a PING monitor, type the following command:
bind monitor PING fw-svc1
To create and bind a transparent monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and then create and bind a transparent monitor.

Configure a wildcard virtual server for traffic coming from the Internet

To configure a wildcard virtual server for traffic coming from the Internet by using the command line interface

At the command prompt, type:

add lb vserver <name> ANY * *
Example

add lb vserver Vserver-LB-1 ANY * *
To configure a wildcard virtual server for traffic coming from the Internet by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers and create a wildcard virtual server. Specify **ANY** in the **Protocol** field and ***** in the Port field.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the command line interface

At the command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

set lb vserver <name>@ -m <RedirectionMode>

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1).
2. Edit the **Basic Settings** section and. click **more**.
3. From the **Redirection Mode** drop-down list, select **MAC Based**.

Bind services to the wildcard virtual server

To bind a service to the wildcard virtual server by using the command line interface

At the command prompt, type:

bind lb vserver <name> <serviceName>

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind a service to the wildcard virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the virtual server for which you want to bind the service.
2. Click in the **Services** section and select a service to bind.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save ns config
- show vserver

Example

```
save config
sh lb vserver FWLBVIP1
FWLBVIP1 (*:*) - ANY    Type: ADDRESS
      State: UP
      Last state change was at Mon Jun 14 06:40:14 2010
      Time since last state change: 0 days, 00:00:11.240
      Effective State: UP  ARP:DISABLED
      Client Idle Timeout: 120 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      No. of Bound Services :  2 (Total)       2 (Active)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Configured Method: SRCIPDESTIPHASH
    Mode: MAC
    Persistence: NONE
    Connection Failover: DISABLED

1) fw_svc_1 (10.102.29.251: *) - ANY State: UP  Weight: 1
2) fw_svc_2 (10.102.29.18: *) - ANY State: UP   Weight: 1
 Done
show service fw-svc1
    fw-svc1 (10.102.29.251:*) - ANY
    State: DOWN
    Last state change was at Thu Jul  8 10:04:50 2010
    Time since last state change: 0 days, 00:00:38.120
    Server Name: 10.102.29.251
    Server ID : 0   Monitor Threshold : 0
    Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
    TCP Buffering(TCPB): YES
    HTTP Compression(CMP): NO
    Idle timeout: Client: 120 sec   Server: 120 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: OFF
    Down state flush: ENABLED

1)    Monitor Name: monitor-HTTP-1
        State: DOWN     Weight: 1
        Probes: 5       Failed [Total: 5 Current: 5]
        Last response: Failure - Time out during TCP connection establishment stage
        Response Time: 2000.0 millisec
2)    Monitor Name: ping
        State: UP       Weight: 1
        Probes: 3       Failed [Total: 0 Current: 0]
        Last response: Success - ICMP echo reply received.
        Response Time: 1.415 millisec
 Done
```

Configuring the Internal NetScaler ADC in a Sandwich Environment

Updated: 2015-06-04

Perform the following tasks to configure the internal NetScaler in a sandwich environment

For traffic from the server (egress)

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Configure a monitor for each wildcard service.
- Configure a wildcard virtual server to load balance the traffic sent to the firewalls.
- Configure the virtual server in MAC rewrite mode.
- Bind firewall services to the wildcard virtual server.

For traffic across private network servers

- Configure a service for each virtual server.
- Configure a monitor for each service.
- Configure an HTTP virtual server to balance traffic sent to the servers.
- Bind HTTP services to the HTTP virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

- enable ns feature LB
- show ns feature

Example

```
> enable ns feature LoadBalancing
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging            WL               OFF
 2)   Surge Protection       SP               ON
 3)   Load Balancing         LB               ON
 .
 .
 .
 24)  NetScaler Push         push             OFF
 Done
```

To enable load balancing by using the configuration utility

Navigate to System > Settings and, in Configure Basic Features, select Load Balancing.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the command line interface

At the command prompt, type:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

add service <name> <serverName> ANY *

Example

add service Service-HTTP-1 10.102.29.5 ANY *

To configure a wildcard service for each firewall by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services and add a service. Specify **ANY** in the **Protocol** field and ***** in the Port field.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-transparent (YES | NO)]
- bind lb monitor <monitorName> <serviceName>

Example

add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1

To create and bind a transparent monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors and create a monitor.
2. In the **Create Monitor** dialog box, enter the required parameters and select **Transparent**.

Configure a wildcard virtual server to load balance the traffic sent to the firewalls

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the command line interface

At the command prompt, type:

add lb vserver <name> ANY * *

Example

add lb vserver Vserver-LB-1 ANY * *

To configure a wildcard virtual server for traffic coming from the Internet by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers and create a wildcard virtual server. Specify **ANY** in the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Protocol field and ***** in the Port field.

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
   - Name—name
4. In Protocol, select ANY, and in IP Address and Port, select *.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the command line interface

At the command prompt, type:

```
set lb vserver <name>@ -m <RedirectionMode>
```

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1).
2. Edit the **Basic Settings** section and. click **more**.
3. From the **Redirection Mode** drop-down list, select **MAC Based**.

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1), and then click Open.
3. On the Advanced tab, under Redirection Mode, click MAC-Based.
4. Click OK.

Bind firewall services to the wildcard virtual server

To bind firewall services to the wildcard virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
```

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind firewall services to the wildcard virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and select a virtual server.
2. Click in the Service section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Configure a service for each virtual server

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure a service for each virtual server by using the command line interface

At the command prompt, type:

add service <name> <serverName> HTTP <port>
Example

add service Service-HTTP-1 10.102.29.5 HTTP 80
To configure a service for each virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services and configure a service for each virtual server. Specify **HTTP** in the **Protocol** field and select **HTTP** under **Available Monitors**.

1.

To configure a service for each virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
   - Service Name—name
   - Server—serverName
   - Port—port
4. In Protocol, specify HTTP. Under Available Monitors, select HTTP.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configure a monitor for each service

To bind a monitor to a service by using the command line interface

At the command prompt, type:

bind lb monitor <monitorName> <ServiceName>
Example

bind mon monitor-HTTP-1 Service-HTTP-1
To bind a monitor to a service by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, double-click a service and add a monitor.
To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and add a monitor.

Configure an HTTP virtual server to balance traffic sent to the servers

To configure an HTTP virtual server to balance traffic sent to the servers by using the command line interface

At the command prompt, type:

add lb vserver <name> HTTP <ip> <port>
Example

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Services and configure an HTTP virtual server. Specify **HTTP** in the **Protocol** field.

1.

To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
   - Name—name
   - IP Address—IPAddress
     Note: If the virtual server uses IPv6, select the IPv6 check box and enter the address in IPv6 format (for example, **1000:0000:0000:0000:0005:0600:700a:888b**).
   - Port—port
4. Under Protocol, select HTTP.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save ns config
- show vserver

Example

```
save config
show lb vserver FWLBVIP2
    FWLBVIP2 (*:*) - ANY    Type: ADDRESS
    State: UP
    Last state change was at Mon Jun 14 07:22:54 2010
    Time since last state change: 0 days, 00:00:32.760
    Effective State: UP
    Client Idle Timeout: 120 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    No. of Bound Services :  2 (Total)      2 (Active)
    Configured Method: LEASTCONNECTION
    Current Method: Round Robin, Reason: A new service is bound
    Mode: MAC
    Persistence: NONE
    Connection Failover: DISABLED
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
      fw-int-svc1 (10.102.29.5:*) - ANY
      State: DOWN
      Last state change was at Thu Jul  8 14:44:51 2010
      Time since last state change: 0 days, 00:01:50.240
      Server Name: 10.102.29.5
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): NO
      HTTP Compression(CMP): NO
      Idle timeout: Client: 120 sec   Server: 120 sec
      Client IP: DISABLED
      Cacheable: NO
      SC: OFF
      SP: OFF
      Down state flush: ENABLED

1)    Monitor Name: monitor-HTTP-1
            State: DOWN     Weight: 1
            Probes: 9       Failed [Total: 9 Current: 9]
            Last response: Failure - Time out during TCP connection establishment stage
            Response Time: 2000.0 millisec
2)    Monitor Name: ping
            State: UP       Weight: 1
            Probes: 3       Failed [Total: 0 Current: 0]
            Last response: Success - ICMP echo reply received.
            Response Time: 1.275 millisec
 Done
```

To save and verify the configuration by using the configuration utility

1. In the details pane, click Save.
2. In the Save Config dialog box, click Yes.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the Details pane are correct.
5. Navigate to Traffic Management > Load Balancing > Services.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the Details pane are correct.

Monitoring a Firewall Load Balancing Setup in a Sandwich Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing the Statistics of a Virtual Server

Updated: 2013-09-05

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the command prompt, type:

stat lb vserver [-detail] [<name>]

Example

```
>stat lb vserver -detail
Virtual Server(s) Summary
                 vsvrIP  port    Protocol      State    Req/s   Hits/s
One                   *    80       HTTP         UP      5/s      0/s
Two                   *     0        TCP       DOWN      0/s      0/s
Three                 *  2598        TCP       DOWN      0/s      0/s
dnsVirtualNS    10.102.29.90    53        DNS       DOWN      0/s      0/s
BRVSERV          10.10.1.1    80       HTTP       DOWN      0/s      0/s
LBVIP         10.102.29.66    80       HTTP         UP      0/s      0/s
 Done
```

To display virtual server statistics by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers > Statistics.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

Viewing the Statistics of a Service

Updated: 2013-08-28

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the command line interface

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

stat service <name>

Example

stat service Service-HTTP-1

To view the statistics of a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

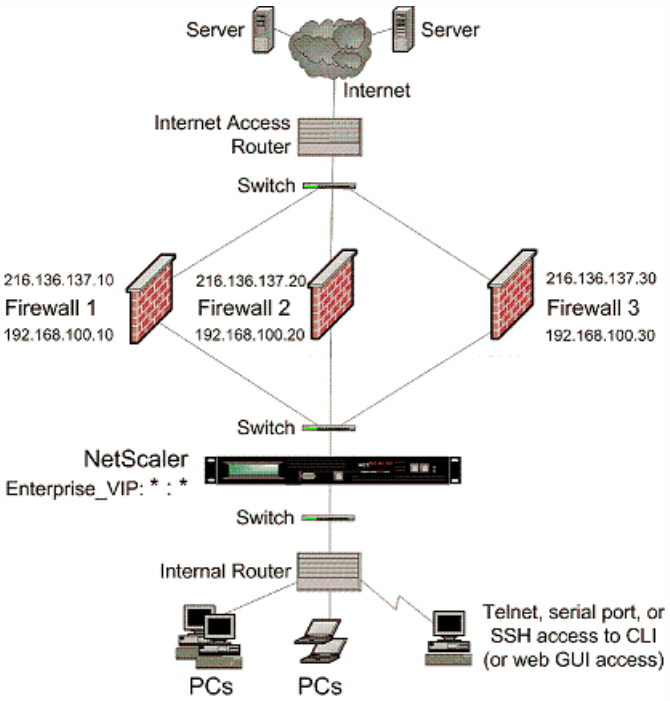
Enterprise Environment

Mar 22, 2012

In the enterprise setup, the NetScaler is placed between the firewalls connecting to the public Internet and the internal private network and handles egress traffic. The NetScaler selects the best firewall based on the configured load balancing policy.

The following diagram shows the enterprise firewall load balancing environment

Figure 1. Firewall Load Balancing (Enterprise)



The service type ANY configures the NetScaler to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and vserver with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the NetScaler in an Enterprise Environment

Updated: 2013-11-08

Perform the following tasks to configure a NetScaler in an enterprise environment.

For traffic from the server (egress)

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.
- Configure a wildcard virtual server to load balance the traffic sent to the firewalls.
- Configure the virtual server in MAC rewrite mode.
- Bind firewall services to the wildcard virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For traffic across private network servers

- Configure a service for each virtual server.
- Configure a monitor for each service.
- Configure an HTTP virtual server to balance traffic sent to the servers.
- Bind HTTP services to the HTTP virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

- enable ns feature LB
- show ns feature

Example

```
> enable ns feature LoadBalancing
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging            WL               OFF
 2)   Surge Protection       SP               ON
 3)   Load Balancing         LB               ON
 .
 .
 .
 24)  NetScaler Push         push             OFF
 Done
```

To enable load balancing by using the configuration utility

Navigate to System > Settings and, in Configure Basic Features, select Load Balancing.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the command line interface

At the command prompt, type:

add service <name> <serverName> ANY *

Example

add service Service-HTTP-1 10.102.29.5 ANY *

To configure a wildcard service for each firewall by using the configuration utility

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
   - Service Name—name
   - Server—serverName
4. In Protocol, select ANY, and in Port, select *.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-transparent (YES | NO)]
- bind lb monitor <monitorName> <serviceName>

Example

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
```

To create and bind a transparent monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. In the details pane, click Add.
3. In the Create Monitor dialog box, specify values as shown:

   - Name*
   - Type*—type
   - Destination IP
   - Transparent

   * A required parameter

4. Click Create, and then click Close. In the Monitors pane, select the monitor that you just configured and verify that the settings displayed at the bottom of the screen are correct.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configure a wildcard virtual server to load balance the traffic sent to the firewalls

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the command line interface

At the command prompt, type:

add lb vserver <name> ANY * *

Example

add lb vserver Vserver-LB-1 ANY * *

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
   - Name—name
4. In Protocol, select ANY, and in IP Address and Port, select *.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the command line interface

At the command prompt, type:

set lb vserver <name>@ -m <RedirectionMode>

Example

set lb vserver Vserver-LB-1 -m MAC

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1), and then click Open.
3. On the Advanced tab, under Redirection Mode, click MAC-Based.
4. Click OK.

Bind firewall services to the wildcard virtual server

To bind firewall services to the wildcard virtual server by using the command line interface

At the command prompt, type:

bind lb vserver <name> <serviceName>

Example

bind lb vserver Vserver-LB-1 Service-HTTP-1

To bind firewall services to the wildcard virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and select a virtual server.
2. Click in the Service section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configure a service for each virtual server

To configure a service for each virtual server by using the command line interface

At the command prompt, type:

add service <name> <serverName> HTTP <port>
Example

add service Service-HTTP-1 10.102.29.5 HTTP 80
To configure a service for each virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
   - Service Name—name
   - Server—serverName
   - Port—port
4. In Protocol, specify HTTP. Under Available Monitors, select HTTP.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configure a monitor for each service

To bind a monitor to a service by using the command line interface

At the command prompt, type:

bind lb monitor <monitorName> <ServiceName>
Example

bind mon monitor-HTTP-1 Service-HTTP-1
To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and add a monitor.

Configure an HTTP virtual server to balance traffic sent to the servers

To configure an HTTP virtual server to balance traffic sent to the servers by using the command line interface

At the command prompt, type:

add lb vserver <name> HTTP <ip> <port>
Example

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
   - Name—name
   - IP Address—IPAddress

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: If the virtual server uses IPv6, select the IPv6 check box and enter the address in IPv6 format (for example, **1000:0000:0000:0000:0005:0600:700a:888b**).

- Port—port

4. Under Protocol, select HTTP.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Bind HTTP services to the HTTP virtual server

To bind HTTP services to the wildcard virtual server by using the command line interface

At the command prompt, type:

bind lb vserver <name> <serviceName>

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind HTTP services to the wildcard virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and select a virtual server.
2. Click in the Service section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save ns config
- show vserver

Example

```
save config
show lb vserver FWLBVIP2
    FWLBVIP2 (*:*) - ANY    Type: ADDRESS
    State: UP
    Last state change was at Mon Jun 14 07:22:54 2010
    Time since last state change: 0 days, 00:00:32.760
    Effective State: UP
    Client Idle Timeout: 120 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    No. of Bound Services :  2 (Total)      2 (Active)
    Configured Method: LEASTCONNECTION
    Current Method: Round Robin, Reason: A new service is bound
    Mode: MAC
    Persistence: NONE
    Connection Failover: DISABLED
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
      fw-int-svc1 (10.102.29.5:*) - ANY
      State: DOWN
      Last state change was at Thu Jul  8 14:44:51 2010
      Time since last state change: 0 days, 00:01:50.240
      Server Name: 10.102.29.5
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): NO
      HTTP Compression(CMP): NO
      Idle timeout: Client: 120 sec   Server: 120 sec
      Client IP: DISABLED
      Cacheable: NO
      SC: OFF
      SP: OFF
      Down state flush: ENABLED

1)    Monitor Name: monitor-HTTP-1
          State: DOWN     Weight: 1
          Probes: 9       Failed [Total: 9 Current: 9]
          Last response: Failure - Time out during TCP connection establishment stage
          Response Time: 2000.0 millisec
2)    Monitor Name: ping
          State: UP       Weight: 1
          Probes: 3       Failed [Total: 0 Current: 0]
          Last response: Success - ICMP echo reply received.
          Response Time: 1.275 millisec
 Done
```

To save and verify the configuration by using the configuration utility

1. In the details pane, click Save.
2. In the Save Config dialog box, click Yes.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the Details pane are correct.
5. Navigate to Traffic Management > Load Balancing > Services.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the Details pane are correct.

Monitoring a Firewall Load Balancing Setup in an Enterprise Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

possible problems.

Viewing the Statistics of a Virtual Server

Updated: 2013-09-05

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the command prompt, type:

stat lb vserver [-detail] [<name>]

Example

```
>stat lb vserver -detail
Virtual Server(s) Summary
                vsvrIP  port    Protocol      State    Req/s   Hits/s
One                  *    80       HTTP         UP      5/s      0/s
Two                  *     0        TCP       DOWN      0/s      0/s
Three                *  2598        TCP       DOWN      0/s      0/s
dnsVirtualNS   10.102.29.90    53        DNS       DOWN      0/s      0/s
BRVSERV         10.10.1.1    80       HTTP       DOWN      0/s      0/s
LBVIP        10.102.29.66    80       HTTP         UP      0/s      0/s
 Done
```

To display virtual server statistics by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers > Statistics.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

Viewing the Statistics of a Service

Updated: 2013-08-28

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To view the statistics of a service by using the command line interface

At the command prompt, type:

stat service <name>

Example

stat service Service-HTTP-1

To view the statistics of a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Multiple-Firewall Environment

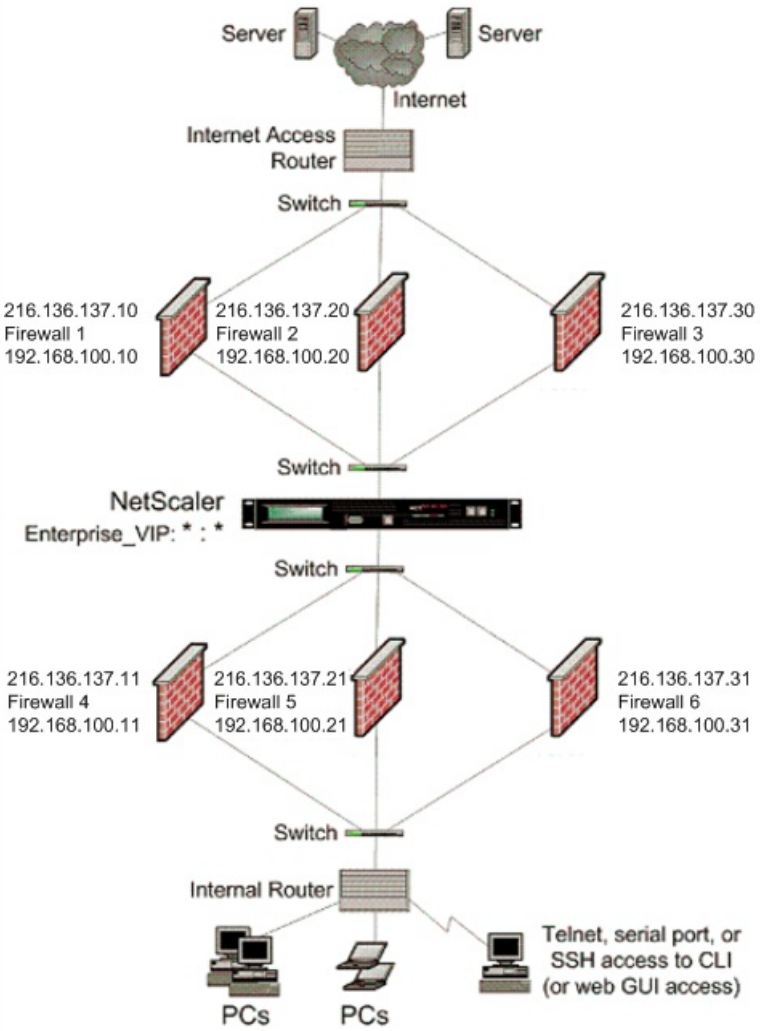
Sep 06, 2012

Note: This feature is available in NetScaler release 9.3.e and 10.

In a multiple-firewall environment, the NetScaler appliance is placed between two sets of firewalls, the external set connecting to the public Internet, and the internal set connecting to the internal private network. The external set typically handles the egress traffic. These firewalls mainly implement access control lists to allow or deny access to external resources. The internal set typically handles the ingress traffic. These firewalls implement security to safeguard the intranet from malicious attacks apart from load-balancing the ingress traffic. The multiple-firewall environment allows you to load-balance traffic coming from another firewall. By default, the traffic coming from a firewall is not load balanced on the other firewall across a NetScaler. Having firewall load balancing enabled on both the sides of NetScaler improves the traffic flow in both the egress and ingress direction and ensures faster processing of the traffic.

Figure 1 shows a multiple-firewall load balancing environment

Figure 1. Firewall Load Balancing (multiple-firewall)



With a configuration like the one shown in Figure 1, you can configure the NetScaler to load balance the traffic through the an internal firewall even if it is load balanced by an external firewall. For example, with this feature configured, the traffic coming from the external firewalls (firewalls 1, 2, and 3) is load balanced on the internal firewalls (firewalls 4, 5, and 6) and vice versa.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Firewall load balancing is supported only for MAC mode LB virtual server.

The service type ANY configures the NetScaler to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and virtual server with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the NetScaler in a Multiple-Firewall Environment

Updated: 2015-05-18

To configure a NetScaler appliance in a multiple-firewall environment, you have to enable the load balancing feature, configure a virtual server to load balance the egress traffic across the external firewalls, configure a virtual server to load balance the ingress traffic across the internal firewalls, and enable firewall load balancing on the NetScaler. To configure a virtual server to load balance traffic across a firewall in the multiple-firewall environment, you need to:

1. Configure a wildcard service for each firewall
2. Configure a monitor for each wildcard service
3. Configure a wildcard virtual server to load balance the traffic sent to the firewalls
4. Configure the virtual server in MAC rewrite mode
5. Bind firewall services to the wildcard virtual server

Enabling the load balancing feature

To configure and implement load balancing entities such as services and virtual servers, you need to enable the load balancing feature on the NetScaler device.

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

- enable ns feature <featureName>
- show ns feature

Example

```
enable ns feature LoadBalancing
Done
show ns feature
Feature Acronym Status
------- ------- ------
1) Web Logging WL OFF
2) Surge Protection SP ON
3) Load Balancing LB ON
.
.
.
24) NetScaler Push push OFF
Done
```

To enable load balancing by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the Settings pane, under Modes and Features, click Change basic features.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Configure Basic Features dialog box, select the Load Balancing check box, and then click Ok.

Configuring a wildcard service for each firewall

To accept traffic from all the protocols, you need to configure wildcard service for each firewall by specifying support for all the protocols and ports.
To configure a wildcard service for each firewall by using the command line interface

At the command prompt, type the following command to configure support for all the protocols and ports:

add service <name>@ <serverName> <serviceType> <port_number>
Example

add service fw-svc1 10.102.29.5 ANY *
To configure a wildcard service for each firewall by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Services dialog box, specify values for the following parameters as shown:

   - Service Name—name
   - Server—serverName

   * A required parameter

4. In Protocol, select Any and in Port, select *.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configuring a monitor for each service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-transparent (YES | NO)]
- bind lb monitor <monitorName> <serviceName>

Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
```

To create and bind a transparent monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. In the details pane, click Add.
3. In the Create Monitor dialog box, specify values for the following parameters as shown:

   - Name*
   - Type*—type
   - Destination IP
   - Transparent

   * A required parameter

4. Click Create, and then click Close. In the Monitors pane, select the monitor that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configuring a virtual server to load balance the traffic sent to the firewalls

To load balance any kind of traffic, you need to configure a wildcard virtual server specifying the protocol and port as any value.

To configure a virtual server to load balance the traffic sent to the firewalls by using the command line interface

At the command prompt, type the following command:

```
add lb vserver <name>@ <serviceType> <IPAddress> <port_number>
```

Example

```
add lb vserver Vserver-LB-1 ANY * *
```

To configure a virtual server to load balance the traffic sent to the firewalls by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In Protocol, select Any, and in IP Address and Port, select *.
4. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Configuring the virtual server to MAC rewrite mode

To configure the virtual server to use MAC address for forwarding the incoming traffic, you need to enable the MAC rewrite mode.

To configure the virtual server in MAC rewrite mode by using the command line interface

At the command prompt, type the following command:

```
set lb vserver <name>@ -m <RedirectionMode>
```

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB1), and then click Open.
3. On the Advanced tab, under the Redirection Mode mode, click Open.
4. Click Ok.

Binding firewall services to the virtual server

To access a service on NetScaler, you need to bind it to a wildcard virtual server.
To bind firewall services to the virtual server by using the command line interface

At the command prompt, type the following command:

bind lb vserver <name>@ <serviceName>
Example

bind lb vserver Vserver-LB-1 Service-HTTP-1
To bind firewall services to the virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Services tab, select the Active check box next to the service that you want to bind to the virtual server(for example, Service-HTTP-1).
4. Click Ok.

Configuring the multiple-firewall load balancing on the NetScaler Appliance

To load balance traffic on both the sides of a NetScaler using firewall load balancing, you need to enable mulitpl-firewall load balancing by using the vServerSpecificMac parameter.
To configure multiple-firewall load balancing by using the command line interface

At the command prompt, type the following command:

set lb parameter -vServerSpecificMac <status>
Example

set lb parameter -vServerSpecificMac ENABLED
To configure multiple-firewall load balancing by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Configure Load Balancing parameters).
3. In the Set Load Balancing Parameters dialog box, select the Virtual Server Specific MAC check box.
4. Click Ok.

Saving and Verifying the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the command line interface

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save ns config
- show vserver

Example

```
save config
show lb vserver FWLBVIP2
    FWLBVIP2 (*:*) - ANY    Type: ADDRESS
    State: UP
    Last state change was at Mon Jun 14 07:22:54 2010
    Time since last state change: 0 days, 00:00:32.760
    Effective State: UP
    Client Idle Timeout: 120 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    No. of Bound Services :  2 (Total)      2 (Active)
    Configured Method: LEASTCONNECTION
    Current Method: Round Robin, Reason: A new service is bound
    Mode: MAC
    Persistence: NONE
    Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
    fw-int-svc1 (10.102.29.5:*) - ANY
    State: DOWN
    Last state change was at Thu Jul  8 14:44:51 2010
    Time since last state change: 0 days, 00:01:50.240
    Server Name: 10.102.29.5
    Server ID : 0   Monitor Threshold : 0
    Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
    TCP Buffering(TCPB): NO
    HTTP Compression(CMP): NO
    Idle timeout: Client: 120 sec   Server: 120 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: OFF
    Down state flush: ENABLED

1)    Monitor Name: monitor-HTTP-1
        State: DOWN     Weight: 1
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        Probes: 9      Failed [Total: 9 Current: 9]
        Last response: Failure - Time out during TCP connection establishment stage
        Response Time: 2000.0 millisec
2)      Monitor Name: ping
        State: UP      Weight: 1
        Probes: 3      Failed [Total: 0 Current: 0]
        Last response: Success - ICMP echo reply received.
        Response Time: 1.275 millisec
 Done
```

To save and verify the configuration by using the configuration utility

1. In the details pane, click Save.
2. In the Save Config dialog box, click Yes.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the Details pane are correct.
5. Navigate to Traffic Management > Load Balancing > Services.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the Details pane are correct.

Monitoring a Firewall Load Balancing Setup in a Multiple-Firewall Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

Updated: 2013-09-05

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the command prompt, type:

```
stat lb vserver [-detail] [<name>]
```

Example

```
>stat lb vserver -detail
Virtual Server(s) Summary
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
                vsvrIP  port    Protocol      State    Req/s   Hits/s
One                  *   80       HTTP         UP      5/s     0/s
Two                  *    0       TCP        DOWN     0/s     0/s
Three                *  2598       TCP        DOWN     0/s     0/s
dnsVirtualNS    10.102.29.90   53       DNS        DOWN     0/s     0/s
BRVSERV         10.10.1.1   80       HTTP       DOWN     0/s     0/s
LBVIP        10.102.29.66   80       HTTP         UP     0/s     0/s
 Done
```

To display virtual server statistics by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers > Statistics.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

Viewing the Statistics of a Service

Updated: 2013-08-28

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the command line interface

At the command prompt, type:

stat service <name>

Example

stat service Service-HTTP-1

To view the statistics of a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Global Server Load Balancing

May 19, 2016

NetScaler appliances configured for global server load balancing (GSLB) provide for disaster recovery and ensure continuous availability of applications by protecting against points of failure in a wide area network (WAN). GSLB can balance the load across data centers by directing client requests to the closest or best performing data center, or to surviving data centers in case of an outage.

Following are some typical GSLB configurations:

- **Active-active data center setup**. Consists of multiple active data centers. Client requests are load balanced across active data centers.
- **Active-standby data center setup**. Consists of an active and a standby data center. When a failover occurs as a result of a disaster event, the standby data center becomes operational.
- **Proximity setup**. Directs client requests to the data center that is closest in geographical distance or network distance.

In a typical configuration, a local DNS server sends client requests to a GSLB virtual server, to which are bound GSLB services. A GSLB service identifies a load balancing or content switching virtual server, which can be at the local site or a remote site. If the GSLB virtual server selects a load balancing or content switching virtual server at a remote site, it sends the virtual server's IP address to the DNS server, which sends it to the client. The client then resends the request to the new virtual server at the new IP.

The GSLB entities that you must configure are the GSLB sites, the GSLB services, the GSLB virtual servers, load balancing or content switching virtual servers, and authoritative DNS (ADNS) services. You must also configure MEP. You can also configure DNS views to expose different parts of your network to clients accessing the network from different locations.

Note: To take full advantage of the NetScaler GSLB features, you should use NetScaler appliances for load balancing or content switching at each data center, so that your GSLB configuration can use the proprietary Metric Exchange Protocol (MEP) to exchange site metrics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How GSLB Works

May 19, 2016

With ordinary DNS, when a client sends a domain name system (DNS) request, it receives a list of IP addresses of the domain or service. Generally, the client chooses the first IP address in the list and initiates a connection with that server. The DNS server uses a technique called DNS round robin to rotate through the IPs on the list, sending the first IP address to the end of the list and promoting the others after it responds to each DNS request. This technique ensures equal distribution of the load, but it does not support disaster recovery, load balancing based on load or proximity of servers, or persistence.

When you configure GSLB on NetScaler appliances and enable Metric Exchange Protocol (MEP), the appliances use the DNS infrastructure to connect the client to the data center that best meets the criteria that you set. The criteria can designate the least loaded data center, the closest data center, the data center that responds most quickly to requests from the client's location, a combination of those metrics, and SNMP metrics. An appliance keeps track of the location, performance, load, and availability of each data center and uses these factors to select the data center to which to send a client request.

A GSLB configuration consists of a group of GSLB entities on each appliance in the configuration. These entities include GSLB sites, GSLB services, GSLB virtual servers, load balancing and/or content switching servers, and ADNS services.

This document includes the following information:

- GSLB Sites
- GSLB Services
- GSLB Virtual Servers
- Load Balancing or Content Switching Virtual Servers
- ADNS Services
- DNS VIPs

GSLB Sites

A typical GSLB setup consists of data centers, each of which has various network appliances that may or may not be NetScaler appliances. The data centers are called GSLB sites. Each GSLB site is managed by a NetScaler appliance that is local to that site. Each of these appliances treats its own site as the local site and all other sites, managed by other appliances, as remote sites.

If the appliance that manages a site is the only NetScaler appliance in that data center, the GSLB site hosted on that appliance acts as a bookkeeping placeholder for auditing purposes, because no metrics can be collected. Typically, this happens when the appliance is used only for GSLB, and other products in the data center are used for load balancing or content switching.

GSLB Services

A GSLB service is usually a representation of a load balancing or content switching virtual server, although it can represent any type of virtual server. The GSLB service identifies the virtual server's IP address, port number, and service type. GSLB services are bound to GSLB virtual servers on the NetScaler appliances managing the GSLB sites. A GSLB service bound to a GSLB virtual server in the same data center is local to the GSLB virtual server. A GSLB service bound to a GSLB virtual server in

© 1999-2017 Citrix Systems, Inc. All rights reserved.

a different data center is remote from that GSLB virtual server.

GSLB Virtual Servers

A GSLB virtual server has one or more GSLB services bound to it, and load balances traffic among those services. It evaluates the configured GSLB methods (algorithms) to select the appropriate service to which to send a client request. Because the GSLB services can represent either local or remote servers, selecting the optimal GSLB service for a request has the effect of selecting the data center that should serve the client request.

The domain for which global server load balancing is configured must be bound to the GSLB virtual server, because one or more services bound to the virtual server will serve requests made for that domain.

Unlike other virtual servers configured on a NetScaler appliance, a GSLB virtual server does not have its own virtual IP address (VIP).

Load Balancing or Content Switching Virtual Servers

Updated: 2013-09-13

A load balancing or content switching virtual server represents one or many physical servers on the local network. Clients send their requests to the load balancing or content switching virtual server's virtual IP (VIP) address, and the virtual server balances the load across the physical servers. After a GSLB virtual server selects a GSLB service representing either a local or a remote load balancing or content switching virtual server, the client sends the request to that virtual server's VIP address.

For more information about load balancing or content switching virtual servers and services, see Load Balancing, or Content Switching.

ADNS Services

An ADNS service is a special kind of service that responds only to DNS requests for domains for which the NetScaler appliance is authoritative. When an ADNS service is configured, the appliance owns that IP address and advertises it. Upon reception of a DNS request by an ADNS service, the appliance checks for a GSLB virtual server bound to that domain. If a GSLB virtual server is bound to the domain, it is queried for the best IP address to which to send the DNS response.

DNS VIPs

A DNS virtual IP is a virtual IP (VIP) address that represents a load balancing DNS virtual server on the NetScaler appliance. DNS requests for domains for which the NetScaler appliance is authoritative can be sent to a DNS VIP.

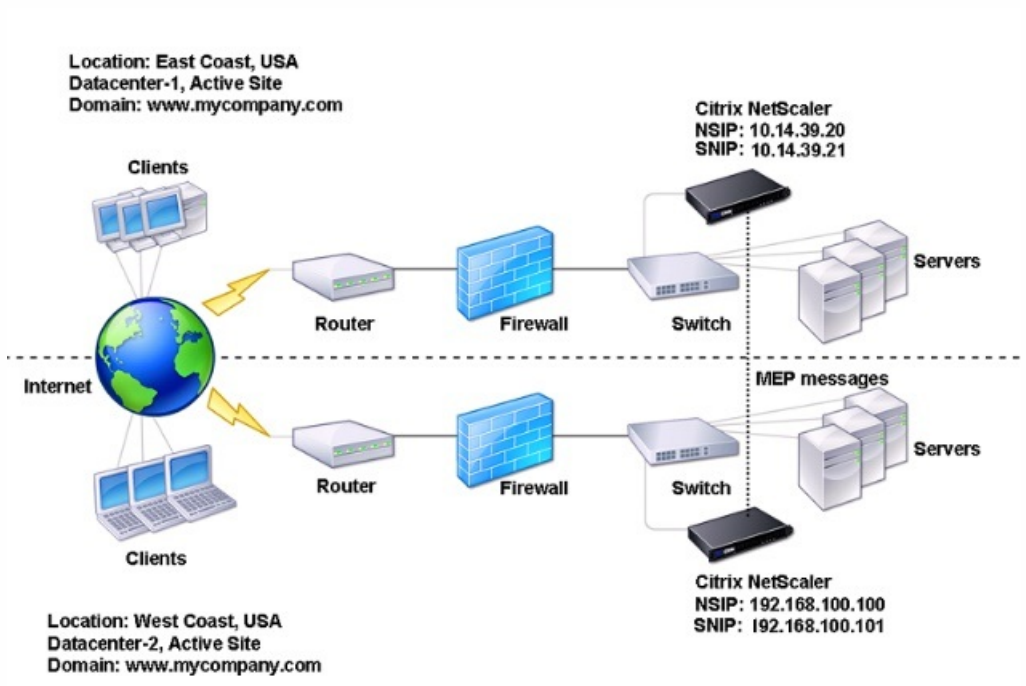
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Global Server Load Balancing (GSLB)

Oct 22, 2017

Global server load balancing is used to manage traffic flow to a web site hosted on two separate server farms that ideally are in different geographic locations. For example, consider a Web site, www.mycompany.com, which is hosted on two geographically separated server farms or data centers. Both server farms use NetScaler appliances. The NetScaler appliances in these server farms are set up in one-arm mode and function as authoritative DNS servers for the www.mycompany.com domain. The following figure illustrates this configuration.

Figure 1. Basic GSLB Topology



To configure such a GSLB setup, you must first configure a standard load balancing setup for each server farm or data center. This enables you to balance load across the different servers in each server farm. Then, configure both NetScaler appliances as authoritative DNS (ADNS) servers. Next, create a GSLB site for each server farm, configure GSLB virtual servers for each site, create GLSB services, and bind the GSLB services to the GSLB virtual servers. Finally, bind the domain to the GSLB virtual servers. The GSLB configurations on the two appliances at the two different sites are identical, although the load-balancing configurations for each site is specific to that site.

Note: To configure a GSLB site in a NetScaler cluster setup, see Setting Up GSLB in a Cluster.

Configuring a Standard Load Balancing Setup

Updated: 2013-08-30

A load balancing virtual server balances the load across different physical servers in the data center. These servers are represented as services on the NetScaler appliance, and the services are bound to the load balancing virtual server.

For details on configuring a basic load balancing setup, see Load Balancing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring an Authoritative DNS Service

Aug 24, 2016

When you configure the NetScaler appliance as an authoritative DNS server, it accepts DNS requests from the client and responds with the IP address of the data center to which the client should send requests.

Note: For the NetScaler to be authoritative, you must also create SOA and NS records. For more information about SOA and NS records, see "Domain Name System".

To create an ADNS service by using the command line interface

At the command prompt, type the following commands to create an ADNS service and verify the configuration:

- add service <name> <IP>@ ADNS <port>
- show service <name>

Example

```
add service Service-ADNS-1 10.14.39.21 ADNS 53
show service Service-ADNS-1
```

To modify an ADNS service by using the command line interface

At the command prompt, type the following command:

```
set service <name> <IPAddress> ADNS <port>
```

Example

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
```

To remove an ADNS service by using the command line interface

At the command prompt, type the following command:

```
rm service <name>
```

Example

```
rm service Service-ADNS-1
```

To configure an ADNS service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Add a new ADNS service, or select an existing service and edit its settings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Basic GSLB Site

Nov 24, 2014

A GSLB site is a representation of a data center in your network and is a logical grouping of GSLB virtual servers, services, and other network entities. Typically, in a GSLB set up, there are many GSLB sites that are equipped to serve the same content to a client. These are usually geographically separated to ensure that the domain is active even if one site goes down completely. All of the sites in the GSLB configuration must be configured on every NetScaler appliance hosting a GSLB site. In other words, at each site, you configure the local GSLB site and each remote GSLB site.

Once GSLB sites are created for a domain, the NetScaler appliance sends client requests to the appropriate GSLB site as determined by the GSLB algorithms configured.

To create a GSLB site by using the command line interface

At the command prompt, type the following commands to create a GSLB site and verify the configuration:

- add gslb site <siteName> <siteIPAddress>
- show gslb site <siteName>

Example

```
add gslb site Site-GSLB-East-Coast 10.14.39.21
show glsb site Site-GSLB-East-Coast
```

To modify or remove a GSLB Site by using the command line interface

- To modify a GSLB site, use the set gslb site command, which is just like using the add gslb site command, except that you enter the name of an existing GSLB Site.
- To unset a site parameter, use the unset gslb site command, followed by the siteName value and the name of the parameter to be reset to its default value.
- To remove a GSLB site, use the rm gslb site command, which accepts only the <name> argument.

To configure a basic GSLB site by using the configuration utility

1. Navigate to Traffic Management > GSLB > Sites.
2. Add a new GSLB site, or select an existing GSLB site and edit its settings.

To view the statistics of a GSLB site by using the command line interface

At the command prompt, type:

stat gslb site <siteName>

Example

```
stat gslb site Site-GSLB-East-Coast
```

To view the statistics of a GSLB site by using the configuration utility

1. Navigate to Traffic Management > GSLB > Sites.
2. Select the GSLB site and click **Statistics**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a GSLB Service

Nov 24, 2014

A GSLB service is a representation of a load balancing or content switching virtual server. A local GSLB service represents a local load balancing or content switching virtual server. A remote GSLB service represents a load balancing or content switching virtual server configured at one of the other sites in the GSLB setup. At each site in the GSLB setup, you can create one local GSLB service and any number of remote GSLB services.

Creating GSLB Services

To create a GSLB service by using the command line interface

At the command prompt, type the following commands to create a GSLB service and verify the configuration:

- add gslb service <serviceName> <serverName | IP> <serviceType> <port>-siteName <string>
- show gslb service <serviceName>

Example

```
add gslb service Service-GSLB-1 10.14.39.14 HTTP 80 –siteName Site-GSLB-East-Coast
show gslb service Service-GSLB-1
```

To modify or remove a GSLB service by using the command line interface

- To modify a GSLB service, use the set gslb service <serviceName> command. For this command, specify the name of the GSLB service whose configuration you want to modify. You can change the existing values of the parameters either specified by you or set by default. You can change the value of more than one parameter in the same command. Refer to the add gslb service command for details about the parameters. Example
  ```
  > set gslb service SKP_GSLB_NOTCNAME_SVC2 -maxBandWidth 25 -maxClient 8
  Done
  > sh gslb service SKP_GSLB_NOTCNAME_SVC2
  SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
  ...
  Max Conn: 8  Max Bandwidth: 25 kbits
  ```
- To reset a parameter to its default value, you can use the unset gslb service <serviceName> command and the parameters to be unset. Example
  ```
  > unset gslb service SKP_GSLB_NOTCNAME_SVC2 maxBandWidth
  Done
  > sh gslb service SKP_GSLB_NOTCNAME_SVC2
  SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
  ...
  Max Conn: 8  Max Bandwidth: 0 kbits
  ```
- To remove a GSLB service, use the rm gslb service <serviceName> command.

To create a GSLB service by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services.
2. Add a new GSLB service, or select an existing service and edit its settings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To view the statistics of a GSLB service by using the command line interface

At the command prompt, type:

stat gslb service <serviceName>

Example

stat gslb service Service-GSLB-1

To view the statistics of a GSLB service by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services.
2. Select the GSLB Service and click **Statistics**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a GSLB Virtual Server

Jun 06, 2017

A GSLB virtual server is an entity that represents one or more GSLB services and balances traffic between them. It evaluates the configured GSLB methods or algorithms to select a GSLB service to which to send the client request.

> **Note**
>
> A GSLB virtual server protocol requirement is mainly to create a relation between the virtual server and the services that are bound to the virtual server. This also keeps CLI/APIs consistent with respect to other types of virtual servers. The Service Type parameter on a service or a virtual server is not leveraged while serving the DNS requests. It is instead referenced during site persistence, monitoring, and for the purpose of doing lookups via MEP.

Creating GSLB Virtual Servers

To create a GSLB virtual server by using the command line interface

At the command prompt, type the following commands to add a GSLB virtual server and verify the configuration:

- add gslb vserver <name> <serviceType> -ipType (IPv4 | IPv6)
- show gslb vserver <name>

Example

```
add gslb vserver Vserver-GSLB-1 HTTP -ipType IPv4
add gslb vserver Vserver-GSLB-2 HTTP -ipType IPv6
show gslb vserver Vserver-GSLB-1
show gslb vserver Vserver-GSLB-2
```

To modify or remove a GSLB virtual server by using the command line interface

- To modify a GSLB virtual server, use the set gslb vserver command, which is just like using the add gslb vserver command, except that you enter the name of an existing GSLB virtual server.
- To reset a parameter to is default value, you can use the unset gslb vserver command followed by the vserverName value and the name of the parameter to be unset.
- To remove a GSLB virtual server, use the rm gslb vserver command, which accepts only the <name> argument.

To configure a GSLB virtual server by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers.
2. Add a new GSLB virtual server, or select an existing GSLB virtual server and edit its settings.

To view the statistics of a GSLB virtual server by using the command line interface

At the command prompt, type:

stat gslb vserver <name>

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
stat gslb vserver Vserver-GSLB-1
```

To view the statistics of a GSLB virtual server by using the configuration utility

Navigate to Traffic Management > GSLB > Virtual Servers, select the virtual server and click **Statistics**.

Statistics of a GSLB service

When you run the stat gslb service command from the command line or click on the Statistics link from the configuration utility, the following details of the service will be displayed:

- **Request bytes.** Total number of request bytes received on this service or virtual server.
- **Response bytes.** Number of response bytes received by this service or virtual server.
- **Current client established connections.** Number of client connections in ESTABLISHED state.
- **Current load on the service.** Load on the service (Calculated from the load monitor bound to the service).

The data of number of requests and responses, and the number of current client and server connections may not be displayed or may not be synchronized with the data of the corresponding load balancing virtual server.

Clearing the GSLB virtual server or service statistics

Note: This feature is available in NetScaler release 10.5.e.
You can now clear the statistics of a GSLB virtual server and service. NetScaler ADC provides the following two options to clear the statistics:

- **Basic**: Clears the statistics that are specific to the virtual server but retains the statistics that are contributed by the bound GLSB service.
- **Full**: Clears both the virtual server and the bound GSLB service statistics.

To clear the statistics of a GSLB virtual server by using the command line interface

At the command prompt, type:

```
stat gslb vserver <name> -clearstats <basic | full>
```

Example

```
stat gslb vserver Vserver-GSLB-1 –clearstats basic
```

To clear the statistics of a GSLB service by using the command line interface

At the command prompt, type:

```
stat gslb service <name> -clearstats <basic | full>
```

Example

```
stat gslb service service-GSLB-1 –clearstats basic
```

To clear the statistics of a GSLB virtual server by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers.
2. Select the GSLB virtual server and, click **Statistics**, and then click **Clear**.
3. From the **Clear** drop-down list, select **Basic** or **Full**, and then click **OK**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To clear the statistics of a GSLB service by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services.
2. Select the GSLB service and, click **Statistics**, and then click **Clear**.
3. From the **Clear** drop-down list, select **Basic** or **Full**, and then click **OK**.

Enabling and Disabling GSLB Virtual Servers

Updated: 2014-11-21

When you create a GSLB virtual server, it is enabled by default. If you disable it, it cannot process traffic. A disabled GSLB virtual server is not included in GSLB configuration but is not removed from the NetScaler appliance.

To enable or disable a GSLB virtual server by using the command line interface

At the command prompt, type one of the following commands:

- enable gslb vserver <name>@
- disable gslb vserver <name>@

Example

```
enable gslb vserver Vserver-GSLB-1
disable gslb vserver Vserver-GSLB-1
```

To enable or disable a GSLB virtual server by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers.
2. Select a virtual server and, from the **Action** list, select **enable** or **disable**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding GSLB Services to a GSLB Virtual Server

Nov 21, 2014

Once the GSLB services and virtual server are configured, relevant GSLB services must be bound to the GSLB virtual server to activate the configuration.

To bind a GSLB service to a GSLB virtual server by using the command line interface

At the command prompt, type the following commands to bind a GSLB service to a GSLB virtual server and verify the configuration:

- bind gslb vserver <name> -serviceName <string>
- show gslb vserver <name>

Example

```
bind gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1
show gslb vserver Vserver-GSLB-1
```

To unbind a GSLB service from a GSLB virtual server by using the command line interface

At the command prompt, type:

```
unbind gslb vserver <name> -serviceName <string>
```

To bind GSLB services by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click a virtual server.
2. Click in the **Domains** section, and configure a domain and bind the domain.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding a Domain to a GSLB Virtual Server

Dec 09, 2014

To make a NetScaler appliance the authoritative DNS server for a domain, you must bind the domain to the GSLB virtual server. When you bind a domain to a GSLB virtual server, the NetScaler adds an address record for the domain, containing the name of the GSLB virtual server. The start of authority (SOA) and name server (NS) records for the GSLB domain must be added manually.

For details on configuring SOA and NS records, see Domain Name System.

To bind a domain to a GSLB virtual server by using the command line interface

At the command prompt, type the following commands to bind a domain to a GSLB virtual server and verify the configuration:

- bind gslb vserver <name> -domainName <string>
- show gslb vserver <name>

Example

```
bind gslb vserver Vserver-GSLB-1 -domainName www.mycompany.com
show gslb vserver Vserver-GSLB-1
```

To unbind a GSLB domain from a GSLB virtual server by using the command line interface

At the command prompt, type:

```
unbind gslb vserver <name> -domainName <string>
```

To bind a domain to a GSLB virtual server by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers.
2. In GSLB Virtual Servers pane, select the GSLB Virtual Server to which you want to bind the domain (for example, Vserver-GSLB-1) and click Open.
3. In the Configure GSLB Virtual Server dialog box, on the Domains tab, do one of the following:
   - To create a new Domain, click Add.
   - To modify an existing Domain, select the domain, and then click Open.
4. In the Create GSLB Domain or Configure GSLB Domain dialog box, specify values for the following parameters as shown:
   - Domain Name*—domainName (for example, www.mycompany.com)

   * A required parameter

5. Click Create.
6. Click OK.

To view the statistics of a domain by using the command line interface

At the command prompt, type:

```
stat gslb domain <name>
```

Example

```
stat gslb domain www.mycompany.com
```

Note: To view statistics for a particular GSLB domain, enter the name of the domain exactly as it was added to the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler appliance. If you do not specify the domain name, or if you specify an incorrect domain name, statistics for all configured GSLB domains are displayed.

To view the statistics of a domain by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers.
2. In GSLB Virtual Servers pane, select the GSLB Virtual Server (for example, Vserver-GSLB-1) and click Open.
3. In the Configure GSLB Virtual Server dialog box, on the Domains tab, select the domain, and then click Statistics.

To view the configuration details of the entities bound to a GSLB domain using the command line

Note: This feature is available in NetScaler release 10.5.e.
At the command prompt, type:

```
show gslb domain <name>
```

Example

```
show gslb domain gslb1.com
    gslb1.com
    gvs1 - HTTP    state: DOWN
    DNS Record Type: A
    Configured Method: LEASTCONNECTION
    Backup Method: ROUNDROBIN
    Persistence Type: NONE
    Empty Down Response: DISABLED
    Multi IP Response: DISABLED
    Dynamic Weights: DISABLED

    gsvc1 (10.102.239.165: 80)- HTTP State: DOWN    Weight: 1
        Dynamic Weight: 0    Cumulative Weight: 1
        Effective State: DOWN
        Threshold : BELOW

        Monitor Name : http
            State: DOWN   Weight: 1
            Probes: 144   Failed [Total: 144 Current: 144]
            Last response: Failure - TCP syn sent, reset received.
            Response Time: 2000 millisec

    gsvc2 (10.102.239.179: 80)- HTTP State: DOWN    Weight: 1
        Dynamic Weight: 0    Cumulative Weight: 1
        Effective State: DOWN
        Threshold : BELOW

        Monitor Name : http-ecv
            State: DOWN   Weight: 1
            Probes: 141   Failed [Total: 141 Current: 141]
            Last response: Failure - TCP syn sent, reset received.
            Response Time: 2000 millisec
Done
```

To view the configuration details of the entities bound to a GSLB domain by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: This feature is available in NetScaler release 10.5.e.

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click a virtual server.
2. Click the field below the **Domains** pane.
3. In the **GSLB Virtual Server Domain Binding** dialog box, select a domain, and then click **Show Bindings**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Synchronizing a Configuration in a GSLB Setup

May 11, 2017

Typically, a GSLB setup has a few data centers with a GSLB site configured for each data center. In each NetScaler, participating in GSLB, configure one GSLB site as a local site and the others as remote sites. When you add another GSLB site at a later point of time, ensure that all the GSLB sites have the same configuration. To have the same configuration on all the GSLB sites, you can use the NetScaler appliance's GSLB configuration synchronization option.

The NetScaler appliance from which you use the synchronization option is referred to as the 'master node' and the GSLB sites on which the configuration is copied as 'slave nodes'. When you synchronize a GSLB configuration, the configurations on all the GSLB sites participating in the GSLB setup are made similar to that on the master node.

Synchronization (may also be referred to as 'auto sync') is carried out in the following manner:

- The master node finds the differences between the configuration of the master node and slave node, and changes the configuration of the slave node to make it similar to the master node.
  If you force a synchronization (use the 'force sync' option), the NetScaler deletes the GSLB configuration from the slave node and then configures the slave to make it similar to the master node.
- During synchronization, if a command fails, synchronization is not aborted and the error message are logged into a **.err** file in the **/var/netscaler/gslb** directory.
- Synchronization is done only on the parent sites. If a GSLB site is configured as a child site, its configuration is not affected by synchronization.
- If you disable the internal user login, the GSLB auto sync uses the SSH keys to synchronize the configuration. But, to use GSLB auto sync in partition environment, you need to enable the internal user login and make sure that the partition username in the local and remote GSLB sites is same.

Note:

- On the remote GSLB site RPC node, configure the firewall to accept auto-sync connections by specifying the remote site IP (cluster IP address for cluster setup) and port (3010 for RPC and 3008 for secure RPC). The source IP address that will be used for auto-sync is the NSIP of the master node (NSIP of the configuration coordinator in a cluster setup). The destination IP is the site IP (remote site IP).
- The source IP address cannot be synchronized across the sites participating in GSLB because the source IP address for a RPC node is specific to each NetScaler appliance. Therefore, after you force a synchronization (using the sync gslb config –forceSync command or by selecting the ForceSync option in the NetScaler GUI), you have to manually change the source IP addresses on the other NetScaler appliances.
- Port 22 is also required for synchronizing the database files to the remote site.

If you use the saveconfig option, the sites that participate in the synchronization process automatically save their configuration, in the following way:

1. The master node saves its configuration immediately before it initiates the process of synchronization.
2. After the process of synchronization is complete, the slave nodes save their configuration. A slave node saves its configuration only if the configuration difference was applied successfully on it. If synchronization fails on a slave node, you must manually investigate the cause of the failure and take corrective action.

Limitations of synchronization:

- On the master node, the names of the remote GSLB sites must be identical to the names of sites configured on the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler appliances hosting those sites.
- During the synchronization, traffic disruptions may occur.
- NetScaler can synchronize only up to 80000 lines of the configuration.
- Synchronization may fail:
  - If the spill over method is changed from CONNECTION to DYNAMIC CONNECTION.
  - If you interchange the site prefix of the GSLB services bound to a GSLB virtual server on the master node and then try to synchronize.
  - If the RPC node passwords are different for NetScaler IP address (NSIP) and loopback IP address.
- If you have configured the GSLB sites as High Availability (HA) pairs, the RPC node passwords of primary and secondary nodes should be same.
- If you rename any GLSB entity that are part of your GSLB configuration (use “show gslb runningConfig” command to display the GSLB configuration). You need to use the force sync option to synchronize the configuration to other GSLB sites.

Note: To overcome the limitations due to some settings in the GSLB configuration, you can use the force sync option. But, if you use the force sync option the GSLB entities are removed and re-added to the configuration and the GSLB statistics are reset to zero. Hence the traffic is disrupted during the configuration change.
Before you start the synchronization of a GSLB setup, make sure that:
- On all the GSLB sites including the master node, management access and SSH should be enabled for the IP address of the corresponding GSLB site. The IP address of a GSLB site must be an IP address owned by the NetScaler.
  For more information about adding the GSLB site IP addresses and enabling Management Access, see Configuring a Basic GSLB Site.

- The GSLB configuration on the NetScaler appliance that is considered as the master node is complete and appropriate to be copied on all the sites.
- If you are synchronizing the GSLB configuration for the first time, all the sites participating in GSLB need to have the GSLB site entity of their respective local sites.
- You are not synchronizing sites that, by design, do not have the same configuration.

Important: After a GSLB configuration is synchronized, the configuration cannot be rolled back on any of the GSLB sites. Run the **sync gslb config** command only if you are sure that the synchronization process will not overwrite the configuration on the remote site. Site synchronization is undesirable when the local and remote sites have different configurations by design, and can lead to site outage. If some commands fail and some commands succeed, the successful commands cannot be rolled back.

To synchronize a GSLB configuration by using the command line interface

At the command prompt, type the following commands to synchronize GSLB sites and verify the configuration:

- sync gslb config [-preview | -forceSync <string> | -nowarn | -saveconfig] [-debug]
- show gslb syncStatus

Example

```
> sync gslb config
[WARNING]: Syncing config may cause configuration loss on other site.
Please confirm whether you want to sync-config (Y/N)? [N]:y
Sync Time: Dec 9 2011 10:56:9
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Retrieving local site info: ok
Retrieving all participating gslb sites info: ok
Gslb_site1[Master]:
    Getting Config: ok
Gslb_site2[Slave]:
    Getting Config: ok
    Comparing config: ok
     Applying changes: ok
Done
```

To synchronize a GSLB configuration by using the configuration utility

1. Navigate to Traffic Management > GSLB and, under GSLB Configuration, click **Synchronize configuration on remote sites** and synchronize the GSLB configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Metrics Exchange Protocol (MEP)

May 21, 2015

The data centers in a GSLB setup exchange metrics with each other through the metrics exchange protocol (MEP), which is a proprietary protocol for Citrix NetScaler appliances. The exchange of the metric information begins when you create a GSLB site. These metrics comprise load, network, and persistence information.

MEP is required for health checking of data centers to ensure their availability. A connection for exchanging network metric (round-trip time) can be initiated by either of the data centers involved in the exchange, but a connection for exchanging site metrics is always initiated by the data center with the lower IP address. By default, the data center uses a subnet IP (SNIP) address or a mapped IP (MIP) address to establish a connection to the IP address of a different data center. However, you can configure a specific SNIP, MIP, or virtual IP (VIP) address, or the NetScaler IP (NSIP) address as the source IP address for metrics exchange. The communication process between GSLB sites uses TCP port 3011 or 3009, so this port must be open on firewalls that are between the NetScaler appliances.

Note: You cannot configure a GSLB site IP address as the source IP address for site metrics exchange.

If the source and target sites (the site that initiates a MEP connection and the site that receives the connection request, respectively) have both private and public IP addresses configured, the sites exchange MEP information by using the public IP addresses.

You can also bind monitors to check the health of remote services as described in "Monitoring GSLB Services." When monitors are bound, metric exchange does not control the state of the remote service. If a monitor is bound to a remote service and metric exchange is enabled, the monitor controls the health status. Binding the monitors to the remote service enables the NetScaler appliance to interact with a non-NetScaler load balancing device. The NetScaler can monitor non-NetScaler devices but cannot perform load balancing on them unless monitors are bound to all GSLB services and only static load balancing methods (such as the round robin, static proximity, or hash-based methods) are used.

This document includes the following information:

- Enabling Site Metric Exchange
- Enabling Network Metric Information Exchange
- Enabling Persistence Information Exchange

Enbaling Site Metrics Exchange

Updated: 2014-11-24

Site metrics exchanged between the GSLB sites include the status of each load balancing and content switching virtual server, the current number of connections, the current packet rate, and current bandwidth usage information.

The NetScaler appliance needs this information to perform load balancing between the sites. The site metric exchange interval is 1 second. A remote GSLB service must be bound to a local GSLB virtual server to enable the exchange of site metrics with the remote service.

To enable or disable site metrics exchange by using the command line interface

At a command prompt, type the following commands to enable or disable site metrics exchange and verify the configuration:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **set gslb site** <siteName> **-metricExchange (ENABLED|DISABLED)**
- **show gslb site** <siteName>

Example

```
set gslb site Site-GSLB-East-Coast -metricExchange ENABLED
set gslb site Site-GSLB-East-Coast -metricExchange DISABLED
show gslb site Site-GSLB-East-Coast
```

To enable or disable site metrics exchange by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Sites**, and select the site.
2. In the **Configure GSLB Site** dialog box, select the **Metric Exchange** option.

Enabling Network Metric Information Exchange

Updated: 2014-11-24

If your GSLB sites use the round trip time (RTT) load balancing method, you can enable or disable the exchange of RTT information about the client's local DNS service. This information is exchanged every 5 seconds.

For details about changing the GSLB method to a method based on RTT, see Changing the GSLB Method.

To enable or disable network metric information exchange by using the command line interface

At the command prompt, type the following commands to enable or disable network metric information exchange and verify the configuration:

- **set gslb site** <siteName> **-nwmetricExchange (ENABLED|DISABLED)**
- **show gslb site** <siteName>

Example

```
set gslb site Site-GSLB-East-Coast -nwmetricExchange ENABLED
set gslb site Site-GSLB-East-Coast -nwmetricExchange DISABLED
show gslb site Site-GSLB-East-Coast
```

To enable or disable network metric information exchange by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Sites**.
2. In the **Configure GSLB Site** dialog box, select the **Network Metric Exchange** option.

Enabling Persistence Information Exchange

Updated: 2014-11-24

You can configure NetScaler appliance to provide persistent connections, so that a client transmission to any virtual server in a group can be directed to a server that has received previous transmissions from the same client.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

You can enable or disable the exchange of persistence information at each site. This information is exchanged every 5 seconds between NetScaler appliances participating in GSLB.

For details about configuring persistence, see Configuring Persistent Connections.

To enable/disable persistence information exchange by using the command line interface

At the command prompt, type the following commands to enable or disable persistence information exchange and verify the configuration:

- set gslb site <siteName> -sessionExchange (ENABLED | DISABLED)
- show gslb site <siteName>

Example

```
set gslb site Site-GSLB-East-Coast -sessionExchange  ENABLED
set gslb site Site-GSLB-East-Coast -sessionExchange  DISABLED
show gslb site Site-GSLB-East-Coast
```

To enable/disable persistence information exchange by using the configuration utility

1. Navigate to Traffic Management > GSLB > Sites, double-click the site.
2. In the **Configure GSLB Site** dialog box, select the **Persistence Session Entry Exchange** option.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Site-to-Site Communication

May 11, 2017

GSLB site-to-site communication is between the remote procedure call (RPC) nodes that are associated with the communicating sites. A master GSLB site establishes connections with slave sites to synchronize GSLB configuration information and to exchange site metrics.

An RPC node is created automatically when a GSLB site is created, and is assigned an internally generated user name and password. The NetScaler appliance uses this user name and password to authenticate itself to remote GSLB sites during connection establishment. No configuration steps are necessary for an RPC node, but you can specify a password of your choice, enhance security by encrypting the information that GSLB sites exchange, and specify a source IP address for the RPC node.

The appliance needs a NetScaler-owned IP address to use as the source IP address when communicating with other GSLB sites. By default, the RPC nodes use either a subnet IP (SNIP) address or a mapped IP (MIP) address, but you might want to specify an IP address of your choice.

The following topics describe the behavior and configuration of RPC nodes on the NetScaler appliance:

- Changing the Password of an RPC Node
- Encrypting the Exchange of Site Metrics
- Configuring the Source IP Address for an RPC Node

Changing the Password of an RPC Node

Updated: 2014-11-21

You can secure the communication between sites in your GSLB setup by changing the password of each RPC node. After you change the password for the RPC node of the local site, you must manually propagate the change to the RPC node at each of the remote sites.

The password is stored in encrypted form. You can verify that the password has changed by using the show rpcNode command to compare the encrypted form of the password before and after the change.

To change the password of an RPC node by using the command line interface

At the command line, type the following commands to change the password of an RPC node:

- set ns rpcNode <IPAddress> {-password}
- show ns rpcNode

Example

```
> set rpcNode 192.0.2.4 -password mypassword
 Done
> show rpcNode
.
.
.
2) IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 SrcIP:  *            Secure:  OFF
Done
>
```

To unset the password of an RPC node by using the command line interface

To unset the password of an RPC node by using the NetScaler command line, type the unset rpcNode command, the IP address of the RPC node, and the password parameter, without a value.

To change the password of an RPC node by using the configuration utility

Navigate to System > Network > RPC, select the RPC node, and change the password.

Encrypting the Exchange of Site Metrics

Updated: 2014-11-24

You can secure the information that is exchanged between GSLB sites by setting the secure option for the RPC nodes in the GSLB setup. With the secure option set, the NetScaler appliance encrypts all communication sent from the node to other RPC nodes.

To encrypt the exchange of site metrics by using the command line interface

At the command prompt, type the following commands to encrypt the exchange of site metrics and verify the configuration:

- set ns rpcNode <IPAddress> [-secure (YES | NO)]
- show rpcNode

Example

```
> set rpcNode 192.0.2.4 -secure YES
 Done
>
> show rpcNode
.
.
.
3) IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e SrcIP:  192.0.2.3     Secure:  ON
 Done
>
```

To unset the secure parameter by using the command line interface

To unset the secure parameter by using the NetScaler command line, type the unset rpcNode command, the IP address of the RPC node, and the secure parameter, without a value.

To encrypt the exchange of site metrics by using the NetScaler configuration utility

1. Navigate to System > Network > RPC and double-click a RPC node.
2. Select the **Secure** option, and click **OK**.

Configuring the Source IP Address for an RPC Node

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2014-11-24

By default, the NetScaler appliance uses a NetScaler-owned subnet IP (SNIP) address or mapped IP (MIP) address as the source IP address for an RPC node, but you can configure the appliance to use a specific SNIP address or MIP address. If neither a SNIP address nor a MIP address is available, the GSLB site cannot communicate with other sites. In such a scenario, you must configure either the NetScaler IP (NSIP) address or a virtual IP (VIP) address as the source IP address for an RPC node. A VIP address can be used as the source IP address of an RPC node only if the RPC node is a remote node. If you configure a VIP address as the source IP address and remove the VIP address, the appliance uses a SNIP address or a MIP address.

> **Note**
>
> From NetScaler 11.0.64.x release onwards, you can configure the appliance to use GSLB Site IP address as the source IP address for an RPC node.

To specify a source IP address for an RPC node by using the command line interface

At the command prompt, type the following commands to change the source IP address for an RPC node and verify the configuration:

- set ns rpcNode <IPAddress> [-srcIP <ip_addr|ipv6_addr|*>]
- show ns rpcNode

Example

```
> set rpcNode 192.0.2.4 -srcIP 192.0.2.3
Done
> show rpcNode
.
.
.
2) IPAddress: 192.0.2.4 Password: d336004164d4352ce39e SrcIP: 192.0.2.3 Secure: OFF
Done
>
```

To unset the source IP address parameter by using the command line interface

To unset the source IP address parameter by using the NetScaler command line, type the unset rpcNodecommand, the IP address of the RPC node, and the srcIP parameter, without a value.

To specify a source IP address for an RPC node by using the NetScaler configuration utility

1. Navigate to System > Network > RPC and double-click a RPC node.
2. In the Source IP Address field, enter the IP address that you want the RPC node to use as the source IP address and click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Important

The source IP address cannot be synchronized across the sites participating in GSLB because the source IP address for a RPC node is specific to each NetScaler appliance. Therefore, after you force a synchronization (using the sync gslb config –forceSync command or by selecting the ForceSync option in the NetScaler GUI), you have to manually change the source IP addresses on the other NetScaler appliances.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Customizing Your GSLB Configuration

Jul 05, 2017

Once your basic GSLB configuration is operational, you can customize it by modifying the bandwidth of a GSLB service, configuring CNAME based GSLB services, static proximity, dynamic RTT, persistent connections, or dynamic weights for services, or changing the GSLB Method.

You can also configure monitoring for GSLB services to determine their states.

These settings depend on your network deployment and the types of clients you expect to connect to your servers.

This document includes the following information:

- Modifying Maximum Connections or Maximum Bandwidth for a GSLB Service
- Creating CNAME-Based GSLB Services
- Configuring Transition Out-Of-Service State (TROFS) in GSLB
- Configuring Dynamic Weights for Services

Modifying Maximum Connections or Maximum Bandwidth for a GSLB Service

Updated: 2014-11-26

You can restrict the number of new clients that can simultaneously connect to a load balancing or content switching virtual server by configuring the maximum number of clients and/or the maximum bandwidth for the GSLB service that represents the virtual server.

To modify the maximum clients or bandwidth of a GSLB service by using the command line interface

At the command prompt, type the following command to modify the maximum number of client connections or the maximum bandwidth of a GSLB service and verify the configuration:

- set gslb service <serviceName> [-maxClients <positive_integer>] [-maxBandwidth <positive_integer>]
- show gslb service <serviceName>

Example

```
set glsb service Service-GSLB-1 –maxBandwidth 100 –maxClients 100
show gslb service Service-GSLB-1
```

To modify the maximum clients or bandwidth of a GSLB service by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services, and double-click a service.
2. Click in the **Other Settings** section and set the following parameters:
   - Max Clients—maxClients
   - Max Bandwidth—maxBandwidth

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating CNAME-Based GSLB Services

Updated: 2014-11-24

To configure a GSLB service, you can use the IP address of the server or a canonical name of the server. If you want to run multiple services (like an FTP and a Web server, each running on different ports) from a single IP address or run multiple HTTP services on the same port, with different names, on the same physical host, you can use canonical names (CNAMES) for the services.

For example, you can have two entries in DNS as ftp.example.com and www.example.com for FTP services and HTTP services on the same domain, example.com. CNAME-based GSLB services are useful in a multilevel domain resolver configuration or in multilevel domain load balancing. Configuring a CNAME-based GSLB service can also help if the IP address of the physical server is likely to change.

If you configure CNAME-based GSLB services for a GSLB domain, when a query is sent for the GSLB domain, the NetScaler appliance provides a CNAME instead of an IP address. If the A record for this CNAME record is not configured, the client must query the CNAME domain for the IP address. If the A record for this CNAME record is configured, the NetScaler provides the CNAME with the corresponding A record (IP address). The NetScaler appliance handles the final resolution of the DNS query, as determined by the GSLB method. The CNAME records can be maintained on a different NetScaler appliance or on a third-party system.

In an IP-address-based GSLB service, the state of a service is determined by the state of the server that it represents. However, a CNAME-based GSLB service has its state set to UP by default; the virtual server IP (VIP) address or metric exchange protocol (MEP) are not used for determining its state. If a desktop-based monitor is bound to a CNAME-based GSLB service, the state of the service is determined according to the result of the monitor probes.

You can bind a CNAME-based GSLB service only to a GSLB virtual server that has the DNS Record Type as CNAME. Also, a NetScaler appliance can contain at most one GSLB service with a given CNAME entry.

The following are some of the features supported for a CNAME-based GSLB service:

- GSLB-policy based site affinity is supported, with the CNAME as the preferred location.
- Source IP persistence is supported. The persistency entry contains the CNAME information instead of the IP address and port of the selected service.

The following are the limitations of CNAME-based GSLB services:

- Site persistence is not supported, because the service referenced by a CNAME can be present at any third-party location.
- Multiple-IP-address response is not supported because one domain cannot have multiple CNAME entries.
- Source IP Hash and Round Robin are the only load balancing methods supported. The Static Proximity method is not supported because a CNAME is not associated with an IP address and static proximity can be maintained only according to the IP addresses.

Note: The Empty-Down-Response feature should be enabled on the GSLB virtual server to which you bind the CNAME-based GSLB service. If you enable the Empty-Down-Response feature, when a GSLB virtual server is DOWN or disabled, the response to a DNS query, for the domains bound to this virtual server, contains an empty record without any IP addresses, instead of an error code.

To create a CNAME-based GSLB service by using the command line interface

At the command prompt, type:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add gslb service <serviceName> -cnameEntry <string> -siteName <string>

Example

```
add gslb service Service-GSLB-1 -cnameEntry transport.mycompany.com -siteName Site-GSLB-East-Coast
add gslb service Service-GSLB-2 -cnameEntry finance.mycompany.com -siteName Site-GSLB-West-Coast
```

To create a CNAME-based GSLB service by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services.
2. Create a service, and set the **Type to Canonical Name Based**.

Configuring Transition Out-Of-Service State (TROFS) in GSLB

When you configure persistence on a GSLB virtual server to which a service is bound, the service continues to serve requests from the client even after it is disabled, accepting new requests or connections only to honor persistence. After a configured period of time, known as the graceful shutdown period, no new requests or connections are directed to the service, and all of the existing connections are closed.

When disabling a service, you can specify a graceful shutdown period, in seconds, by using the delay argument. During the graceful shutdown period, if the service is bound to a virtual server, its state appears as Out of Service.

Configuring Dynamic Weights for Services

Updated: 2015-06-02

In a typical network, there are servers that have a higher capacity for traffic than others. However, with a regular load balancing configuration, the load is evenly distributed across all services even though different services represent servers with different capacities.

To optimize your GSLB resources, you can configure dynamic weights on a GSLB virtual server. The dynamic weights can be based on either the total number of services bound to the virtual server or the sum of the weights of the individual services bound to the virtual server. Traffic distribution is then based on the weights configured for the services.

When dynamic weights are configured on the GSLB virtual server, requests are distributed according to the load balancing method, the weight of the GSLB service, and the dynamic weight. The product of the weight of the GSLB service and the dynamic weight is known as the cumulative weight. Therefore, when dynamic weight is configured on the GSLB virtual server, requests are distributed on the basis of the load balancing method and the cumulative weight.

When dynamic weight for a virtual server is disabled, the numerical value is set to 1. This ensures that the cumulative weight is a non-zero integer at all times.

Dynamic weight can be based on the total number of active services bound to load balancing virtual servers or on the weights assigned to the services.

Consider a configuration with two GSLB sites configured for a domain and each site has two services that can serve the client. If a service at either site goes down, the other server in that site has to handle twice as much traffic as a service at the other site. If dynamic weight is based on the number of active services, the site with both services active has twice the weight of the site with one service down and therefore receives twice as much traffic.

Alternatively, consider a configuration in which the services at the first site represent servers that are twice as powerful as servers at the second site. If dynamic weight is based on the weights assigned to the services, twice as much traffic can be

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

sent to the first site as to the second.

Note: For details on assigning weights to load balancing services, see "Assigning Weights to Services".
As an illustration of how dynamic weight is calculated, consider a GSLB virtual server that has a GSLB service bound to it. The GSLB service represents a load balancing virtual server that in turn has two services bound to it. The weight assigned to the GSLB service is 3. The weights assigned to the two services are 1 and 2 respectively. In this example, when dynamic weight is set to:

- **Disabled**:The cumulative weight of the GSLB virtual server is the product of the dynamic weight (disabled = 1) and the weight of the GSLB service (3), so the cumulative weight is 3.
- **SERVICECOUNT**: The count is the sum of the number of services bound to the load balancing virtual servers corresponding to the GSLB service (2), and the cumulative weight is the product of the dynamic weight (2) and the weight of the GSLB service (3), which is 6.
- **SERVICEWEIGHT**: The dynamic weight is the sum of the number of services bound to the GSLB service (2), and the cumulative weight is the product of the dynamic weight (2) and the weight of the GSLB service (3), which is 6.

Note: Dynamic weights are not applicable when content switching virtual servers are configured.

To configure a GSLB virtual server to use dynamic weights by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -dynamicWeight SERVICECOUNT | SERVICEWEIGHT
```

Example

```
set gslb vserver vserver-GSLB-1 -dynamicWeight SERVICECOUNT
```

To set GSLB virtual server to use dynamic weights by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers, double-click the GSLB virtual server whose method you want to change (for example, vserver-GSLB-1).
2. Click the **Method** section and, from the **Dynamic Weight** drop-down list, select **SERVICECOUNT** or **SERVICEWEIGHT**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Changing the GSLB Method

May 19, 2016

Unlike traditional DNS servers that simply respond with the IP addresses of the configured servers, a NetScaler appliance configured for GSLB responds with the IP addresses of the services, as determined by the configured GSLB method. By default, the GSLB virtual server is set to the least connection method. If all GSLB services are down, the NetScaler responds with the IP addresses of all the configured GSLB services.

GSLB methods are algorithms that the GSLB virtual server uses to select the best-performing GSLB service. After the host name in the Web address is resolved, the client sends traffic directly to the resolved service IP address.

The NetScaler appliance provides the following GSLB methods:

- Round Robin
- Least Connections
- Least Response Time
- Least Bandwidth
- Least Packets
- Source IP Hash
- Custom Load
- Round Trip Time (RTT)
- Static Proximity

For GSLB methods to work with a remote site, either MEP must be enabled or explicit monitors must be bound to the remote services. If MEP is disabled, RTT, Least Connections, Least Bandwidth, Least Packets and Least Response Time methods default to Round Robin.

The Static Proximity and RTT load balancing methods are specific to GSLB.

Specifying a GSLB Method Other than Static Proximity or Dynamic (RTT)

For information about the Round Robin, Least Connections, Least Response Time, Least Bandwidth, Least Packets, Source IP Hash, or Custom Load method, see Load Balancing.

To change the GSLB method by using the command line interface

At the command prompt, type:

set gslb vserver <name> -lbMethod GSLBMethod

Example

```
set gslb vserver Vserver-GSLB-1 -lbMethod ROUNDROBIN
```

To change the GSLB method by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers.
2. In the details pane, select a GSLB virtual server and click Open.
3. In the Configure GSLB Virtual Server dialog box, on the Method and Persistence tab, under Method, select a method from the Choose Method list.
4. Click OK, and verify that the method you selected appears under Details at the bottom of the screen.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Static Proximity

Jul 16, 2017

The static proximity method for GSLB uses an IP-address based static proximity database to determine the proximity between the client's local DNS server and the GSLB sites. The NetScaler appliance responds with the IP address of a site that best matches the proximity criteria.

If two or more GSLB sites at different geographic locations serve the same content, the NetScaler appliance maintains a database of IP address ranges and uses the database for decisions about the GSLB sites to which to direct incoming client requests.

For the static proximity method to work, you must either configure the NetScaler appliance to use an existing static proximity database populated through a location file or add custom entries to the static proximity database. After adding custom entries, you can set their location qualifiers. After configuring the database, you are ready to specify static proximity as the GSLB method.

This document includes the following information:

- Adding a Location File to Create a Static Proximity Database
- Adding Custom Entries to a Static Proximity Database
- Setting the Location Qualifiers
- Specifying the Proximity Method
- Synchronizing GSLB Static Proximity Database

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding a Location File to Create a Static Proximity Database

Oct 09, 2016

A static proximity database is a UNIX-based ASCII file. Entries added to this database from a location file are called static entries. Only one location file can be loaded on a NetScaler appliance. Adding a new location file overrides the existing file. The number of entries in the static proximity database is limited by the configured memory in the NetScaler appliance.

The static proximity database can be created in the default format or in a format derived from commercially configured third party databases (such as www.maxmind.com and www.ip2location.com).

The NetScaler ADC includes an IP geolocation database, GeoLite2 (published by MaxMind). The database is available in a format supported by NetScaler ADC at: **/var/netscaler/inbuilt_db/Citrix_Netscaler_InBuilt_GeoIP_DB.csv**. You can use this IP geolocation database as the location file for the static proximity based GSLB method or in location based policies.

Note: The ADC includes the GeoLite2 database available from http://www.maxmind.com.

These databases vary in the details they provide. There is no strict enforcement of the database file format, except that the default file has format tags. The database files are ASCII files that use a comma as the field delimiter. There are differences in the structure of fields and the representation of IP addresses in the locations.

The format parameter describes the structure of the file to the NetScaler appliance. Specifying an incorrect value for the format option can corrupt the internal data.

Note: The default location of the database file is /var/netscaler/locdb, and on a high availability (HA) setup, an identical copy of the file must be present in the same location on both NetScaler appliances.
The following abbreviations are used in this section:

- **CSHN**. Short name of a country based on the country code standard of ISO-3166.
- **LCN**. Long name of the country.
- **RC**. Region code based on ISO-3166-2 (for US and Canada). The region code “FIPS-10-4” is used for the other regions.

Note: Some databases provide short country names according to ISO-3166 and long country names as well. The NetScaler uses short names when storing and matching qualifiers.
To create a static proximity database, log on to the UNIX shell of the NetScaler appliance and use an editor to create a file with the location details in one of the NetScaler-supported formats.

To add a static location file by using the command line interface

At the command prompt, type:

- add locationFile <locationFile> [-format <format>]
- show locationFile

Example

```
> add locationFile /var/nsmap/locdb/nsgeo1.0 -format netscaler
 Done
> show locationFile
Location File: /var/nsmap/locdb/nsgeo1.0
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Format: netscaler
 Done
>
```

To add a static location file by using the configuration utility

1. Navigate to AppExpert > Location, click the **Static Database** tab.
2. Click **Add** to add a static location file.

You can view an imported location file database by using the View Database dialog box in the configuration utility. There is no NetScaler command line equivalent.

To view a static location file by using the configuration utility

1. Navigate to AppExpert > Location, click the **Static Database** tab.
2. Select a static location file, and from the **Action** list, click **View Database**.

To convert a location file into the netscaler format

By default, when you add a location file, it is saved in the netscaler format. You can convert a location file of other formats into the netscaler format.

Note: The nsmap option can be accessed only from the command line interface. The conversion is possible only into the netscaler format.
To convert the static database format, at the NetScaler command prompt, type the following command:

```
nsmap -f <inputFileFormat> -o <outputFileName> <inputFileName>
```

Example

```
 nsmap -f ip-country-region-city -o nsfile.ns ip-country-region-city.csv
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding Custom Entries to a Static Proximity Database

Jan 22, 2015

Custom entries take precedence over static entries in the proximity database. You can add a maximum of 500 custom entries. For a custom entry, denote all omitted qualifiers with an asterisk (*) and, if qualifiers have a period or space in the name, enclose the parameter in double quotation marks. The first 31 characters are evaluated for each qualifier. You can also provide the longitude and latitude of the geographical location of the IP address-range for selecting a service with the static proximity GSLB method.

To add custom entries by using the command line interface

At the command prompt, type the following commands to add a custom entry to the static proximity database and verify the configuration:

- add location < IPfrom> < IPto> <preferredLocation> [-longitude <integer>[-latitude <integer>]]
- show location

Example

```
>add location 192.168.100.1 192.168.100.100 *.us.ca.mycity
>show location
```

Parameters for adding custom entries

IPfrom

First IP address in the range, in dotted decimal notation. This is a mandatory argument.

IPto

Last IP address in the range, in dotted decimal notation. This is a mandatory argument.

preferredLocation

String of qualifiers, in dotted notation, describing the geographical location of the IP address range. Each qualifier is more specific than the one that precedes it, as in continent.country.region.city.isp.organization. For example,"NA.US.CA.San Jose.ATT.citrix".

Note: A qualifier that includes a dot (.) or space ( ) must be enclosed in double quotation marks.

This is a mandatory argument. Maximum Length: 197

longitude

Numerical value, in degrees, specifying the longitude of the geographical location of the IP address-range.

Note: Longitude and latitude parameters are used for selecting a service with the static proximity GSLB method. If they are not specified, selection is based on the qualifiers specified for the location.

Maximum value: 180

latitude

Numerical value, in degrees, specifying the latitude of the geographical location of the IP address-range.

Note: Longitude and latitude parameters are used for selecting a service with the static proximity GSLB method. If they are not specified, selection is based on the qualifiers specified for the location.

Maximum value: 180

To add custom entries by using the configuration utility

Navigate to AppExpert > Location, click the **Custom Entries** tab, and add the custom entries.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting the Location Qualifiers

Nov 24, 2014

The database used to implement static proximity contains the location of the GSLB sites. Each location contains an IP address range and up to six qualifiers for that range. The qualifiers are literal strings and are compared in a prescribed order at run time. Every location must have at least one qualifier. The meaning of the qualifiers (context) is defined by the qualifier labels, which are user defined. The NetScaler has two built-in contexts:

Geographic context, which has the following qualifier labels:

- Qualifier 1 – “Continent”
- Qualifier 2 – “Country”
- Qualifier 3 – “State”
- Qualifier 4 – “City”
- Qualifier 5 – “ISP”
- Qualifier 6 – “Organization”

Custom entries, which have the following qualifier labels:

- Qualifier 1 – “Qualifier 1”
- Qualifier 2 – “Qualifier 2”
- Qualifier 3 – “Qualifier 3”
- Qualifier 4 – “Qualifier 4”
- Qualifier 5 – “Qualifier 5”
- Qualifier 6 – “Qualifier 6”

If the geographic context is set with no Continent qualifier, Continent is derived from Country. Even the built-in qualifier labels are based on the context, and the labels can be changed. These qualifier labels specify the locations mapped with the IP addresses used to make static proximity decisions.

To perform a static proximity-based decision, the NetScaler appliance compares the location attributes (qualifiers) derived from the IP address of the local DNS server resolver with the location attributes of the participating sites. If only one site matches, the appliance returns the IP address of that site. If there are multiple matches, the site selected is the result of a round robin on the matching GSLB sites. If there is no match, the site selected is a result of a round robin on all configured sites. A site that does not have any qualifiers is considered a match.

To set the location qualifiers by using the command line interface

At the command prompt, type:

set locationparameter -context <context> -q1label <string> [-q2label <string>] [-q3label <string>] [-q4label <string>] [-q5label <string>] [-q6label <string>]

Example

```
set locationparameter -context custom -q1label asia
```

To set the location qualifiers by using the configuration utility

1. Navigate to AppExpert > Location.
2. From the **Action** list, click **Location Parameters** and set the location qualifiers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Specifying the Proximity Method

Nov 24, 2014

When you have configured the static proximity database, you are ready to specify static proximity as the GLSB method.

To specify static proximity by using the command line interface

At the command prompt, type the following commands to configure static proximity and verify the configuration:

- set gslb vserver <name> -lbMethod STATICPROXIMITY
- show gslb vserver <name>

Example

```
set gslb vserver Vserver-GSLB-1 -lbMethod STATICPROXIMITY
show gslb vserver
```

To specify static proximity by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server.
2. Click the **Method** section and from the **Choose Method** drop-down list, select **STATICPROXIMITY**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Synchronizing GSLB Static Proximity Database

Apr 08, 2014

Synchronizing a global server load balancing (GSLB) static proximity database requires that one of the sites be identified as the master GSLB node. Any site in the topology can be designated as the master node. The rest of the GSLB nodes are automatically designated as slave nodes.

Synchronizing GSLB static proximity databases synchronizes the files in the /var/netscaler/locdb directory across the slave nodes. During the synchronization process, the master node fetches the running configuration from each of the slave nodes and compares it to the configuration on the master node. The master GSLB node uses the rsync program to synchronize the static proximity database across the slave nodes. To speed up the synchronization process, the rsync program makes only enough changes to eliminate the differences between the two files. The synchronization process cannot be rolled back.

The following example synchronizes Site2, which is a slave site, to master site Site1. The administrator enters the **sync gslb config** command on Site1:

```
sync gslb config -nowarn
Sync Time: Feb 24 2014 14:56:16
Retrieving local site info: ok
Retrieving all participating gslb sites info:
0 bytes in 0 blocks
ok
site1[Master]:
    Getting Config: ok
site2[Slave]:
    Syncing gslb static proximity database: ok
    Getting Config: ok
    Comparing config: ok
    Applying changes: ok
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Dynamic Method (RTT)

May 22, 2015

Dynamic round trip time (RTT) is a measure of time or delay in the network between the client's local DNS server and a data resource. To measure dynamic RTT, the NetScaler appliance probes the client's local DNS server and gathers RTT metric information. The appliance then uses this metric to make its load balancing decision. Global server load balancing monitors the real-time status of the network and dynamically directs the client request to the data center with the lowest RTT value.

When a client's DNS request for a domain comes to the NetScaler appliance configured as the authoritative DNS for that domain, the appliance uses the RTT value to select the IP address of the best performing site to send it as a response to the DNS request.

The NetScaler appliance uses different mechanisms, such as ICMP echo request / reply (PING), UDP, and TCP to gather the RTT metrics for connections between the local DNS server and participating sites. The appliance first sends a ping probe to determine the RTT. If the ping probe fails, a DNS UDP probe is used. If that probe also fails, the appliance uses a DNS TCP probe.

These mechanisms are represented on the Netscaler appliance as Load Balancing Monitors and are easily identified due to their use of the "ldns" prefix. The three monitors, in their default order, are:

- ldns-ping
- ldns-dns
- ldns-tcp

These monitors are built in to the appliance and are set to safe defaults, but may be customized just like any other monitor on the appliance.

The default order may also be changed by setting it explicitly as a GSLB parameter. For example, to set the order to be the DNS UDP query followed by the PING and then TCP, type the following command:

set gslb parameter -ldnsprobeOrder DNS PING TCP

Unless they have been customized, the NetScaler appliance performs UDP and TCP probing on port 53, however unlike regular load balancing monitors the probes need not be successful in order to provide valid RTT information. ICMP port unavailable messages, TCP Resets and DNS error responses, which would usually constitute a failure are all acceptable for calculating the RTT value.

Once the RTT data has been compiled, the Netscaler uses the proprietary metrics exchange protocol (MEP) to exchange RTT values between participating sites. After calculating RTT metrics, the appliance sorts the RTT values to identify the data center with the best (smallest) RTT metric."

If RTT information is not available (for example, when a client's local DNS server accesses the site for the first time), the NetScaler appliance selects a site by using the round robin method and directs the client to the site.

To configure the dynamic method, you configure the site's GSLB virtual server for dynamic RTT. You can also set the interval at which local DNS servers are probed to a value other than the default.

This document includes the following information:

- Configuring a GSLB Virtual Server for Dynamic RTT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Setting the Probing Interval of Local DNS Servers

Configuring a GSLB Virtual Server for Dynamic RTT

Updated: 2014-11-24

To configure a GSLB virtual server for dynamic RTT, you specify the RTT load balancing method.

The NetScaler appliance regularly validates the timing information for a given local server. If a change in latency exceeds the configured tolerance factor, the appliance updates its database with the new timing information and sends the new value to other GSLB sites by performing a MEP exchange. The default tolerance factor is 5 milliseconds (ms).

The RTT tolerance factor must be the same throughout the GSLB domain. If you change it for a site, you must configure identical RTT tolerance factors on all NetScaler appliances deployed in the GSLB domain.

To configure a GSLB virtual server for dynamic RTT by using the command line interface

At the command prompt, type:

set gslb vserver <name> -lbMethod RTT -tolerance <value>
Example

```
set gslb vserver Vserver-GSLB-1 -lbMethod RTT -tolerance 10
```

To configure a GSLB virtual server for dynamic RTT by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server.

Setting the Probing Interval of Local DNS Servers

Updated: 2014-11-24

The NetScaler appliance uses different mechanisms, such as ICMP echo request / reply (PING), TCP, and UDP to obtain RTT metrics for connections between the local DNS server and participating GSLB sites. By default, the appliance uses a ping monitor and probes the local DNS server every 5 seconds. The appliance then waits 2 seconds for the response and, if a response is not received in that time, it uses the TCP DNS monitor for probing.

However, you can modify the time interval for probing the local DNS server to accommodate your configuration.

To modify the probing interval by using the command line interface

At the command prompt, type:

set lb monitor <monitorName> <type> -interval <integer> <units> -resptimeout <integer> <units>
Example

```
set lb monitor monitor-HTTP-1 HTTP -interval 10 sec -resptimeout 5 sec
```

To modify the probing interval by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Navigate to Traffic Management > Load Balancing > Monitors, and double-click the monitor that you want to modify (for example, ping).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Persistent Connections

Jun 03, 2015

Persistence ensures that a series of client requests for a particular domain name is sent to the same data center instead of being load balanced. If persistence is configured for a particular domain, it takes precedence over the configured GSLB method. Persistence is useful for deployments that deal with e-commerce, such as shopping card usage, where the server needs to maintain the state of the connection to track the transaction. To maintain the state of connection, you must configure persistence on a virtual server. With persistence configured, NetScaler selects a data center to process a client request and forwards the IP address of the selected data center for all subsequent DNS requests. If the configured persistence applies to a site that is down, the NetScaler appliance uses a GSLB method to select a new site, and the new site becomes persistent for subsequent requests from the client.

The GSLB virtual server is responsible for DNS-based site persistence, and it controls the site persistence for a remote GSLB service. The NetScaler appliance supports persistence based on the source IP address or on HTTP cookies.

When you bring a physical service DOWN with a delay time, the physical service goes into the transition out of service (TROFS) state. Site persistence is supported as long as the service is in the TROFS state. That is, if the same client sends a request for the same service within the specified delay time after a service is marked TROFS, the same GSLB site (data center) services the request.

Note: If connection proxy is specified as the site persistence method and if you also want to configure persistence of the physical servers, do not configure SOURCEIP persistence. When the connection is proxied, an IP address owned by the NetScaler is used, and not the actual IP address of the client. Configure methods such as cookie persistence or rule-based persistence on the load balancing virtual server.

Configuring Persistence Based on Source IP Address

Updated: 2014-11-24

With source-IP persistence, when a DNS request is received at a data center, the NetScaler appliance first looks for an entry in the persistence table and, if an entry for the local DNS server exists and the server mentioned in the entry is configured, the IP address of that server is sent as the DNS response.

For the first request from a particular client, the NetScaler appliance selects the best GSLB site for the request and sends its IP address to the client. Since persistence is configured for the source IP address of the client, all subsequent requests by that client or another local DNS server in the same IP subnet are sent the IP address of the GSLB site that was selected for the first request.

For source-IP address based persistence, the same set of persistence identifiers must be configured on the GSLB virtual servers in all data centers. A persistence identifier is a number used by the data centers to identify a particular GSLB virtual server. A cookie transmits the persistence identifier, enabling the NetScaler appliance to identify the domain so that it can forward all appropriate requests to the same domain. When persistence is enabled, the persistence information is also exchanged as part of metrics exchange.

For the NetScaler appliance to support persistence across sites, persistence must be enabled on the GSLB virtual servers of all participating sites. When you use source IP address persistence on the network identifier, you must configure a subnet mask. For any domain, persistence takes precedence over any other configured GSLB method.

To configure persistence based on source IP address by using the command line

© 1999-2017 Citrix Systems, Inc. All rights reserved.

interface

At the command prompt, type:

set gslb vserver <name> -persistenceType (SOURCEIP | NONE) -persistenceId <positive_integer> [-persistMask <netmask>] –[timeout <mins>]

Example

```
set gslb vserver vserver-GSLB-1 -persistenceType SOURCEIP -persistenceId 23 -persistMask 255.255.255.255 –timeout 2
```

To configure persistence based on source IP address by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the GSLB virtual server whose method you want to change (for example, vserver-GSLB-1).
2. Click the **Persistence** section and, from the **Persistence** drop-down list, select **SOURCEIP** and set the following parameters:
   - Persistence Id—persistenceID
   - Time-out—timeout
   - IPv4 Netmask or IPv6 Mask length—persistMask

Configuring Persistence Based on HTTP Cookies

Updated: 2014-11-26

The NetScaler appliance provides persistence at the HTTP-request level by using connection proxy and HTTP redirect. With these persistence methods, the appliance uses an HTTP cookie (known as a “site cookie”) to reconnect the client to the same server. The NetScaler inserts the site cookie in the first HTTP response.

The site cookie contains information about the selected GSLB service on which the client has a persistent connection. The cookie expiration is based on the cookie timeout configured on the NetScaler appliance. If the virtual server names are not identical on all the sites, you must use the persistence identifier. Cookies inserted are compliant with RFC 2109.

When the NetScaler appliance responds to a client DNS request by sending the IP address of the selected GSLB site, the client sends an HTTP request to that GSLB site. The physical server in that GSLB site adds a site cookie to the HTTP header, and connection persistence is in effect.

If the DNS entry in the client cache expires, and then the client sends another DNS query and is directed to a different GSLB site, the new GSLB site uses the site cookie present in the client request header to implement persistence. If the GSLB configuration at the new site uses connection-proxy persistence, the new site creates a connection to the GSLB site that inserted the site cookie, proxies the client request to the original site, receives a response from the original GSLB site, relays that response back to the client, and closes the connection. If the GSLB configuration uses HTTP redirect persistence, the new site redirects the request to the site that originally inserted the cookie.

Note: Connection proxy persistence can be configured only for local services. However, connection proxy persistence must be enabled on both local and remote GSLB services that are configured for the GSLB virtual server.
Connection proxy occurs when the following conditions are satisfied:

- Requests are sent from a domain participating in GSLB. The domain is obtained from the URL/Host header.
- Requests are sent from a local GSLB service whose public IP address matches the public IP address of an active service

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

bound to the GSLB virtual server.
- The local GSLB service has connection proxy enabled.
- The request includes a valid cookie that contains the IP address of an active remote GSLB service.

If one of the conditions is not met, connection proxy does not occur, but a site cookie is added if the local GSLB service has connection proxy enabled AND:
- No site cookie is supplied; OR,
- The site cookie refers to an IP address that is not an active GSLB remote service; OR,
- The cookie refers to the IP address of the virtual server on which the request is received.

The following are the limitations of using connection proxy site cookies:
- Site cookies do not work for non-HTTP(S) protocols.
- If an HTTP request is sent to a back-up virtual server, the virtual server does not add a cookie.
- Site cookies do not work if SSL client authentication is required.
- At the local site, the statistics for a GSLB service on a remote site are not the same as the statistics recorded for that service at the remote site. At the local site, the statistics for a remote GSLB service are slightly higher than the statistics that the remote site records for that same service.

Redirect persistence can be used only:
- For HTTP or HTTPS protocols.
- If the domain name is present in the request (either in the URL or in the HOST header), and the domain is a GSLB domain.
- When the request is received on a backup VIP or a GSLB local service that is in the down state.

> ### Note
>
> In a GSLB parent-child configuration, connection proxy works as intended even when a GSLB service is not configured on a child site. However, if you have additional configuration such as client authentication, client IP address insertion, or other SSL-specific requirement, you must add an explicit GSLB service on the site and configure it accordingly.
>
> For more information about parent-child topology, see Configuring Parent-Child Topology.

To set persistence based on HTTP cookies by using the command line interface

At the command prompt, type:

set gslb service <serviceName> -sitePersistence (ConnectionProxy [-sitePrefix <prefix>] | HTTPredirect -sitePrefix <prefix>)

Example

set gslb service service-GSLB-1 -sitePersistence ConnectionProxy
set gslb service service-GSLB-1 -sitePersistence HTTPRedirect -sitePrefix vserver-GSLB-1

To set persistence based on cookies by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services** and select the service that you want to configure for site persistence (for example, service-GSLB-1).
2. Click the **Site Persistence** section and set persistence based on cookies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Overriding Static Proximity Behavior by Configuring Preferred Locations

Aug 06, 2017

You might want to direct traffic from a local DNS (LDNS) server or network to a GSLB service other than the GSLB service that the static proximity method selects for that traffic. That is, you have a *preferred location* for that traffic. To override the static proximity method with preferred locations, you can do the following:

1. Configure a DNS action that consists of a list of preferred locations. For more information about configuring a DNS action, see Configuring a DNS Action.

2. Configure a DNS policy to identify the traffic arriving from the LDNS server or network for which you want to override static proximity, and apply the action in the policy.
3. Bind the policy to the global request bind point.

In the DNS action, you can configure a list of up to 8 preferred locations. The locations must be provided in the dotted qualifier notation, which is the notation in which you add custom locations to the static proximity database. The locations can include wildcards for qualifiers that you want to omit. For information about the dotted qualifier notation for locations, see Adding Custom Entries to a Static Proximity Database. When entering the preferred locations, you must enter them in the descending order of priority.

When a policy evaluates to TRUE, the NetScaler appliance matches the preferred locations, in priority order, with the locations of GSLB services. Matches are of the following two types:

- If all the non-wildcard qualifiers in a preferred location match the corresponding qualifiers in the location of a GSLB service, the match is considered a perfect match. For example, a GSLB service location of *.UK.*.* or Europe.UK.*.* is a perfect match for the preferred location *.UK.*.*.
- If only a subset of the non-wildcard qualifiers match, the match is considered a partial match. For example, a GSLB service location of Europe.EG is a partial match for the preferred location Europe.UK.

When a DNS policy evaluates to TRUE, the following algorithm is used to select a GSLB service:

1. The appliance evaluates the preferred location that has the highest priority and moves down the priority order until a perfect match is found between a preferred location and the location of a GSLB service.

   If a perfect match is found, the appliance checks whether the corresponding GSLB service is up. If it is up, it returns the IP address of the GSLB service in the DNS response. If multiple perfect matches are found (which can happen when one or more wildcards are used in a preferred location), the appliance checks the state of each of the corresponding GSLB services and load balances the GSLB services that are up.

2. If a perfect match is not found for any of the preferred locations, the appliance returns to the preferred location that has the highest priority and moves down the priority order until a partial match is found between a preferred location and the location of a GSLB service.

   If a partial match is found, the appliance checks whether the corresponding GSLB service is up. If it is up, it returns the IP address of the GSLB service in the DNS response. If multiple partial matches are found, the appliance checks the state of each of the corresponding GSLB services and load balances the GSLB services that are up.

3. If none of the perfect and partial matches are up, the appliance load balances all other available GSLB services.

In this way, the appliance implements a type of site affinity for traffic that matches the DNS policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

Consider a GSLB configuration that consists of the following eight GSLB services:

- Asia.IN
- Asia.JPN
- Asia.HK
- Europe.UK
- Europe.RU
- Europe.EG
- Africa.SD
- Africa.ZMB

Further consider the following DNS action and policy configuration:

```
> add dns action prefLoc11 GslbPrefLoc -preferredLocList "Asia.HK" "Europe.UK"
 Done
> add dns policy dnsPolPrefLoc "CLIENT.IP.SRC.MATCHES_LOCATION(\"*.ZMB.*.*\")" prefLoc11
 Done
```

When the appliance receives a request from the location *.ZMB.*.*, the preferred locations are evaluated as follows:

1. The appliance attempts to find a GSLB service whose location is a perfect match for Asia.HK, which is the preferred location that has the highest priority. It finds that the GSLB service at Asia.HK is a perfect match. If the GSLB service is up, it sends the client the IP address of the GSLB service.
2. If the GSLB service at Asia.HK is down, the appliance attempts to find a perfect match for the second preferred location, Europe.UK. It finds that the GSLB service at Europe.UK is a perfect match. If the GSLB service is up, it sends the client the IP address of the service.
3. If the GSLB service at Europe.UK is down, it returns to the preferred location that has the highest priority, Asia.HK, and looks for partial matches. For Asia.HK, it finds that Asia.IN and Asia.JPN are partial matches. If only one of the corresponding GSLB services is up, it sends the client the IP address of the service. If both locations are up, it load balances the two services.
4. If all partial matches for Asia.HK are down, the appliance looks for partial matches for Europe.UK. It finds that Europe.RU and Europe.EG are partial matches for the preferred location. If only one of the corresponding GSLB services is up, it sends the client the IP address of the service. If both locations are up, it load balances the two services.
5. If all partial matches for Europe.UK are down, the appliance load balances all other available GSLB services. In the current example, the appliance load balances Africa.SD and Africa.ZMB because the remaining six GSLB services have been found to be down.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring GSLB Services

May 22, 2015

When you bind a remote service to a GSLB virtual server, the GSLB sites exchange metric information, including network metric Information, which is the round-trip-time and persistence Information.

If a metric exchange connection is momentarily lost between any of the participating sites, the remote site is marked as DOWN and load balancing is performed on the remaining sites that are UP. When metric exchange for a site is DOWN, the remote services belonging to the site are marked DOWN as well.

The NetScaler appliance periodically evaluates the state of the remote GSLB services by using either MEP or monitors that are explicitly bound to the remote services. Binding explicit monitors to local services is not required, because the state of the local GSLB service is updated by default using the MEP. However, you can bind explicit monitors to a remote service. When monitors are explicitly bound, the state of the remote service is not controlled by the metric exchange.

By default, when you bind a monitor to a remote GSLB service, the NetScaler appliance uses the state of the service reported by the monitor. However, you can configure the NetScaler appliance to use monitors to evaluate services in the following situations:

- Always use monitors (default setting).
- Use monitors when MEP is DOWN.
- Use monitors when remote services and MEP are DOWN.

The second and third of the above settings enable the NetScaler to stop monitoring when MEP is UP. For example, in a hierarchical GSLB setup, a GSLB site provides the MEP information about its child sites to its parent site. Such an intermediate site may evaluate the state of the child site as DOWN because of network issues, though the actual state of the site is UP. In this case, you can bind monitors to the services of the parent site and disable MEP to determine the actual state of the remote service. This option enables you to control the manner in which the states of the remote services are determined.

To use monitors, first create them, and then bind them to GSLB services.

This document includes the following information:

- Adding or Removing Monitors
- Binding Monitors to a GSLB Service

Adding or Removing Monitors

Updated: 2014-11-24

To add a monitor, you specify the type and the port. You cannot remove a monitor that is bound to a service. You must first unbind the monitor from the service.

To add a monitor by using the command line interface

At the command prompt, type the following commands to create a monitor and verify the configuration:

- add lb monitor <monitorName> -type <monitorType> -destPort <portNumber>
- show lb monitor <monitorName>

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add lb monitor monitor-HTTP-1 -type HTTP -destPort 80
show lb monitor monitor-HTTP-1
```

To remove a monitor by using the command line interface

At the command prompt, type:

```
rm lb monitor <monitorName>
```

To add a monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and add or delete a monitor.

Binding Monitors to a GSLB Service

Updated: 2014-11-24

Once you create monitors, you must bind them to GSLB services. When binding monitors to the services, you can specify a weight for the monitor. After binding one or more weighted monitors, you can configure a monitor threshold for the service. This threshold takes the service down if the sum of the bound monitor weights falls below the threshold value.

Note: In the configuration utility, you can set both the weight and the monitoring threshold at the same time that you bind the monitor. When using the command line, you must issue a separate command to set the service's monitoring threshold.

To bind the monitor to the GSLB service by using the command line interface

At the command prompt, type:

```
bind monitor <name> <serviceName> [ -state (Enabled | Disabled) ] -weight <positiveInteger>
```

Example

```
bind monitor monitor-HTTP-1 service-GSLB-1 -state enabled -weight 2
```

To set the monitoring threshold for a GSLB service by using the command line interface

At the command prompt, type:

```
set gslb service <ServiceName> -monThreshold <PositiveInteger>
```

Example

```
set gslb service service-GSLB-1 -monThreshold 9
```

To bind the monitor to the GSLB service by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services.
2. Click the **Monitor** section and bind the monitor to the GSLB service.

To set the monitoring threshold for a GSLB service by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services.
2. Click the **Monitor Threshold** section and enter a threshold value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring GSLB Sites

May 19, 2016

The NetScaler appliance uses MEP or monitors to determine the state of the GSLB sites. You can configure a GSLB site to always use monitors (the default), use monitors when MEP is down, or use monitors when both the remote service and MEP are down. In the latter two cases, the NetScaler appliance stops monitoring when MEP returns to the UP state.

To configure monitor triggering by using the command line interface

At the command prompt, type:

set gslb site <siteName> –triggerMonitor (ALWAYS | MEPDOWN | MEPDOWN_SVCDOWN)

Example

```
> set gslb site Site-GSLB-North-America –triggerMonitor Always
 Done
```

To configure monitor triggering by using the configuration utility

1. Navigate to Traffic Management > GSLB > Sites and double-click the site.
2. In the **Trigger Monitors** drop-down list, select an option for when to trigger monitoring.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Protecting the GSLB Setup Against Failure

May 22, 2015

You can protect your GSLB setup against failure of a GSLB site or a GSLB virtual server by configuring a backup GSLB virtual server, configuring the NetScaler appliance to respond with multiple IP addresses, or configuring a Backup IP address for a GSLB domain. You can also divert excess traffic to a backup virtual server by using spillover.

This document includes the following information:

- Configuring a Backup GSLB Virtual Server
- Configuring a GSLB Setup to Respond with Multiple IP Addresses
- Configuring a GSLB Virtual Server to Respond with an Empty Address Record When DOWN
- Configuring a Backup IP Address for a GSLB Domain
- Diverting Excess Traffic to a Backup Virtual Server

Configuring a Backup GSLB Virtual Server

Updated: 2015-05-04

Configuring a backup entity for a GSLB virtual server ensures that DNS traffic to a site is not interrupted if the GSLB virtual server goes down. The backup entity can be another GSLB virtual server, or it can be a backup IP address. With a backup entity configured, if the primary GSLB virtual server goes down, the backup entity handles DNS requests. To specify what should happen when the primary GSLB virtual server comes back up again, you can configure the backup entity to continue handling traffic until you manually enable the primary virtual server to take over (using the disablePrimaryOnDown option), or you can configure a timeout period after which the primary takes over.

Note: You can configure a single backup entity as a backup for multiple GSLB virtual server.
If you configure both the timeout and the disablePrimaryOnDown option for the backup entity, the backup session time-out takes precedence over the disablePrimaryOnDown setting.

To configure a backup GSLB virtual server by using the command line interface

At the command prompt, type the following commands to configure a GSLB virtual server as a backup virtual server and verify the configuration:

- set gslb vserver <name> -backupVServer <name> [-backupSessionTimeout <timeoutValue>] [-disablePrimaryOnDown (ENABLED | DISABLED)]
- show gslb vserver <name>

Example

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2 -backupSessionTimeout 3 -disablePrimaryOnDown ENABLED
show gslb vserver vserver-GSLB-1
```

To set GSLB virtual server as a backup virtual server by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers, and double-click the GSLB virtual server.
2. Click the **Backup Virtual Server** section and select the backup virtual server.

Configuring a GSLB Setup to Respond with Multiple IP Addresses

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2014-11-24

A typical DNS response contains the IP address of the best performing GSLB service. However, if you enable multiple IP response (MIR), the NetScaler appliance sends the best GSLB service as the first record in the response and adds the remaining active services as additional records. If MIR is disabled (the default), the NetScaler appliance sends the best service as the only record in the response.

To configure a GSLB virtual server for multiple IP responses by using the command line interface

At the command prompt, type the following commands to configure a GSLB virtual server for multiple IP responses and verify the configuration:

- set gslb vserver<name> -MIR (ENABLED | DISABLED)
- show gslb vserver <name>

Example

```
set gslb vserver vserver-GSLB-1 -MIR ENABLED
 show gslb vserver <vserverName>
```

To set a GSLB virtual server for multiple IP responses by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the GSLB virtual server for which you want to configure a backup virtual server (for example, vserver-GSLB-1).
2. On the Advanced tab, under When this VServer is “UP,” select the Send all “active” service IP in response (MIR) check box, and click OK.

Configuring a GSLB Virtual Server to Respond with an Empty Address Record When DOWN

Updated: 2014-11-24

A DNS response can contain either the IP address of the requested domain or an answer stating that the IP address for the domain is not known by the DNS server, in which case the query is forwarded to another name server. These are the only possible responses to a DNS query.

When a GSLB virtual server is disabled or in a DOWN state, the response to a DNS query for the GSLB domain bound to that virtual server contains the IP addresses of all the services bound to the virtual server. However, you can configure the GSLB virtual server to in this case send an empty down response (EDR). When this option is set, a DNS response from a GSLB virtual server that is in a DOWN state does not contain IP address records, but the response code is successful. This prevents clients from attempting to connect to GSLB sites that are down.

Note: You must configure this setting for each virtual server to which you want it to apply.

To configure a GSLB virtual server for empty down responses by using the command line interface

At the command prompt, type:

set gslb vserver<name> -EDR (ENABLED | DISABLED)

Example

```
> set gslb vserver vserver-GSLB-1 -EDR ENABLED
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To set a GSLB virtual server for empty down responses by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the GSLB virtual server for which you want to configure a backup virtual server (for example, vserver-GSLB-1).
2. On the Advanced tab, under When this VServer is “Down,” select the Do not send any service’s IP address in response (EDR) check box.
3. Click OK.

Configuring a Backup IP Address for a GSLB Domain

Updated: 2014-11-24

You can configure a backup site for your GSLB configuration. With this configuration in place, if all of the primary sites go DOWN, the IP address of the backup site is provided in the DNS response.

Typically, if a GSLB virtual server is active, that virtual server sends a DNS response with one of the active site IP addresses as selected by the configured GSLB method. If all the configured primary sites in the GSLB virtual server are inactive (in the DOWN state), the authoritative domain name system (ADNS) server or DNS server sends a DNS response with the backup site’s IP address.

Note: When a backup IP address is sent, persistence is not honored.

To set a backup IP address for a domain by using the command line interface

At the command prompt, type the following commands to set a backup IP address and verify the configuration:

- set gslb vserver <name> -domainName <string> -backupIP <IPAddress>
- show gslb vserver <name>

Example

```
set gslb vserver vserver-GSLB-1 -domainName www.abc.com -backupIP 10.102.29.66
show gslb vserver vserver-GSLB-1
```

To set a backup IP address for a domain by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the GSLB virtual server to which you want to bind the backup domain (for example, vserver-GSLB-1).
2. Click the **Domains** section, configure the GSLB domain and specify the IP address of the backup domain in the **Backup IP** field.

Diverting Excess Traffic to a Backup Virtual Server

Updated: 2014-11-24

Once the number of connections to a primary GSLB virtual server exceeds the configured threshold value, you can use the spillover option to divert new connections to a backup GSLB virtual server. This threshold value can be calculated dynamically or set manually. Once the number of connections to the primary virtual server drops below the threshold, the primary GSLB virtual server resumes serving client requests.

You can configure persistence with spillover. When persistence is configured, new clients are diverted to the backup virtual server if that client is not already connected to a primary virtual server. When persistence is configured, connections that were diverted to the backup virtual server are not moved back to the primary virtual server after the number of connections to the primary virtual server drops below the threshold. Instead, the backup virtual server continues to process those connections until they are terminated by the user. Meanwhile, the primary virtual server accepts new clients.

The threshold can be measured either by the number of connections or by the bandwidth.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If the backup virtual server reaches the configured threshold and is unable to take any additional load, the primary virtual server diverts all requests to the designated redirect URL. If a redirect URL is not configured on the primary virtual server, subsequent requests are dropped.

The spillover feature prevents the remote backup GSLB service (backup GSLB site) from getting flooded with client requests when the primary GSLB virtual server fails. This occurs when a monitor is bound to a remote GSLB service, and the service experiences a failure that causes its state to go DOWN. The monitor continues to keep the state of the remote GSLB service UP, however, because of the spillover feature.

As part of the resolution to this problem, two states are maintained for a GSLB service, the primary state and effective state. The primary state is the state of the primary virtual server and the effective state is the cumulative state of the virtual servers (primary and backup chain). The effective state is set to UP if any of the virtual servers in the chain of virtual servers is UP. A flag that indicates that the primary VIP has reached the threshold is also provided. The threshold can be measured by either the number of connections or the bandwidth.

A service is considered for GSLB only if its primary state is UP. Traffic is directed to the backup GSLB service only when all the primary virtual servers are DOWN. Typically, such deployments will have only one backup GSLB service.

Adding primary and effective states to a GSLB service has the following effects:

- When source IP persistence is configured, the local DNS is directed to the previously selected site only if the primary virtual server on the selected site is UP and below threshold. Persistence can be ignored in the round robin mode.
- If cookie-based persistence is configured, client requests are redirected only when the primary virtual server on the selected site is UP.
- If the primary virtual server has reached its saturation and the backup VIP(s) is absent or down, the effective state is set to DOWN.
- If external monitors are bound to an HTTP-HTTPS virtual server, the monitor decides the primary state.
- If there is no backup virtual server to the primary virtual server and the primary virtual server has reached its threshold, the effective state is set to DOWN.

To configure a backup GSLB virtual server by using the command line interface

At the command prompt, type the following commands to configure a backup GSLB virtual server and verify the configuration:

- set gslb vserver <name> -soMethod <method> -soThreshold <threshold> -soPersistence (**ENABLED** | **DISABLED**) -soPersistenceTimeout <timeout>
- show gslb vserver <name>

Example

```
set gslb vserver Vserver-GSLB-1 -soMethod CONNECTION -soThreshold 1000 -soPersistence ENABLED -soPersistenceTimeout 2
show gslb vserver Vserver-GSLB-1
```

To configure a backup GSLB virtual server by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server that you want to configure as a backup (for example, Vserver-LB-1).
2. Click the **SpillOver** section and set the following parameters:
   - Method— soMethod
   - Threshold— soThreshold
   - Persistence Time-out (min) — soPersistenceTimeout
3. Select the Persistence option and click **OK**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Client Connections

Aug 07, 2017

To facilitate management of client connections, you can enable delayed cleanup of connections to the virtual server. You can then manage local DNS traffic by configuring DNS policies.

This document includes the following information:

- Enabling Delayed Cleanup of Virtual Server Connections
- Managing Local DNS Traffic by Using DNS Policies
- Adding DNS Views

Enabling Delayed Cleanup of Virtual Server Connections

Updated: 2014-11-24

The state of a virtual server depends on the states of the services bound to it, and the state of each service depends on the monitors bound to it. If a server is slow or down, the monitoring probes time out and the service that represents the server is marked as DOWN. A virtual server is marked as DOWN only when all services bound to it are marked as DOWN. You can configure services and virtual servers to either terminate all connections when they go down, or allow the connections to go through. The latter setting is for situations in which a service is marked as DOWN because of a slow server.

When you configure the down state flush option, the NetScaler appliance performs a delayed cleanup of connections to a GSLB service that is down.

To enable delayed cleanup of virtual server connections by using the command line interface

At the command prompt, type the following commands to configure delayed connection cleanup and verify the configuration:

- set gslb service <name> -downStateFlush (ENABLED | DISABLED)
- show gslb service <name>

Example

```
> set gslb service Service-GSLB-1 -downStateFlush ENABLED
 Done
> show gslb service Service-GSLB-1
 Done
```

To enable delayed cleanup of virtual server connections by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services and double-click the service.
2. Click the **Other Settings** section and select the **Down State Flush** option.

Managing Local DNS Traffic by Using DNS Policies

Updated: 2015-05-22

You can use DNS policies to implement site affinity by directing traffic from the IP address of a local DNS resolver or network to a predefined target GSLB site. This is configured by creating DNS policies with DNS expressions and binding the policies globally on the NetScaler appliance.

DNS Expressions

Updated: 2013-07-18

The NetScaler appliance provides certain predefined DNS expressions that can be used for configuring actions specific to a domain. Such actions can, for example, drop certain requests, select a specific view for a specific domain, or redirect certain requests to a specific location.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

These DNS expressions (also called *rules*) are combined to create DNS policies that are then bound globally on the NetScaler appliance.

Following is the list of predefined DNS qualifiers available on the NetScaler appliance:

- CLIENT.UDP.DNS.DOMAIN.EQ("domainname")
- CLIENT.UDP.DNS.IS_AREC
- CLIENT.UDP.DNS.IS_AAAAREC
- CLIENT.UDP.DNS.IS_SRVREC
- CLIENT.UDP.DNS.IS_MXREC
- CLIENT.UDP.DNS.IS_SOAREC
- CLIENT.UDP.DNS.IS_PTRREC
- CLIENT.UDP.DNS.IS_CNAME
- CLIENT.UDP.DNS.IS_NSREC
- CLIENT.UDP.DNS.IS_ANYREC

The CLIENT.UDP.DNS.DOMAIN DNS expression can be used with string expressions. If you are using domain names as part of the expression, they must end with a period (.). For example, CLIENT.UDP.DNS.DOMAIN.ENDSWITH("abc.com.")

To create an expression by using the configuration utility

1. Click the icon next to the Expression text box. Click Add. (Leave the Flow Type and Protocol drop-down list boxes empty.) Follow these steps to create a rule.
2. In the Qualifier box, select a qualifier (for example, LOCATION).
3. In the Operator box, select an operator (for example, ==).
4. In the Value box, type a value (for example, Asia, Japan....).
5. Click OK. Click Create and click Close. The rule is created.
6. Click OK.

Configuring DNS Actions

Updated: 2014-11-24

A DNS policy includes the name of a DNS action to be performed when the policy rule evaluates to TRUE. A DNS action can do one of the following:

- Send the client an IP address for which you have configured a DNS view. For more information about DNS views, see Adding DNS Views.
- Send the client the IP address of a GSLB service after referring to a list of preferred locations that overrides static proximity behavior. For more information about preferred locations, see Overriding Static Proximity Behavior by Configuring Preferred Locations.
- Send the client a specific IP address as determined by the evaluation of the DNS query or response (DNS response rewrite).
- Forward a request to the name server without performing a lookup in the appliance's DNS cache.
- Drop a request.

You cannot create a DNS action for dropping a DNS request or for bypassing the DNS cache on the appliance. If you want to drop a DNS request, use the built-in action, dns_default_act_Drop. If you want to bypass the DNS cache, use the built-in action, dns_default_act_Cachebypass. Both actions are available along with custom actions in the Create DNS Policy and the Configure DNS Policy dialog boxes. These built-in actions cannot be modified or removed.

To configure a DNS action by using the command line interface

At the command prompt, type the following commands to configure a DNS action and verify the configuration:

- add dns action <actionName> <actionType> (-IPAddress <ip_addr | ipv6_addr> ... | -viewName <string> | -preferredLocList <string> ...) [-TTL <secs>]
- show dns action [<actionName>]

Examples

Example 1: Configuring DNS Response Rewrite. The following DNS action sends the client a preconfigured IP address when the policy to which the action is bound evaluates to true:

```
> add dns action dns_act_response_rewrite Rewrite_Response -IPAddress 192.0.2.20 192.0.2.56 198.51.100.10
 Done
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> show dns action dns_act_response_rewrite
1) ActionName: dns_act_response_rewrite ActionType: Rewrite_Response TTL: 3600 IPAddress: 192.0.2.20 192.0.2.56 198.51.100.10
 Done
```

Example 2: Configuring a DNS-View Based Response. The following DNS action sends the client an IP address for which you have configured a DNS view:

```
> add dns action send_ip_from_view_internal_ip ViewName -viewName view_internal_ip
 Done
> show dns action send_ip_from_view_internal_ip
1) ActionName: send_ip_from_view_internal_ip ActionType: ViewName ViewName: view_internal_ip
 Done
```

Example 3: Configuring a Response Based on a Preferred Location List. The following DNS action sends the client the IP address that corresponds to the preferred location that it selects from the specified list of locations:

```
> add dns action send_preferred_location GslbPrefLoc -preferredLocList NA.tx.ns1.*.*.* NA.tx.ns2.*.*.* NA.tx.ns3.*.*.*
 Done
> show dns action  send_preferred_location
1) ActionName: send_preferred_location ActionType: GslbPrefLoc PreferredLocList: "NA.tx.ns1.*.*.*" "NA.tx.ns2.*.*.*" "NA.tx.ns3.*.*.*"
 Done
```

To configure a DNS action by using the NetScaler configuration utility

1. Navigate to Traffic Management > DNS > Actions, create or edit a DNS action.
2. In the Create DNS Action or Configure DNS Action dialog box, set the following parameters:
   - Action Name (cannot be changed for an existing DNS action)
   - Type (cannot be changed for an existing DNS action)
     To set the Type parameter, do one of the following:
     - To create a DNS action that is associated with a DNS view, select View Name. Then, from the View Name list, select the DNS view that you want to use in the action.
     - To create a DNS action with a preferred location list, select Preferred Location List. In Preferred Location, enter a location, and then click Add. Add as many DNS locations as you want.
     - To configure a DNS action for rewriting a DNS response on the basis of policy evaluation, select Rewrite Response. In IP Address, enter an IP address, and then click Add. Add as many IP addresses as you want.
   - TTL (applicable only to the Rewrite Response action type)

Configuring DNS Policies

Updated: 2014-11-24

DNS policies operate on a location database that uses static and custom IP addresses. The attributes of the incoming local DNS request are defined as part of an expression, and the target site is defined as part of a DNS policy. While defining actions and expressions, you can use a pair of single quotation marks (‘’) as a wildcard qualifier to specify more than one location. When a DNS policy is configured and a GSLB request is received, the custom IP address database is first queried for an entry that defines the location attributes for the source:

- When a DNS query comes from an LDNS, the characteristics of the LDNS are evaluated against the configured policies. If they match, an appropriate action (site affinity) is executed. If the LDNS characteristics match more than one site, the request is load balanced between the sites that match the LDNS characteristics.
- If the entry is not found in the custom database, the static IP address database is queried for an entry, and if there is a match, the above policy evaluation is repeated.
- If the entry is not found in either the custom or static databases, the best site is selected and sent in the DNS response on the basis of the configured load balancing method.

The following restrictions apply to DNS policies created on the NetScaler appliance.

- A maximum of 64 policies are supported.

- DNS policies are global to the NetScaler and cannot be applied to a specific virtual server or domain.
- Domain or virtual server specific binding of policy is not supported.

You can use DNS policies to direct clients that match a certain IP address range to a specific site. For example, if you have a GSLB setup with multiple GSLB sites that are separated geographically, you can direct all clients whose IP address is within a specific range to a particular data

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

center.

Both TCP-based and UDP-based DNS traffic can be evaluated. Policy expressions are available for UDP-based DNS traffic on the server and for both UDP-based DNS traffic and TCP-based DNS traffic on the client side. Additionally, you can configure expressions to evaluate queries and responses that involve only the following DNS question types (or QTYPE values):

- A
- AAAA
- NS
- SRV
- PTR
- CNAME
- SOA
- MX
- ANY

The following response codes (RCODE values) are also supported:

- NOERROR - No error
- FORMERR - Format error
- SERVFAIL - Server failure
- NXDOMAIN - Non-existent domain
- NOTIMP - Query type not implemented
- REFUSED - Query refused

You can configure expressions to evaluate DNS traffic. A DNS expression begins with the DNS.REQ or DNS.RES prefixes. Functions are available for evaluating the queried domain, the query type, and the carrier protocol. For more information about DNS expressions, see "Expressions for Evaluating a DNS Message and Identifying Its Carrier Protocol" in "Policy Configuration and Reference".

To add a DNS policy by using the command line interface

At the command prompt, type the following commands to create a DNS policy and verify the configuration:

- add dns policy <name> <rule> <actionName>
- show dns policy <name>

Example

```
> add dns policy policy-GSLB-1 'CLIENT.UDP.DNS.DOMAIN.EQ(\"domainname\")' my_dns_action
 Done
> show dns policy policy-GSLB-1
 Name: policy-GSLB-1
 Rule: CLIENT.UDP.DNS.DOMAIN.EQ("domainname")
 Action Name: my_dns_action
 Hits: 0
 Undef Hits: 0

 Done
```

To remove a configured DNS policy by using the command line interface

At the command prompt, type:

rm dns policy <name>

To configure a DNS policy by using the NetScaler configuration utility

1. Navigate to Traffic Management > DNS > Policies and create a DNS policy.
2. In the Create DNS Policy or Configure DNS Policy dialog box, set the following parameters:
   - Policy Name (cannot be changed for an existing policy)
   - Action
   - Expression
     To specify an expression, do the following:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Click Add, and then, in the drop-down box that appears, select the expression element with which you want to begin the expression. A second list appears. The list contains a set of expression elements that you can use immediately after the firs expression element.
2. In the second list, select the expression element that you want, and then enter a period.
3. After each selection, if you enter a period, the next set of valid expression elements appear in a list. Select expression elements and fill in arguments to functions until you have the expression you want.

3. Click Create or OK, and then click Close.

Binding DNS Policies

Updated: 2013-08-29

DNS policies are bound globally on the NetScaler appliance and are available for all configured GSLB virtual servers. Even though DNS policies are globally bound, policy execution can be limited to a specific GSLB virtual server by specifying the domain in the expression.

Note: Even though the bind dns global command accepts REQ_OVERRIDE and RES_OVERRIDE as valid bind points, those bind points are redundant, because DNS policies can be bound only globally. Bind your DNS policies only to the REQ_DEFAULT and RES_DEFAULT bind points.

To bind a DNS policy globally by using the command line interface

At the command prompt, type the following commands to bind a DNS policy globally and verify the configuration:

- bind dns global <policyName> <priority> [-gotoPriorityExpression <string>] [-type <type>]
- show dns global -type <type>

Example

```
> bind dns global policy-GSLB-1 10 -gotoPriorityExpression END
 Done
> show dns global -type REQ_DEFAULT
1) Policy Name: policy-GSLB-1
 Priority: 10
 GotoPriorityExpression: END

 Done
```

To bind a DNS policy globally by using the configuration utility

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings.
3. In the Bind/Unbind DNS Policy(s) to Global dialog box, click Insert Policy.
4. In the Policy Name column, select, from the list, the policy that you want to bind. Alternatively, in the list, click New Policy, and then create a DNS policy by setting parameters in the Create DNS Policy dialog box.
5. To modify a policy that is already bound globally, click the name of the policy, and then click Modify Policy. Then, in the Configure DNS Policy dialog box, modify the policy, and then click OK.
6. To unbind a policy, click the name of the policy, and then click Unbind Policy.
7. To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
8. To regenerate assigned priorities, click Regenerate Priorities. The priority values are modified to begin at 100, with increments of 10, without affecting the order of evaluation.
9. Click OK.

To view the global bindings of a DNS policy by using the command line interface

At the command prompt, type:

show dns global

To view the global bindings of a DNS policy by using the configuration utility

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings. The global bindings of all DNS policies appear in this dialog box.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding DNS Views

You can configure DNS views to identify various types of clients and provide an appropriate IP address to a group of clients who query for the same GSLB domain. DNS views are configured by using DNS policies that select the IP addresses sent back to the client.

For example, if you have configured GSLB for your company's domain and have the server hosted in your company's network, clients querying for the domain from within your company's internal network can be provided with the server's internal IP address instead of the public IP address. Clients that query DNS for the domain from the Internet, on the other hand, can be provided the domain's public IP address.

To add a DNS view, you assign it a name of up to 31 characters. The leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ). After adding the view, you configure a policy to associate it with clients and a part of the network, and you bind the policy globally. To configure and bind a DNS policy, see Managing Local DNS Traffic by Using DNS Policies.

To add a DNS view by using the command line interface

At the command prompt, type the following commands to create a DNS view and verify the configuration:

- add dns view <viewName>
- show dns view <viewName>

Example

```
add dns view PrivateSubnet
show dns view PrivateSubnet
```

To remove a DNS view by using the command line interface

At the command prompt, type:

```
rm dns view <viewName>
```

To add a DNS view by using the configuration utility

Navigate to **Traffic Management** > **DNS** > **Views** and add a DNS view.

For details on how to create a DNS policy and how to bind DNS policies globally, see Managing Local DNS Traffic by Using DNS Policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring GSLB for Disaster Recovery

Aug 06, 2017

Disaster recovery capability is critical, because downtime is costly. A NetScaler appliance configured for GSLB forwards traffic to the least-loaded or the best-performing data center. This configuration, referred to as an active-active setup, not only improves performance, but also provides immediate disaster recovery by routing traffic to other data centers if a data center that is part of the setup goes down. Alternatively, you can configure an active-standby GSLB setup for disaster recovery only.

This document includes the following information:

- Configuring GSLB for Disaster Recovery in an Active-Standby Data Center Setup
- Configuring for Disaster Recovery in an Active-Active Data Center Setup
- Configuring for Disaster Recovery with Weighted Round Robin
- Configuring for Disaster Recovery with Data Center Persistence

Configuring GSLB for Disaster Recovery in an Active-Standby Data Center Setup

Updated: 2014-11-24

A conventional disaster recovery setup includes an active data center and a standby data center. The standby data center is a remote site. When a failover occurs as a result of a disaster event that causes the primary active data center to be inactive, the standby data center becomes operational.
Configuring disaster recovery in an active-standby data-center setup consists of the following tasks.

- Create the active data center.
  - Add a local GSLB site.
  - Add a GSLB vserver, which represents the active data center.
  - Bind the domain to the GSLB virtual server.
  - Add gslb services and bind the services to active GSLB virtual server.
- Create the standby data center.
  - Add a remote gslb site.
  - Add a gslb vserver, which represents standby data center.
  - Add gslb services which represents standby data center and bind the services to the standby gslb vserver.
  - Designate the standby data center by configuring the standby GSLB virtual server as the backup virtual server for the active GSLB virtual server.

Once you have configured the primary data center, replicate the configuration for the backup data center and designate it as the standby GSLB site by designating a GSLB virtual server at that site as the backup virtual server.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

To designate the standby GSLB site by using the command line interface

At both the active site and the remote site, at the command prompt, type:

set gslb vserver <name> -backupVserver <string>

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2
```

To configure the standby site by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the GSLB virtual server for the primary site.
2. Click the **Backup Virtual Server** section and select a backup virtual server.

By default, once the primary virtual server becomes active, it starts receiving traffic. However, if you want the traffic to be directed to the backup virtual server even after the primary virtual server becomes active, use the 'disable primary on down' option.

Configuring for Disaster Recovery in an Active-Active Data Center Setup

An active-active GSLB deployment, in which both GSLB sites are active, removes any risk that may arise in having a standby data center. With such a setup, web or application content can be mirrored in geographically separate locations. This ensures that data is consistently available at each distributed data center.

To configure GSLB for disaster recovery in an active-active data center set up, you must first configure the basic GSLB setup on the first data center and then configure all other data centers.

First create at least two GSLB sites. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server in the local site. Finally, at the local site, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

Once you have configured the first data center, replicate the configuration for other data centers part of the setup.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Configuring for Disaster Recovery with Weighted Round Robin

Updated: 2014-11-24

When you configure GSLB to use the weighted round robin method, weights are added to the GSLB services and the configured percentage of incoming traffic is sent to each GSLB site. For example, you can configure your GSLB setup to forward 80 percent of the traffic to one site and 20 percent of the traffic to another. After you do this, the NetScaler appliance will send four requests to the first site for each request that it sends to the second.

To set up the weighted round robin method, first create two GSLB sites, local and remote. Next, for the local site create a GSLB virtual server and GSLB services, and bind the services to the virtual server. Configure the GSLB method as round robin. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

Each service that represents a physical server in the network has weights associated with it. Therefore the GSLB service is assigned a dynamic weight that is the sum of weights of all services bound to it. Traffic is then split between the GSLB services based on the ratio of the dynamic weight of the particular service to the total weight. You can also configure individual weights for each GSLB service instead of the dynamic weight.

If the services do not have weights associated with them, you can configure the GSLB virtual server to use the number of services bound to it to calculate the weight dynamically.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Once you configure a basic GSLB setup, you must configure the weighted round robin method such that the traffic is split between the configured GSLB sites according to the weights configured for the individual services.

To configure a virtual server to assign weights to services by using the command line interface

At the command prompt, type one of the following commands, depending upon whether you want to create a new load balancing virtual server or configure an existing one:

- add lb vserver <name>@ -weight <WeightValue> <ServiceName>
- set lb vserver <name>@ -weight <WeightValue> <ServiceName>

Example

```
add lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
set lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
```

To set dynamic weight by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -dynamicWeight DynamicWeightType
```

Example

```
set gslb vserver Vserver-GSLB-1 -dynamicWeight ServiceWeight
```

To add weights to the GSLB services by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -serviceName GSLBServiceName -weight WeightValue
```

Example

```
set gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1 -weight 1
```

To configure a virtual server to assign weights to services by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** and double-click the virtual server (for example, Vserver-LB-1).
2. Click the **Services** section and set the weight of a service.

To add weights to the GSLB services by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the virtual server (for example, vserver-GSLB-1)
2. Click the **Services** section and set the weight of the service in the **Weight** field.

To set dynamic weight by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the virtual server (for example, vserver-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GSLB-1).

2. Click the **Method** section and, from the **Dynamic Weight** drop-down list select **SERVICEWEIGHT**.

Configuring for Disaster Recovery with Data Center Persistence

Data center persistence is required for web applications that require maintaining a connection with the same server instead of having the requests load balanced. For example, in an e-commerce portal, maintaining a connection between the client and the same server is critical. For such applications, HTTP redirect persistence can be configured in an active-active setup.

To configure GSLB for disaster recovery with data center persistence, you must first configure the basic GSLB set up and then configure HTTP redirect persistence.

First create two GSLB sites, local and remote. Next, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Next, create a load balancing virtual server with the same virtual server IP address as the GSLB service. Finally, duplicate the previous steps for the remote configuration, or configure the NetScaler appliance to autosynchronize your GSLB configuration.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure HTTP redirect precedence to enable data center persistence.

To configure HTTP redirect by using the command line interface

At the command prompt, type the following commands to configure HTTP redirect and verify the configuration:

- set gslb service <serviceName> -sitePersistence <sitePersistence> -sitePrefix <string>
- show gslb service <serviceName>

Example

```
set gslb service Service-GSLB-1 -sitePersistence HTTPRedirect -sitePrefix vserver-GSLB-1
show gslb service Service-GSLB-1
```

To configure HTTP redirect by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services** and double-click the GSLB service to be configured.
2. Click the **Site Persistence** section, select the **HTTPRedirect** option, and in the **Site Prefix** text box, enter the site prefix (for example, vserver-GSLB-1).

> ### Note
> When site persistence is not configured and if a load balancing virtual server that is configured as a local GSLB service is DOWN, the HTTP requests are redirected to other healthy GSLB sites using a 302 redirect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring GSLB for Proximity

May 19, 2016

When you configure GSLB for proximity, client requests are forwarded to the closest data center. The main benefit of the proximity-based GSLB method is faster response times resulting from the selection of the closest available data center. Such a deployment is critical for applications that require fast access to large volumes of data.

You can configure GSLB for proximity based on the round trip time (RTT), static proximity, or a combination of the two.

Configuring Dynamic Method (RTT)

Dynamic round trip time (RTT) is a measure of time or delay in the network between the client's local DNS server and a data resource. To measure dynamic RTT, the NetScaler appliance probes the client's local DNS server and gathers RTT metric information. The NetScaler then uses this metric to make its load balancing decision. Global server load balancing monitors the real-time status of the network and dynamically directs the client request to the data center with the lowest RTT value

To configure GSLB for proximity with dynamic method, you must first configure the basic GSLB set up and then configure dynamic RTT.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure the dynamic RTT method.

For details on how to configure the GSLB virtual server to use the dynamic RTT method for load balancing, see Configuring Dynamic RTT.

Configuring Static Proximity

The static proximity method for GSLB uses an IP address-based static proximity database to determine the proximity between the client's local DNS server and the GSLB sites. The NetScaler appliance responds with the IP address of a site that best matches the proximity criteria.

If two or more GSLB sites at different geographic locations serve the same content, the NetScaler appliance maintains a database of IP address ranges and uses the database for decisions about the GSLB sites to which to direct incoming client requests.

To configure GSLB for proximity with static proximity, you must first configure the basic GSLB set up and then configure static proximity.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure static proximity.

For details on how to configure the GSLB virtual server to use static proximity for load balancing, see Configuring Static Proximity.

Configuring Static Proximity and Dynamic RTT

You can configure the GSLB virtual server to use a combination of static proximity and dynamic RTT when you have some clients coming from an internal network like a branch office. You can configure GSLB such that the clients coming from the branch office or any other internal network are directed to a particular GSLB site that is geographically close to the client network. For all other requests, you can use dynamic RTT.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure the GSLB virtual server to use static proximity for all traffic originating from an internal network and then use dynamic RTT for all other traffic.

For details on how to configure static proximity, see Configuring Static Proximity and for details on how to configure dynamic RTT, see Configuring Dynamic RTT.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Parent-Child Topology

Nov 06, 2016

NetScaler appliances configured for global server load balancing (GSLB) provide for disaster recovery and ensure continuous availability of applications by protecting against points of failure in a wide area network (WAN). GSLB can balance the load across data centers by directing client requests to the closest or best performing data center, or to surviving data centers in the event of an outage.

There are three fundamental entities that must be configured for GSLB:

- **Site**: A GSLB site represents a NetScaler or a high availability (HA) pair of NetScaler appliances that maintain GSLB state information and provide information about how the NetScaler nodes should communicate. A site can also represent a data center.
- **GSLB virtual server**: A GSLB virtual server represents a group of resources to which users can be directed, and the logic used to select one resource verses another.
- **GSLB service**: A GSLB service represents a target resource and is bound to a GSLB virtual server. The target resource might be a load balancing virtual server on a NetScaler, or it could represent a third party server.

Sites and services are inherently linked to indicate proximity between the two. That is, all services must belong to a site, and are assumed to be in the same location as the GSLB site for proximity purposes. Likewise, services and virtual servers are linked, so that the logic is linked to the resources that are available.

Relationships among GSLB Sites

The concept of sites is central to NetScaler GSLB implementations. Unless otherwise specified, sites form a peer relationship among themselves. This relationship is used first to exchange health information and then to distribute load as determined by the selected algorithm. In many situations, however, a peer relationship among all GSLB sites is not desirable. Reasons for not having an all-peer implementation could be

1. To clearly separate GSLB sites. For example, to separate sites that participate in resolving DNS queries from the traffic management sites.
2. To reduce the volume of Metric Exchange Protocol (MEP) traffic, which increases exponentially with an increasing number of peer sites.

These goals can be achieved by using parent and child GSLB sites. Parent-child relationships can be used to build a two-level hierarchical GSLB design with the following characteristics:

- At the top level are parent sites that have peer relationships with other parents.
- Each parent can have multiple children, but each child can have only one parent.
- Each parent site exchanges health information with its children and with other parent sites.
- A child communicates only with its parent.

Note

- In a parent-child relationship for GSLB, only the parent site responds to ADNS queries. The child sites act as normal load balancing sites.
- ADNS service or DNS load balancing virtual servers should be configured only in the parent site.
- A parent site can have a normal GSLB configuration, that is, services from local and all remote sites but a child site can have local services only.
- In a parent-child topology, GSLB services are not always required to be configured on a child site. However, if you have additional configuration such as client authentication, client IP address insertion, or other SSL-specific requirement, you must add an explicit GSLB service on the child site and configure it accordingly.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

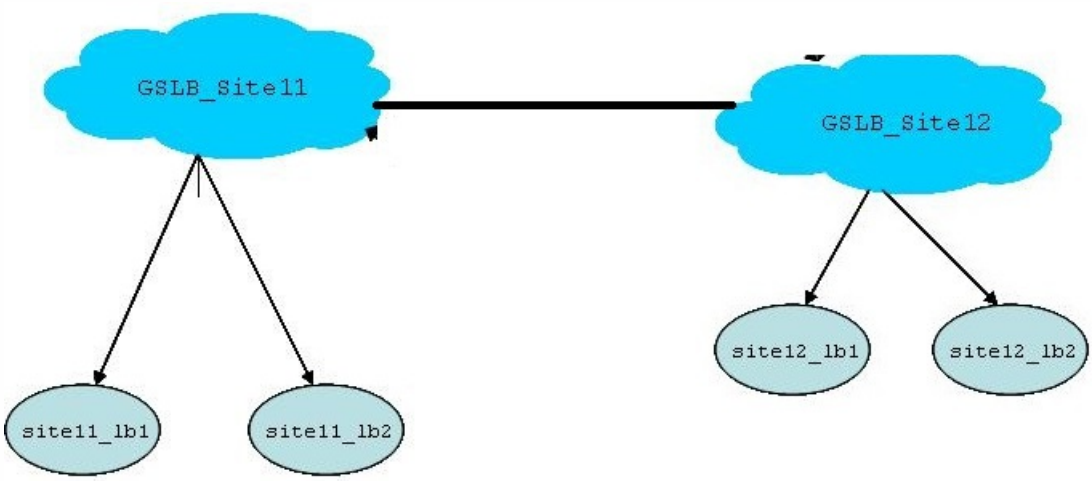
- In a parent-child topology, the parent site and the child site can be on different NetScaler software versions. However, to use the GSLB automaticConfigSync option to synchronize the configuration across the parent sites, all parent sites must be on the same NetScaler software versions. If you are not using the automaticConfigSync option, then the parent site and the child site can be on different NetScaler software versions but make sure that you are not using any of the new features in the latest release. This is also applicable, in general, to two NetScaler nodes participating in GSLB.

Limitations of GSLB Parent-Child site configuration:

- You can configure upto 32 Parent sites and a total of 1024 Child sites.
- On the Child site, by default, the nwmetricExchange and sessionExchange options are disabled.
- Round Trip Time (RTT) GSLB method is not recommended for GSLB Parent-Child site configuration.

Setting Up a Parent-Child Configuration for Global Server Load BalancingIf you have a firewall configured at a GSLB site, make sure that port 3011 is open. Follow the procedures at the following location to create services and virtual servers: Configuring Global Server Load Balancing (GSLB)

Figure 1. GSLB Parent-Child Topology



In the above figure:

- GSLB_Site11 and GSLB_Site12 are parent sites in a peer relationship.
- site11_lb1 and site11_lb2 are the child sites of GSLB_Site11, while site12_lb1 and site_lb2 are the child sites of GSLB_Site12.

The configuration of each parent site includes the information about all the child sites associated with it, but the configuration of each child site pertains only to that child and its parent. A child site is not aware about any other parent site or other child sites in the configuration. For example, in the above figure, the configuration of child site site11_lb1 would include only information about its parent site, GSLB_Site11.

Note: GSLB auto sync syncs only the GSLB configuration across the parent sites. It does not sync any configuration to the child sites.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To set up a parent-child configuration for GSLB by using the NetScaler command line

1. On each parent site, enter the following command: add gslb site<siteName><siteIPAddress> [-publicIP<ip_addr|ipv6_addr|*>] [-parentSite<string>] For example: # add gslb site gslb_site11 1.1.1.1 -publicIP 1.1.1.1 # add gslb site site11_lb1 1.1.1.2 -publicIP 1.1.1.2 -parentSite gslb_site11

   # add gslb site site11_lb2 1.1.1.3 -publicIP 1.1.1.3 -parentSite gslb_site11

   # add gslb site gslb_site12 3.3.3.1 -publicIP 3.3.3.1

   # add gslb site site12_lb1 3.3.3.2 -publicIP 3.3.3.2 -parentSite gslb_site12

   # add gslb site site12_lb2 3.3.3.3 -publicIP 3.3.3.3 -parentSite gslb_site12

   The above command makes the parent site aware of its child sites as well as of the other parent site in the configuration.

2. On each child site, enter the following command: add gslb site<siteName><siteIPAddress> [-publicIP<ip_addr|ipv6_addr|*>] [-parentSite<string>] For example: # add gslb site gslb_site11 1.1.1.1 -publicIP 1.1.1.1 # add gslb site site11_lb1 1 1.1.1.2 -publicIP 1.1.1.2 -parentSite gslb_site11

   The above command creates the child site and adds the parent-site information to child site's configuration.

   In the child sites, only the child site and its respective parent site needs to configured and not other parent or child sites.

   Network metrics, such as RTT and persistence session information, are synced only across the parent sites. Therefore, parameters like nwMetric and sessionExchange are disabled by default on all the child sites.

To verify correct parent-child configuration, check the states of all the GSLB services bound to the parent sites.

Note

If the load balancing virtual server IP address is a private IP address and the public IP address is different from this IP address, you need to configure a GSLB service for the local load balancing virtual server on the child site. This is required for statistics collection between the parent and the child site.

On the child site, at the command prompt, type:

add gslb service <name> <private IP/lb vserver IP> http 80 –sitename <childsite name> -publicip <public IP of LB vserver>

Example:

add gslb service Service-GSLB 192.168.1.3 http 80 -GSLB_Site11 site 11_lb1 172.16.1.1

Where 192.168.1.3 is a private IP address of the load balancing virtual server and 172.16.1.1 is a public IP address of the load balancing virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring GSLB Service Selection Using Content Switching

May 19, 2016

In a typical GSLB deployment, you can prioritize the selection of a set of GSLB services bound to a GSLB virtual server, but you cannot do the following:

- Restrict the selection of a GSLB service from a subset of GSLB services bound to a GSLB virtual server for the given domain.
- Apply different load balancing methods on the different subsets of GSLB services in the deployment.
- Apply spillover policies on a subset of GSLB services, and you cannot have a backup for a subset of GSLB services.
- Configure a subset of GSLB services to serve different content. That is, you cannot content switch between servers in different GSLB sites. The GSLB configuration assumes that the servers contain the same content.
- Define a subset GSLB services with different priorities and specify an order in which the services in the subset are applied to a request.

You can now configure a content switching (CS) policy to customize the GSLB deployment. First configure a set of GSLB services and bind it to a GSLB virtual server. Then, configure a CS virtual server of target type GSLB, define a CS policy and action with the GSLB virtual server as target virtual server, and bind the CS policy to CS virtual server.

> **Important**
>
> - Only CS policies with DNS based expressions can be bound to a CS virtual server of target type GSLB.
> - If a GLSB service is bound to a CS virtual server through a GSLB virtual server, you cannot bind another GSLB virtual server bound with the same GSLB service to the CS virtual server.

Example

Consider a GLSB deployment that includes two GSLB sites. At each site, four GSLB services (S-1, S-2, S-3, and S-4) are bound to GSLB virtual server VS-1. You can configure a content switching (CS) virtual server of target type GSLB and define a CS policy and action with VS-1 as the target virtual server, so that requests for content in English are served only by S-1 and S-2, and requests for content in the local language are served only by S-3 and S-4.

You could give S-1 priority by configuring a backup virtual server to VS-1 and binding S-2 to the backup virtual server. Client requests would then be served by S-1 unless the server it represents went down, in which case the requests would be served by S-2. If both S-1 and S-2 were down, clients would receive an empty response.

To configure GSLB Service Selection using Content Switching:

1. Configure GSLB. For instructions, see Configuring Global Server Load Balancing.
2. Configure a Content Switching (CS) virtual server of target type GSLB. For more information, see Creating Content Switching Virtual Servers.
3. Configure Content Switching (CS) policies. For more information, see Configuring Content Switching Policies.
4. Configure CS actions that designate a GSLB virtual server as the target virtual server. For more information, see Configuring a Content Switching Action.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Bind the CS policies to the CS virtual server. For more information, see Binding Policies to a Content Switching Virtual Server.
6. Bind the domain to the CS virtual server instead of the GSLB virtual server.

Sample Configuration

The following sample configuration sends requests from the client with IP address 5.5.5.5 to SERVICE_GSLB1 and SERVICE_GSLB2. SERVICE_GSLB1 has a higher priority than SERVICE_GSLB2, and SERVICE_GSLB2 serves the client requests only when SERVICE_GSLB1 is down. If both SERVICE_GSLB1 and SERVICE_GSLB2 are down, SERVICE_GSLB3 and service-GSLB4 are not considered, and a blank response is sent to the client.

Example COPY

```
>add cs vs CSVSERVER_GSLB http –targettype GSLB

Done

>add gslb vs VSERVER_GSLB1 http

Done

>add gslb vs VSERVER_GSLB2 http

Done

>add gslb vs VSERVER_GSLB_BACKUP1 http

Done

>set gslb vs VSERVER_GSLB1 -backupvserver VSERVER_GSLB_BACKUP1

Done

>add gslb service SERVICE_GSLB1 1.1.1.1 HTTP 80 -sitename site1

Done

>add gslb service SERVICE_GSLB2 1.1.1.2 HTTP 80 -sitename site1

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>add gslb service SERVICE_GSLB3 1.1.1.3 HTTP 80 -sitename site2

Done

>add gslb service SERVICE_GSLB4 1.1.1.4 HTTP 80 -sitename site2

Done

>bind gslb vs VSERVER_GSLB1 -servicename SERVICE_GSLB1

Done

>bind gslb vs VSERVER_GSLB_BACKUP1 -servicename SERVICE_GSLB2

Done

>bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB3

Done

>bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB4

Done

>add cs action a1 -targetvserver VSERVER_GSLB1

Done

>add cs policy p1 -rule "CLIENT.IP.SRC.EQ(5.5.5.5)" -action a1

Done

>bind cs vs CSVSERVER_GSLB -domainName www.abc.com

Done

>bind cs vs CSVSERVER_GSLB -policyname p1 -priority 1

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>add cs action a2 -targetvserver VSERVER_GSLB2

Done

>add cs policy p2 -rule "CLIENT.IP.SRC.EQ(6.6.6.6)" -action a2

Done

>bind cs vs CSVSERVER_GSLB -policyname p2 -priority 2

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Global Server Load Balancing for DNS Queries with NAPTR records

May 16, 2017

In a typical Global Server Load Balancing (GSLB) deployment, the NetScaler appliance receives DNS queries for A/AAAA records, selects the most appropriate GSLB service according to the configured load balancing method, and returns the service's IP address as a reply to the DNS query. You can now configure the appliance to receive DNS queries for NAPTR records and respond with the list of services configured for a domain. The appliance also monitors the health of the services, and in the response it provides a list of only the services that are up.

Example

In Telco deployments, you can configure a NetScaler appliance to receive DNS queries with NAPTR records from clients such as mobile management entities (MMEs), which play the role of a DNS resolver to discover all the services that are offered by the domain name. The appliance responds to the query with NAPTR records for all the services that are up. The MME can use this NAPTR response to run the S-NAPTR procedure to select the nodes on the basis of the service offered, colocation, topological closeness, and so on.

If multiple nodes qualify for selection, the MME can use the preference field in the NAPTR record from the NetScaler appliance to determine the node.

NAPTR Record Format

While responding to a DNS query with NAPTR record, a NetScaler appliance constructs a response NAPTR record for each GSLB service.

The following table lists the files in the NAPTR record:

Field	
Domain	The GSLB domain
TTL	The amount of time for which the NAPTR record can be cached.
Class	The class of the record. By default, this value is set to IN.
Type	The DNS record type.
Order	Specifies the order in which the NAPTR record MUST be processed. You can specify the order in the GSLB service. Otherwise, it is set to 1.
Preference	Specifies the order in which NAPTR records with equal "order" values SHOULD be processed, low numbers being processed before high numbers. If the order is not specified in the GSLB service, it is set to 1.
	Controls the aspects of the rewriting and interpretation of the fields in the record. The NetScaler appliance sets this value

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Field	
Service	Specifies the available service(s).
Regular Expression	Regular expressions are not supported, so this value is set to NULL.
Replacement	The domain name of the node that hosts the services.

Configuration Procedure

For detailed GSLB configuration instructions, see Configuring Global Server Load Balancing (GSLB). Make sure that you do the following:

- Set the following parameters while adding the GSLB virtual server:
  - serviceType: ANY
  - dnsRecordType: NAPTR
  - lbMethod: CUSTOMLOAD

  Example
  add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD

- While adding a GSLB site, set the *naptrReplacementSuffix* parameter to the domain name that you want to embed in the NAPTR records.

  Example
  add gslb site site1 10.102.218.200 -naptrReplacementSuffix example.com

- Set the following parameters while adding the GSLB service:
  - naptrreplacement
  - naptrOrder
  - naptrServices
  - naptrDomainTTL
  - naptrPreference

Sample Configuration

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD

Done

>add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com

Done

>add gslb service sgw1 3.3.3.13 ANY *  -siteName site1 -naptrreplacement sgw1.site1. -naptrOrder 2 -naptrServices x-3gpp-sgw:x-s5-gtp

Done

>add gslb service sgw2 3.3.3.11 ANY *  -siteName site1 -naptrreplacement sgw2.site1. -naptrOrder 5 -naptrServices x-3gpp-sgw:x-s5-gtp

Done

>add gslb service sgw3 3.3.3.12 ANY *  -siteName site2 -naptrreplacement sgw3.site1. -naptrOrder 10 -naptrServices x-3gpp-sgw:x-s5-g

>bind gslb vserver gslb_vs -serviceName sgw1

Done

>bind gslb vserver gslb_vs -serviceName sgw2

Done

>bind gslb vserver gslb_vs -serviceName sgw3

Done

>bind gslb service sgw1 -monitorName ping

Done

>bind gslb service sgw2 -monitorName ping

Done

>bind gslb service sgw3 -monitorName ping

Done

>bind gslb vserver gslb_vs -domainName gslb.com -TTL 5

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

DNS queries with NAPTR records are not supported in parent-child configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Link Load Balancing

Jul 03, 2017

Link load balancing (LLB) balances outbound traffic across multiple Internet connections provided by different service providers. LLB enables the Citrix® NetScaler® appliance to monitor and control traffic so that packets are transmitted seamlessly over the best possible link. Unlike with server load balancing, where a service represents a server, with LLB, a service represents a router or the next hop. A link is a connection between the NetScaler and the router.

To configure link load balancing, many users begin by configuring a basic setup with default settings. Configuring a basic setup involves configuring services, virtual servers, monitors, routes, an LLB method, and, optionally, configuring persistence. Once a basic setup is operational, you can customize it for your environment.

Load balancing methods that are applicable to LLB are round robin, destination IP hash, least bandwidth, and least packets. You can optionally configure persistence for connections to be sustained on a specific link. The available persistence types are source IP address-based, destination IP address-based, and source IP and destination IP address-based. PING is the default monitor but configuring a transparent monitor is recommended.

You can customize your setup by configuring reverse NAT (RNAT) and backup links.

This document includes the following information:

- Configuring a Basic LLB Setup
- Configuring RNAT with LLB
- Configuring a Backup Route
- Resilient LLB Deployment Scenario
- Monitoring an LLB Setup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Basic LLB Setup

Jul 13, 2017

To configure LLB, you first create services representing each router to the Internet Service Providers (ISPs). A PING monitor is bound by default to each service. Binding a transparent monitor is optional but recommended. Then, you create a virtual server, bind the services to the virtual server, and configure a route for the virtual server. The route identifies the virtual server as the gateway to the physical routers represented by the services. The virtual server selects a router by using the load balancing method that you specify. Optionally, you can configure persistence to make sure that all traffic for a particular session is sent over a specific link.

To configure a basic LLB setup, do the following:

- Configure services
- Configure an LLB virtual server and binding a service
- Configure the LLB method and persistence
- Configure an LLB route
- Create and bind a transparent monitor

Configuring Services

Updated: 2014-10-27

A default monitor (PING) is automatically bound to a service type of ANY when the service is created, but you can replace the default monitor with a transparent monitor, as described in "Creating and Binding a Transparent Monitor."

To create a service by using the command line interface

At the command prompt, type:

- add service <name> <IP> <serviceType> <port>
- show service <name>

Example

```
add service ISP1R_svc_any 10.10.10.254 any *
show service ISP1R_svc_any
    ISP1R_svc_any (10.10.10.254:*) - ANY
    State: DOWN
    Last state change was at Tue Aug 31 04:31:13 2010
    Time since last state change: 2 days, 05:34:18.600
    Server Name: 10.10.10.254
    Server ID : 0   Monitor Threshold : 0
    Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
    TCP Buffering(TCPB): YES
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    HTTP Compression(CMP): NO
    Idle timeout: Client: 120 sec   Server: 120 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: OFF
    Down state flush: ENABLED

1)    Monitor Name: ping
          State: UP       Weight: 1
          Probes: 244705  Failed [Total: 0 Current: 0]
          Last response: Success - ICMP echo reply received.
          Response Time: 1.322 millisec
 Done
```

To create services by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, and create a service.

To create services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters:
   - Service Name*—name
   - Server—IP
   - Protocol*—serviceType (Select ANY from the drop-down list.)
   - Port*—port

   * A required parameter

4. Click Create.
5. Repeat Steps 2-4 to create another service.
6. Click Close.
7. In the Services pane, select the services that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configuring an LLB Virtual Server and Binding a Service

Updated: 2014-10-28

After you create a service, create a virtual server and bind services to the virtual server. The default LB method of least connections is not supported in LLB. For information about changing the LB method, see "Configuring the LLB Method and Persistence."

To create a link load balancing virtual server and bind a service by using the command line interface

At the command prompt, type:

- add lb vserver <name> <serviceType>
- bind lb vserver < name> <serviceName>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- show lb vserver < name>

Example

```
add lb vserver LLB-vip any
bind lb vserver LLB-vip ISP1R_svc_any
sh lb vserver LLB-vip
    LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
    State: DOWN
    Last state change was at Thu Sep  2 10:51:32 2010
    Time since last state change: 0 days, 17:51:46.770
    Effective State: DOWN
    Client Idle Timeout: 120 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    No. of Bound Services :  1 (Total)      0 (Active)
    Configured Method: ROUNDROBIN
    Mode: IP
    Persistence: NONE
    Connection Failover: DISABLED

1) ISP1R_svc_any (10.10.10.254: *) - ANY State: DOWN    Weight: 1
 Done
```

To create a link load balancing virtual server and bind a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and create a virtual server for link load balancing. Specify **ANY** in the **Protocol** field.
   Note: Make sure that **Directly Addressable** is unchecked.
2. Under the **Services** tab, in the **Active** column, select the check box for the service that you want to bind to the virtual server.

Configuring the LLB Method and Persistence

Updated: 2014-10-28

By default, the NetScaler appliance uses the least connections method to select the service for redirecting each client request, but you should set the LLB method to one of the supported methods. You can also configure persistence, so that different transmissions from the same client are directed to the same server.

To configure the LLB method and/or persistence by using the command line interface

At the command prompt, type the following command:

- set lb vserver <name> -lbMethod <lbMethod> -persistenceType <persistenceType>
- show lb vserver <name>

Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set lb vserver LLB-vip -lbmethod ROUNDROBIN -persistencetype SOURCEIP

show lb vserver LLB-vip
    LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
    State: DOWN
    Last state change was at Fri Sep  3 04:46:48 2010
    Time since last state change: 0 days, 00:52:21.200
    Effective State: DOWN
    Client Idle Timeout: 120 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    No. of Bound Services :  0 (Total)       0 (Active)
    Configured Method: ROUNDROBIN
    Mode: IP
    Persistence: SOURCEIP
  Persistence Mask: 255.255.255.255       Persistence v6MaskLength: 128   Persistence Timeout: 2 min
    Connection Failover: DISABLED
```

To configure the link load balancing method and/or persistence by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the virtual server for which you want to configure the load balancing method and/or persistence settings.
2. In the **Advanced** section, select Method and configure the load balancing method.
3. In the **Advanced** section, select **Persistence** and configure the persistence parameters.

Configuring an LLB Route

Updated: 2014-10-28

After configuring the IPv4 or IPv6 services, virtual servers, LLB methods, and persistence, you configure an IPv4 or IPv6 LLB route for the network specifying the LLB virtual server as the gateway. A route is a collection of links that are load balanced. Requests are sent to the LLB virtual server IP address that acts as the gateway for all outbound traffic and selects the router based on the LLB method configured.

To configure an IPv4 LLB route by using the command line interface

At the command prompt, type:

- add lb route <network> <netmask> <gatewayName>
- show lb route [<network> <netmask>]

Example

```
add lb route 0.0.0.0 0.0.0.0 LLB-vip
show lb route 0.0.0.0 0.0.0.0
     Network         Netmask       Gateway/VIP           Flags
    -----------     -------------  --------------        --------
1)   0.0.0.0         0.0.0.0       LLB-vip               UP
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure an IPv6 LLB route by using the command line interface

At the command prompt, type:

- add lb route6 <network> <gatewayName>
- show lb route6

add lb route6 ::/0 llb6_vs show lb route6 Network VIP Flags ---------- --------- -------- 1) ::/0 llb6_vs UP
Example

To configure an LLB route by using the configuration utility

Navigate to System > Network > Routes, and select **LLB**, and configure the LLB route.

Note: Select LLBV6 to configure an IPV6 route.

To configure an LLB route by using the configuration utility

1. Navigate to System > Network > Routes.
2. In the details pane, select one of the following:
   - Click LLB to configure an IPv4 route.
   - Click LLBV6 to configure an IPv4 route.
3. In the Create LB Route or Create LB IPV6 Routedialog box, set the following parameters:
   - Network*
   - Netmask*—Required for IPV4 routes.
   - Gateway Name*—gatewayName

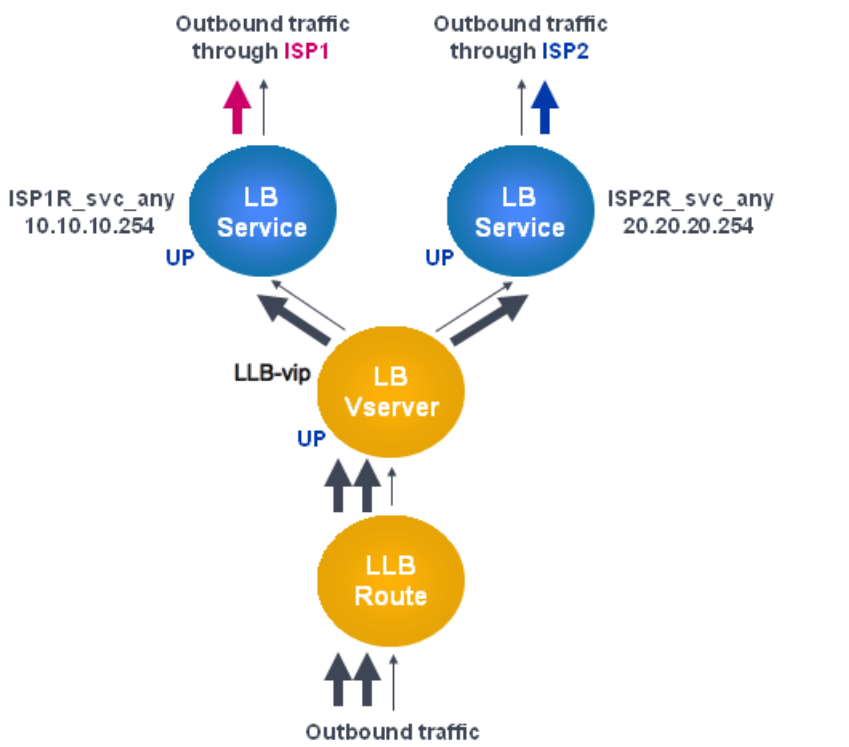
   * A required parameter

4. Click Create, and then click Close. The route that you just created appears on the LLB or the LLB6 tab in the Routes pane.

The following diagram shows a basic LLB setup. A service is configured for each of the two links (ISPs) and PING monitors are bound by default to these services. A link is selected based on the LLB method configured.

Figure 1. Basic LLB Setup

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Note

If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

Creating and Binding a Transparent Monitor

Updated: 2014-10-28

You create a transparent monitor to monitor the health of upstream devices, such as routers. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the router is UP but one of the next hop devices from that router is down, the appliance includes the router while performing load balancing and forwards the packet to the router. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the router) are down, the service is marked as DOWN and the router is not included when the appliance performs link load balancing.

To create a transparent monitor by using the command line interface

At the command prompt, type:

- add lb monitor <monitorName> <type> -destIP <ip_addr|*> -transparent YES
- show lb monitor [<monitorName>]

Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add lb monitor monitor-1 PING -destIP 10.10.10.11 -transparent YES
> show lb monitor monitor-1
1)  Name.......: monitor-1  Type......:     PING   State....:   ENABLED
Standard parameters:
  Interval.........:          5 sec   Retries..........:             3
  Response timeout.:          2 sec   Down time........:         30 sec
  Reverse..........:            NO   Transparent.......:           YES
  Secure...........:            NO   LRTM.............:        ENABLED
  Action...........:   Not applicable   Deviation.........:          0 sec
  Destination IP...:     10.10.10.11
  Destination port.:    Bound service
  Iptunnel.........:            NO
  TOS..............:            NO   TOS ID...........:             0
  SNMP Alert Retries:            0     Success Retries..:             1
  Failure Retries..:             0
```

To create a transparent monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors and configure a transparent monitor.

To create a transparent monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. In the Monitors pane, click Add.
3. In the Create Monitor dialog box, set the following parameters:

   - Name*
   - Type*
   - Destination IP
   - Transparent

   * A required parameter

4. Click Create, and then click Close.
5. In the Monitors pane, select the monitor that you just configured and verify that the settings displayed in the Details pane are correct.

To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. On the **Monitors** tab, under **Available**, select the monitor that you want to bind to the service, and then click **Add**.

To bind a monitor to a service by using the command line interface

At the command prompt, type:

- bind lb monitor <monitorName> <serviceName>
- show service <name>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

```
bind lb monitor monitor-HTTP-1 ISP1R_svc_any
 Done
> show service ISP1R_svc_any
      ISP1R_svc_any (10.10.10.254:*) - ANY
      State: UP
      Last state change was at Thu Sep  2 10:51:07 2010
      Time since last state change: 0 days, 18:41:55.130
      Server Name: 10.10.10.254
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): YES
      HTTP Compression(CMP): NO
      Idle timeout: Client: 120 sec   Server: 120 sec
      Client IP: DISABLED
      Cacheable: NO
      SC: OFF
      SP: OFF
      Down state flush: ENABLED

1)     Monitor Name: monitor-HTTP-1
            State: UP  Weight: 1
            Probes: 1256       Failed [Total: 0 Current: 0]
            Last response: Success - ICMP echo reply received.
            Response Time: 1.322 millisec
 Done
```

To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, select a service to which you want to bind a monitor, and then click Open.
3. In the Configure Service dialog box, on the Monitors tab, under Available, select the monitor that you want to bind to the service, and then click Add.
4. Click OK.
5. In the Services pane, select the service that you just configured and verify that the settings displayed in the Details pane are correct.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring RNAT with LLB

Dec 15, 2016

You can configure an LLB setup for reverse network address translation (RNAT) for outbound traffic. This ensures that the return network traffic for a specific flow is routed through the same path. First configure basic LLB, as described in "Configuring a Basic LLB Setup", and then configure RNAT. You must then enable use subnet IP (USNIP) mode.

To add SNIPs for ISP routers by using the command line interface

At the command prompt, type:

- **Add IP NS** <network><subnet of first ISP in the IP router> <subnet mask> **-type SNIP**
- **Add IP NS** <in the IP subnet of second ISP router> <subnet mask> **-type SNIP**

example COPY

```
add ns ip 10.140.23.1 255.255.255.0 -type snip

add ns ip 10.141.23.1 255.255.255.0 -type snip
```

To configure RNAT by using the command line interface

At the command prompt, type:

- set rnat <network> <netmask>
- show rnat

example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> set rnat 10.102.29.0 255.255.255.0 -natIP 10.140.23.1

> set rnat 10.102.29.0 255.255.255.0 -natIP 10.141.23.1

> show rnat

  1) Network: 10.102.29.0      Netmask: 255.255.255.0

     NatIP: 10.140.23.1

  2) Network: 10.102.29.0      Netmask: 255.255.255.0

     NatIP: 10.141.23.1
```

To configure RNAT by using the configuration utility

1. Navigate to System > Network > Routes.
2. On the **RNAT** tab, from the **Actions** drop-down list, select **Configure RNAT**.
3. Specify the network on which to perform RNAT.

> **Note**
>
> You can also configure RNAT by using Access Control Lists (ACLs). Refer Configuring RNAT for details.

To enable Use Subnet IP mode by using the command line interface

At the command prompt, type:

- enable ns mode USNIP
- show ns mode

Example

```
enable ns mode USNIP
> show ns mode
     Mode                    Acronym          Status
     -------                 -------          ------
 1)  Fast Ramp               FR               ON
 2)  ....
 8)  Use Subnet IP           USNIP            ON
 9)  ...
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To enable Use Subnet IP mode by using the configuration utility

1. Navigate to System > Settings and, under **Modes and Features**, click **Configure Modes**.
2. In the **Configure Modes** dialog box, select **Use Subnet IP**, and then click **OK**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

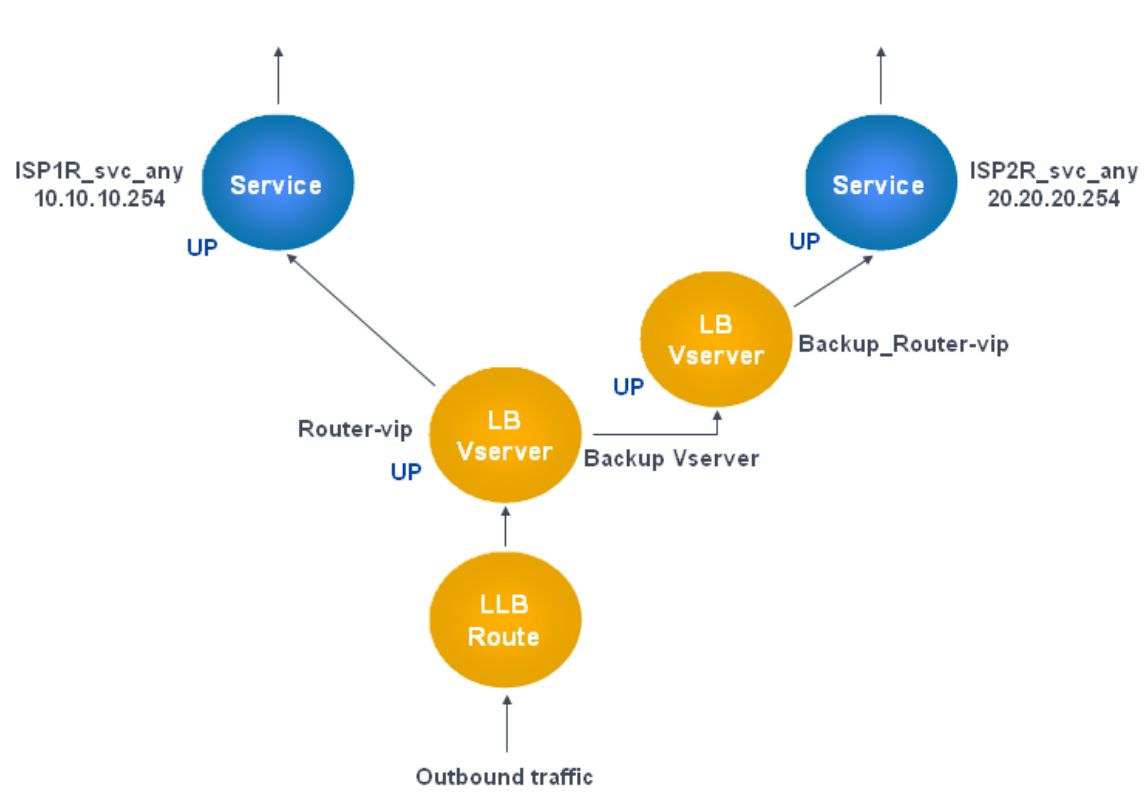
Configuring a Backup Route

Jul 13, 2017

To prevent disruption in services when the primary route is down, you can configure a backup route. Once the backup route is configured, the NetScaler appliance automatically uses it when the primary route fails. First create a primary virtual server as described in "Configuring an LLB Virtual Server and Binding a Service." To configure a backup route, create a secondary virtual server similar to a primary virtual server and then designate this virtual server as a backup virtual server (route).

In the following diagram, **Router-vip** is the primary virtual server, and **Backup_Router-vip** is the secondary virtual server designated as the backup virtual server.

Figure 1. Backup Route Setup

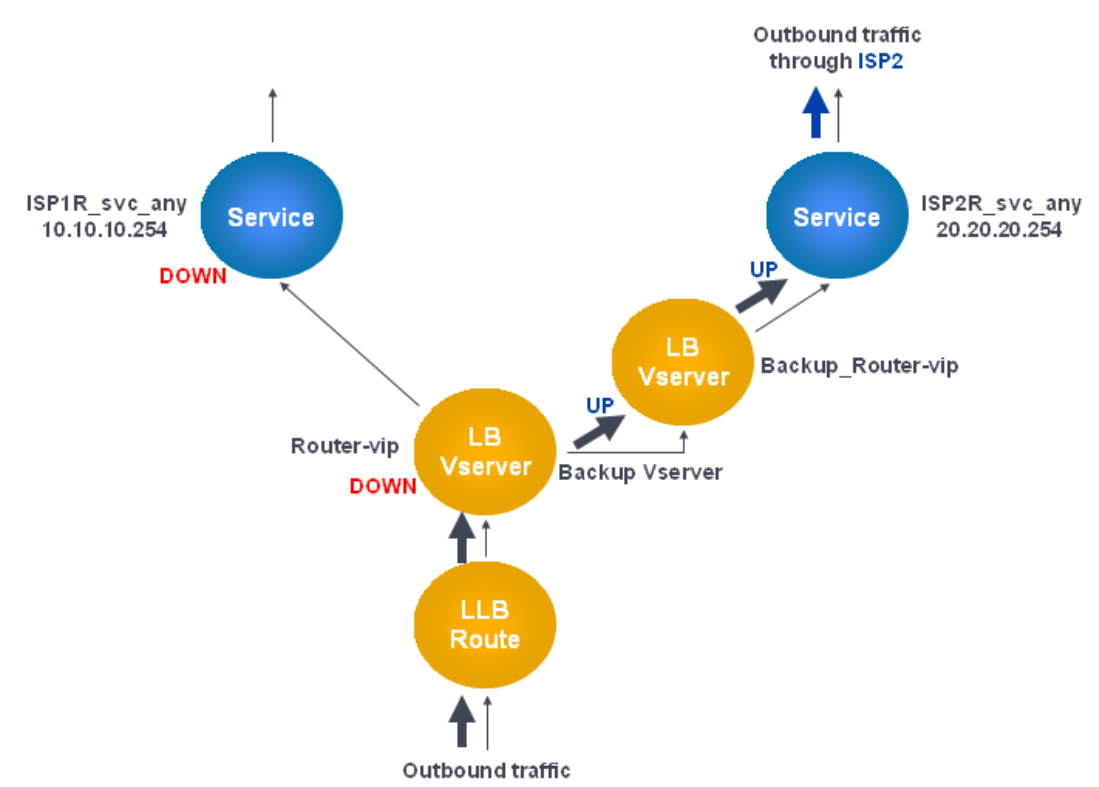


Note: If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

By default, all traffic is sent through the primary route. However, when the primary route fails, all traffic is diverted to the backup route as shown in the following diagram.

Figure 2. Backup Routing in Operation

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Note: If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

To set the secondary virtual server as the backup virtual server by using the command line interface

At the command prompt, type:

set lb vserver <name> -backupVserver <string>

Example

```
set lb vserver Router-vip -backupVServer Backup_Router-vip
> show lb vserver Router-vip
    Router-vip (0.0.0.0:0) - ANY    Type: ADDRESS
    State: UP
    Last state change was at Fri Sep  3 04:46:48 2010
    Time since last state change: 0 days, 03:09:45.600
    Effective State: UP
    Client Idle Timeout: 120 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    No. of Bound Services :  1 (Total)      1 (Active)
    Configured Method: ROUNDROBIN
    Mode: IP
    Persistence: DESTIP     Persistence Mask: 255.255.255.255      Persistence v6MaskLength: 128   Persistence Timeout: 2 min
    Backup: Router2-vip
    Connection Failover: DISABLED
 Done
```

To set the secondary virtual server as the backup virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the secondary virtual server for which you

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

want to configure the backup virtual server.

2. In the **Load Balancing Virtual Server** dialog box, under **Advanced**, select **Protection**.
3. In the **Backup Virtual Server** drop-down list, select the secondary backup virtual server, and then click **OK**.

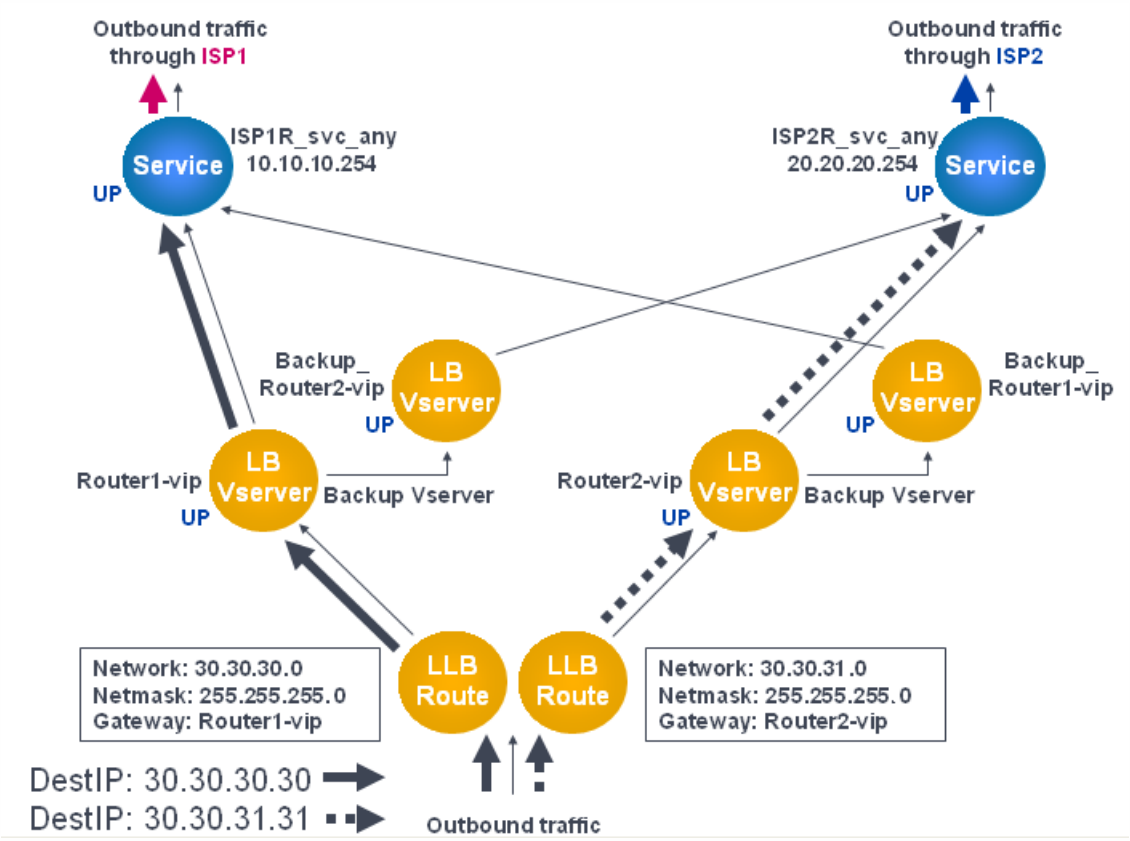
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Resilient LLB Deployment Scenario

Mar 22, 2012

In the following diagram, there are two networks: 30.30.30.0 and 30.30.31.0. Link load balancing is configured based on the destination IP address. Two routes are configured with gateways **Router1-vip** and **Router2-vip**, respectively. **Router1-vip** is configured as a backup to **Router2-vip** and vice versa. All traffic with the destination IP specified as 30.30.30.30 is sent through **Router1-vip** and traffic with the destination IP specified as 30.30.31.31 is sent through **Router2-vip**.

Figure 1. Resilient LLB Deployment Setup

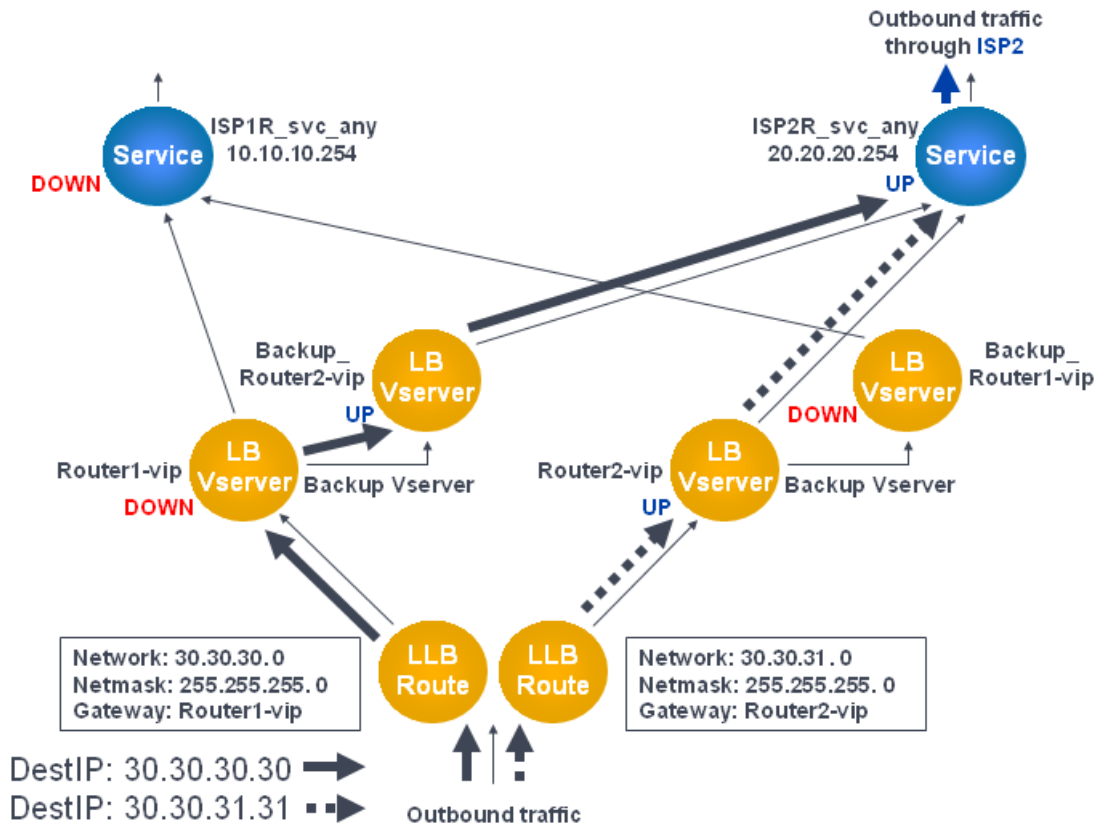


Note: If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

However, if any one of the gateways (**Router1-vip** or **Router2-vip**) is DOWN, traffic is routed through the backup router. In the following diagram, **Router1-vip** for ISP1 is DOWN, so all traffic with the destination IP specified as 30.30.30.30 is also sent through ISP2.

Figure 2. Resilient LLB Deployment Scenario

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Note: If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring an LLB Setup

Sep 20, 2010

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

Updated: 2013-09-05

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the command prompt, type:

stat lb vserver [-detail] [<name>]

Example

```
>stat lb vserver -detail
Virtual Server(s) Summary
                vsvrIP  port    Protocol      State    Req/s   Hits/s
One                  *    80       HTTP         UP      5/s      0/s
Two                  *     0        TCP       DOWN      0/s      0/s
Three                *  2598        TCP       DOWN      0/s      0/s
dnsVirtualNS   10.102.29.90   53        DNS       DOWN      0/s      0/s
BRVSERV         10.10.1.1    80       HTTP       DOWN      0/s      0/s
LBVIP        10.102.29.66    80       HTTP         UP      0/s      0/s
 Done
```

To display virtual server statistics by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers > Statistics.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

Viewing the Statistics of a Service

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2013-08-28

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the command line interface

At the command prompt, type:

stat service <name>

Example

```
stat service Service-HTTP-1
```

To view the statistics of a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing

Mar 24, 2015

The load balancing feature distributes user requests for web pages and other protected applications across multiple servers that all host (or mirror) the same content. You use load balancing primarily to manage user requests to heavily used applications, preventing poor performance and outages and ensuring that users can access your protected applications. Load balancing also provides fault tolerance; when one server that hosts a protected application becomes unavailable, the feature distributes user requests to the other servers that host the same application.

You can configure the load balancing feature to:

- Distribute all requests for a specific protected website, application, or resource between two or more identically configured servers.
- Use any of several different algorithms to determine which server should receive each incoming user request, basing the decision on different factors, such as which server has the fewest current user connections or which server has the lightest load.

The load balancing feature is a core feature of the NetScaler appliance. Most users first set up a working basic configuration and then customize various settings, including persistence for connections. In addition, you can configure features for protecting the configuration against failure, managing client traffic, managing and monitoring servers, and managing a large scale deployment.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How Load Balancing Works

Dec 27, 2017

In a basic load balancing setup, clients send their requests to the IP address of a virtual server configured on the NetScaler appliance. The virtual server distributes them to the load-balanced application servers according to a preset pattern, called the load balancing algorithm. In some cases, you might want to assign the load balancing virtual server a wildcard address instead of a specific IP address. For instructions about specifying a global HTTP port on the appliance, see Global HTTP Ports.

To understand how load balancing works, see the following sections:

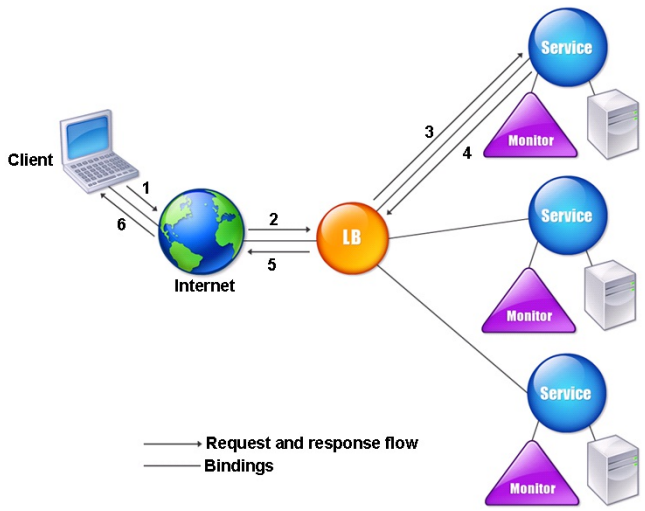
- Load Balancing Basics
- Understanding the Topology
- Use of Wildcards Instead of IP Addresses and Ports

To understand how load balancing works, see the following sections:

Load Balancing Basics

A load balancing setup includes a load-balancing virtual server and multiple load-balanced application servers. The virtual server receives incoming client requests, uses the load balancing algorithm to select an application server, and forwards the requests to the selected application server. The following conceptual drawing illustrates a typical load balancing deployment. Another variation involves assigning a global HTTP port.

Figure 1. Load Balancing Architecture



The load balancing virtual server can use any of a number of algorithms (or methods) to determine how to distribute load among the load-balanced servers that it manages. The default load balancing method is the least connection method, in which the NetScaler appliance forwards each incoming client connection to whichever load-balanced application server currently has the fewest active user connections.

The entities that you configure in a typical NetScaler load balancing setup are:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Load balancing virtual server.** The IP address, port, and protocol combination to which a client sends connection requests for a particular load-balanced website or application. If the application is accessible from the Internet, the virtual server IP (VIP) address is a public IP address. If the application is accessible only from the local area network (LAN) or wide area network (WAN), the VIP is usually a private (ICANN non-routable) IP address.
- **Service.** The IP address, port, and protocol combination used to route requests to a specific load-balanced application server. A service can be a logical representation of the application server itself, or of an application running on a server that hosts multiple applications. After creating a service, you bind it to a load balancing virtual server.
- **Server object.** A virtual entity that enables you to assign a name to a physical server instead of identifying the server by its IP address. If you create a server object, you can specify its name instead of the server's IP address when you create a service. Otherwise, you must specify the server's IP address when you create a service, and the IP address becomes the name of the server.
- **Monitor.** An entity on the NetScaler appliance that tracks a service and ensures that it is operating correctly. The monitor periodically probes (or performs a health check on) each service to which you assign it. If the service does not respond within the time specified by the time-out, and a specified number of health checks fail, that service is marked DOWN. The NetScaler appliance then skips that service when performing load balancing, until the issues that caused the service to quit responding are fixed.

The virtual server, services, and load balanced application servers in a load balancing setup can use either Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) IP addresses. You can mix IPv4 and IPv6 addresses in a single load balancing setup.

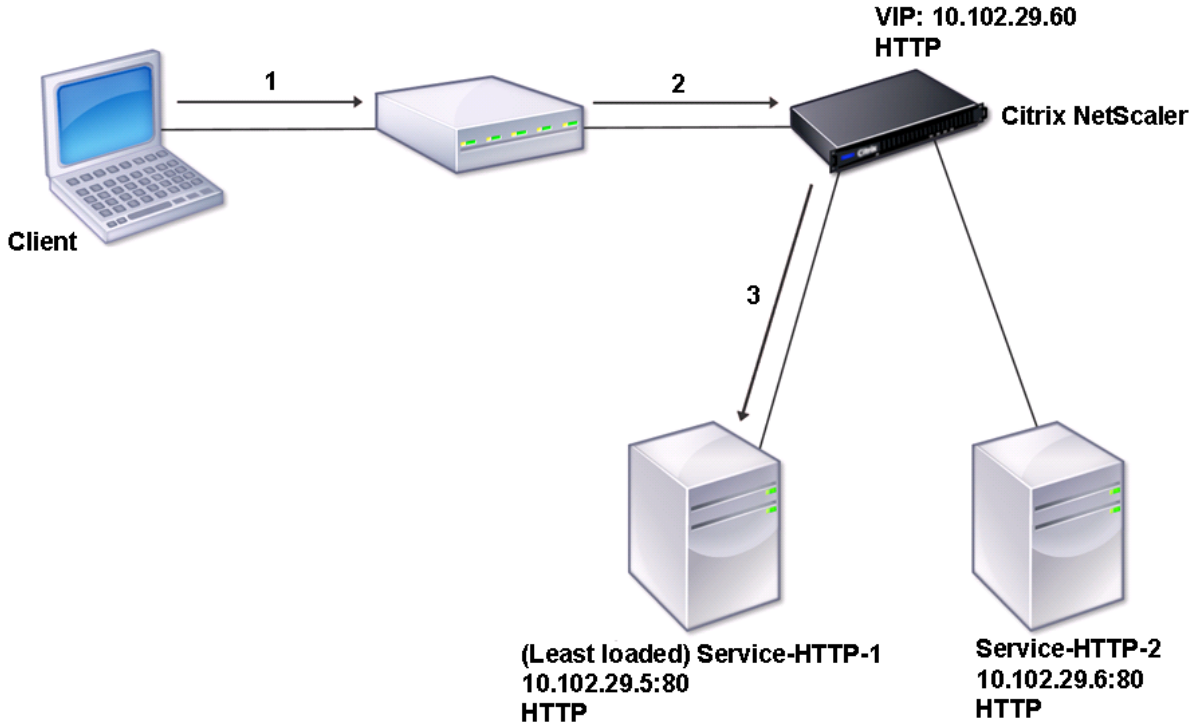
For variations in the load balancing setup, see the following use cases:

- Configuring Load Balancing in Direct Server Return Mode
- Configuring LINUX Servers in DSR Mode
- Configuring DSR Mode When Using TOS
- Configuring Load Balancing in DSR Mode by Using IP Over IP
- Configuring Load Balancing in One-arm Mode
- Configuring Load Balancing in the Inline Mode
- Load Balancing of Intrusion Detection System Servers
- Load Balancing of Remote Desktop Protocol (RDP) Servers

Understanding the Topology

In a load balancing setup, the load balancing server is logically located between the client and the server farm, and manages traffic flow to the servers in the server farm. On the NetScaler appliance, the application servers are represented by virtual entities called services. The following diagram shows the topology of a basic load balancing configuration.
Figure 2. Basic Load Balancing Topology

© 1999-2017 Citrix Systems, Inc. All rights reserved.



In the diagram, load balancing is used to manage traffic flow to the servers. The virtual server selects the service and assigns it to serve client requests. Consider a scenario where the services Service-HTTP-1 and Service-HTTP-2 are created and bound to the virtual server named Vserver-LB-1. Vserver-LB-1 forwards the client request to either Service-HTTP-1 or Service-HTTP-2. The NetScaler appliance uses the least connection load balancing method to select the service for each request. The following table lists the names and values of the basic entities that must be configured on the appliance.

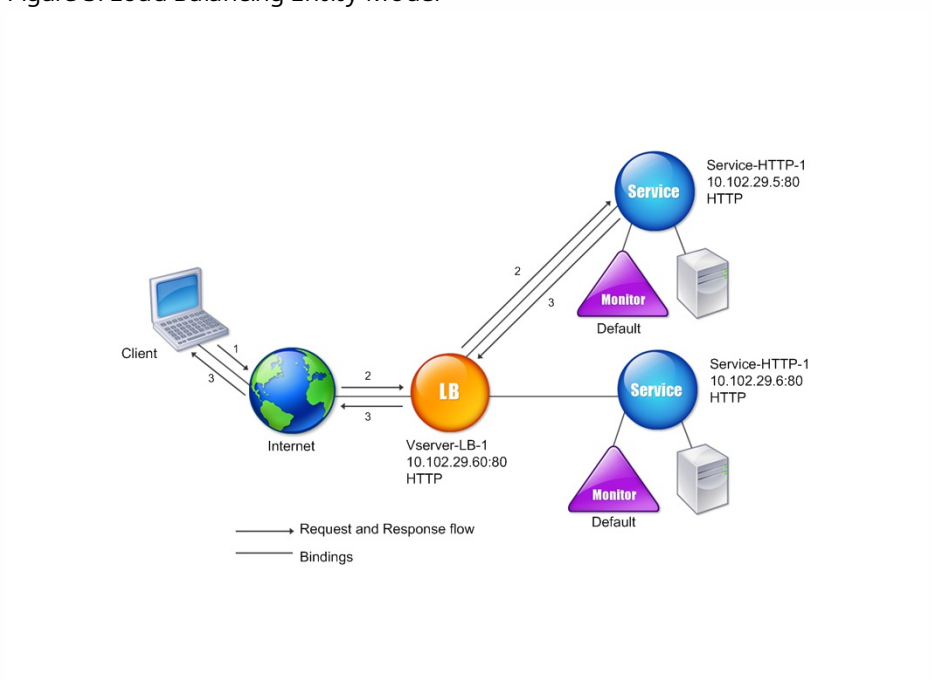
Entity	**Mandatory Parameters and Sample Values**			
	Name	**IP Address**	**Port**	**Protocol**
Virtual server	Vserver-LB-1	10.102.29.60	80	HTTP
Services	Service-HTTP-1	10.102.29.5	80	HTTP
	Service-HTTP-2	10.102.29.6	80	HTTP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitors	Default	None	None	None

The following diagram shows the load balancing sample values and mandatory parameters that are described in the preceding table.

Figure 3. Load Balancing Entity Model



Use of Wildcards Instead of IP Addresses and Ports

In some cases you might need to use a wildcard for the IP address or the port of a virtual server or for the port of a service. The following cases may require using a wildcard:

- If the NetScaler appliance is configured as a transparent pass through, which must accept all traffic that is sent to it regardless of the IP or port to which it is sent.
- If one or more services listen on ports that are not well known.
- If one or more services, over time, change the ports that they listen on.
- If you reach the limit for the number of IP addresses and ports that you can configure on a single NetScaler appliance.
- If you want to create virtual servers that listen for all traffic on a specific virtual LAN.

When a wildcard-configured virtual server or service receives traffic, the NetScaler appliance determines the actual IP address or port and creates new records for the service and associated load balanced application server. These dynamically created records are called dynamically learned server and service records.

For example, a firewall load balancing configuration can use wildcards for both the IP address and port. If you bind a wildcard TCP service to this type of load balancing virtual server, the virtual server receives and processes all TCP traffic that does not match any other service or virtual server.

The following table describes some of the different types of wildcard configurations and when each should be used.

IP	Port	Protocol	Description
*	*	TCP	A general wildcard virtual server that accepts traffic sent to any IP address and port on

© 1999-2017 Citrix Systems, Inc. All rights reserved.

IP	Port	Protocol	Description
			the NetScaler appliance. When using a wildcarded virtual server, the appliance dynamically learns the IP and port of each service and creates the necessary records as it processes traffic.
*	*	TCP	A firewall load balancing virtual server. You can bind firewall services to this virtual server, and the NetScaler appliance passes traffic through the firewall to the destination.
IP Address	*	TCP,UDP, and ANY	A virtual server that accepts all traffic that is sent to the specified IP address, regardless of the port. You must explicitly bind to this type of virtual server the services to which it will redirect traffic. It will not dynamically learn them. Note: You do not configure services or virtual servers for a global HTTP port. In this case, you configure a specific port as a global HTTP port (for example, set ns param -httpPort 80). The appliance then accepts all traffic that matches the port number, and processes it as HTTP traffic. The appliance dynamically learns and creates services for this traffic.
*	port	SSL, SSL_TCP	A virtual server that accepts all traffic sent to any IP address on a specific port. Used for global transparent SSL offloading. All SSL, HTTP, and TCP processing that usually is performed for a service of the same protocol type is applied to traffic that is directed to this specific port. The appliance uses the port to dynamically learn the IP of the service it should use. If –cleartext is not specified, the NetScaler appliance uses end-to-end SSL.
*	port	Not applicable	All other virtual servers that can accept traffic to the port. You do not bind services to these virtual servers; the NetScaler appliance learns them dynamically.

Note: If you have configured your NetScaler appliance as a transparent pass through that makes use of global (wildcard) ports, you may want to turn on Edge mode. For more information, see "Configuring Edge Mode."
The NetScaler appliance attempts to locate virtual servers and services by first attempting an exact match. If none is found, it continues to search for a match based on wildcards, in the following order:

1. Specific IP address and specific port number
2. Specific IP address and a * (wildcard) port
3. * (wildcard) IP address and a specific port
4. * (wildcard) IP address and a * (wildcard) port

If the appliance is unable to select a virtual server by IP address or port number, it searches for a virtual server on the basis of the protocol used in the request, in the following order:

1. HTTP
2. TCP
3. ANY

Configuring Global HTTP Ports

You do not configure services or virtual servers for a global HTTP port. Instead, you configure a specific port by using the set ns param command. After configuring this port, the NetScaler appliance accepts all traffic that matches the port number, and processes it as HTTP traffic, dynamically learning and creating services for that traffic.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

You can configure more than one port number as a global HTTP port. If you are specifying more than one port number in a single set ns param command, separate the port numbers by a single white space. If one or more ports have already been specified as global HTTP ports, and you want to add one or more ports without removing the ports that are currently configured, you must specify all the port numbers, current and new, in the command. Before you add port numbers, use the show ns param command to view the ports that are currently configured.

To configure a global HTTP port by using the command line interface

At the command prompt, type the following commands to configure a global HTTP port and verify the configuration:

- set ns param –httpPort <port>
- show ns param

Example 1: Configuring a port as a global HTTP port

In this example, port 80 is configured as a global HTTP port.

```
> set ns param -httpPort 80
 Done
> show ns param
        Global configuration settings:
                    HTTP port(s): 80
                 Max connections: 0
          Max requests per connection: 0
               Client IP insertion: DISABLED
                   Cookie version: 0
       Persistence Cookie Secure Flag: ENABLED
       ...
       ...
```

Example 2: Adding ports when one or more global HTTP ports are already configured

In this example, port 8888 is added to the global HTTP port list. Port 80 is already configured as a global HTTP port.

```
> show ns param

       Global configuration settings:
                    HTTP port(s): 80
                 Max connections: 0
          Max requests per connection: 0
               Client IP insertion: DISABLED
                   Cookie version: 0
       Persistence Cookie Secure Flag: ENABLED
                    Min Path MTU: 576
       ...
       ...
 Done
> set ns param -httpPort 80 8888
 Done
>  show ns param
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Global configuration settings:
                HTTP port(s): 80,8888
             Max connections: 0
  Max requests per connection: 0
        Client IP insertion: DISABLED
             Cookie version: 0
    Persistence Cookie Secure Flag: ENABLED
              Min Path MTU: 576

    ...
    ...
 Done
>
```

To configure a global HTTP port by using the configuration utility

1. Navigate to System > Settings > Change HTTP Parameters, and then add an HTTP port number.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting Up Basic Load Balancing

Nov 10, 2016

Before configuring your initial load balancing setup, enable the load balancing feature. Then begin by creating at least one service for each server in the load balancing group. With the services configured, you are ready to create a load balancing virtual server, and bind each service to the virtual server. That completes the initial setup. Before proceeding with further configuration, verify your configuration to make sure that each element was configured properly and is operating as expected.

To set up basic load balancing, see the following sections:

- Enabling Load Balancing
- Configuring a Server Object
- Configuring Services
- Creating a Virtual Server
- Binding Services to the Virtual Server
- Verifying the Configuration

Enabling Load Balancing

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

- enable ns feature LB
- show ns feature

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> enable ns feature LoadBalancing

 Done

> show ns feature


      Feature                  Acronym          Status

      -------                  -------          ------

1)    Web Logging              WL               OFF

2)    Surge Protection         SP               ON

3)    Load Balancing           LB               ON

.

.

.

24)   NetScaler Push           push             OFF

Done
```

To enable load balancing by using the configuration utility

Navigate to **System** > **Settings** and, in **Configure Basic Features**, select **Load Balancing**.

Configuring a Server Object

Create an entry for your server on the NetScaler appliance. The NetScaler appliance supports IP address based servers and domain-based servers. If you create an IP address based server, you can specify the name of ther server instead of its IP address when you create a service. For information about setting up DNS for a domain-based server, see Domain Name System.

To create a server object by using the NetScaler command line

At the comand prompt, type:

add server <name>@ <IPAddress>@ | <domain>

To create a server object by using the NetScaler GUI

Navigate to **Traffic Management** > **Load Balancing** > **Servers**, and add a server object.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Example COPY

```
add server web_serv 10.102.27.150
```

Configuring Services

After you enable the load balancing feature, you must create at least one service for each application server that is to be included in your load balancing setup. The services that you configure provide the connections between the NetScaler appliance and the load balanced servers. Each service has a name and specifies an IP address, a port, and the type of data that is served.

If you create a service without first creating a server object, the IP address of the service is also the name of the server that hosts the service. If you prefer to identify servers by name rather than IP address, you can create server objects and then specify a server's name instead of its IP address when you create a service.

When you create a service that uses UDP as the transport layer protocol, a ping monitor is automatically bound to the service. A ping monitor is the most basic of the built-in monitors. When you create a service that uses TCP as the transport layer protocol, a TCP_default monitor is automatically bound to the service. When you develop a strategy for managing your load balancing setup, you might decide to bind a different type of monitor, or multiple monitors, to the service.

Creating a Service

Before you create a service, you need to understand the different service types and how each is used. The following list describes the types of services supported on the NetScaler appliance.

HTTP

Used for load-balanced servers that accept HTTP traffic, such as standard web sites and web applications. The HTTP service type enables the NetScaler appliance to provide compression, content filtering, caching, and client keep-alive support for your Layer 7 web servers. This service type also supports virtual server IP port insertion, redirect port rewriting, Web 2.0 Push, and URL redirection support.

Because HTTP is a TCP-based application protocol, you can also use the TCP service type for web servers. If you do so, however, the NetScaler appliance is able to perform only Layer 4 load balancing. It cannot provide any of the Layer 7 support described earlier.

SSL

Used for servers that accept HTTPS traffic, such as ecommerce web sites and shopping cart applications. The SSL service type enables the NetScaler appliance to encrypt and decrypt SSL traffic (perform SSL offloading) for your secure web applications. It also supports HTTP persistence, content switching, rewrite, virtual server IP port insertion, Web 2.0 Push, and URL redirection.

You can also use the SSL_BRIDGE, SSL_TCP, or TCP service types. If you do so, however, the NetScaler performs only Layer 4 load balancing. It cannot provide SSL offloading or any of the Layer 7 support described above

FTP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Used for servers that accept FTP traffic. The FTP service type enables the NetScaler appliance to support specific details of the FTP protocol.

You can also use TCP or ANY service types for FTP servers.

TCP

Used for servers that accept many different types of TCP traffic, or that accept a type of TCP traffic for which a more specific type of service is not available.

You can also use the ANY service type for these servers.

SSL_TCP

Used for servers that accept non-HTTP-based SSL traffic, to support SSL offloading.

You can also use the TCP service type for these services. If you do so, the NetScaler appliance performs both the Layer 4 load balancing and SSL offloading.

UDP

Used for servers that accept UDP traffic. You can also use the ANY service type.

SSL_BRIDGE

Used for servers that accept SSL traffic when you do not want the NetScaler appliance to perform SSL offloading. Alternatively, you can use the SSL_TCP service type.

NNTP

Used for servers that accept Network News Transfer Protocol (NNTP) traffic, typically Usenet sites.

DNS

Used for servers that accept DNS traffic, typically nameservers. With the DNS service type, the NetScaler appliance validates the packet format of each DNS request and response. It can also cache DNS responses. You can apply DNS policies to DNS services.

You can also use the UDP service type for these services. If you do, however, the NetScaler appliance can only perform Layer 4 load balancing. It cannot provide support for DNS-specific features.

ANY

Used for servers that accept any type of TCP, UDP, or ICMP traffic. The ANY parameter is used primarily with firewall load balancing and link load balancing.

SIP-UDP

Used for servers that accept UDP-based Session Initiation Protocol (SIP) traffic. SIP initiates, manages, and terminates multimedia communications sessions, and has emerged as the standard for Internet telephony (VoIP).

You can also use the UDP service type for these services. If you do, however, the NetScaler appliance performs only Layer 4 load balancing. It cannot provide support for SIP-specific features.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

DNS-TCP

Used for servers that accept DNS traffic, where the NetScaler appliance acts as a proxy for TCP traffic sent to DNS servers. With services of the DNS-TCP service type, the NetScaler appliance validates the packet format of each DNS request and response and can cache DNS responses, just as with the DNS service type.

You can also use the TCP service type for these services. If you do, however, the NetScaler appliance only performs Layer 4 load balancing of external DNS name servers. It cannot provide support for any DNS-specific features.

RTSP

Used for servers that accept Real Time Streaming Protocol (RTSP) traffic. RTSP provides delivery of multimedia and other streaming data. Select this type to support audio, video, and other types of streamed media.

You can also use the TCP service type for these services. If you do, however, the NetScaler appliance performs only Layer 4 load balancing. It cannot parse the RTSP stream or provide support for RTSPID persistence or RTSP NATting.

DHCPRA

Used for servers that accept DHCP traffic. The DHCPRA service type can be used to relay DHCP requests and responses between VLANs.

DIAMETER

Used for load balancing Diameter traffic among multiple Diameter servers. Diameter uses message-based load balancing.

SSL_DIAMETER

Used for load balancing Diameter traffic over SSL.

Services are designated as DISABLED until the NetScaler appliance connects to the associated load-balanced server and verifies that it is operational. At that point, the service is designated as ENABLED. Thereafter, the NetScaler appliance periodically monitors the status of the servers, and places any that fail to respond to monitoring probes (called health checks) back in the DISABLED state until they respond.

Note: You can create a range of services from a single CLI command or the same dialog box. The names in the range vary by a number used as a suffix/prefix. For example, service1, service2, and so on. From the configuration utility, you can specify a range only in the last octet of the IP address, which is the fourth in case of an IPv4 address and the eighth in case of an IPv6 address. From the command line, you can specify the range in any octet of the IP address.

To create a service by using the command line interface

At the command prompt, type:

add service <name> <serverName> <serviceType> <port>

Example COPY

```
add service Service-HTTP-1 web_serv HTTP 80
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To create a service by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.

2. In the details pane, click **Add**.

3. In the Create Service dialog box, specify values for the following parameters:

- Service Name—name
- Server—serverName
- Protocol—serviceType
- Port—port

4. Click **Create**, and then click **Close**. The service you created appears in the Services pane.

Creating a Virtual Server

After you create your services, you must create a virtual server to accept traffic for the load balanced Web sites, applications, or servers. Once load balancing is configured, users connect to the load-balanced Web site, application, or server through the virtual server's IP address or FQDN.

Note: The virtual server is designated as DOWN until you bind the services that you created to it, and until the NetScaler appliance connects to those services and verifies that they are operational. Only then is the virtual server designated as UP.

To create a virtual server by using the command line interface

At the command prompt, type:

add lb vserver <name> <serviceType> <ip> <port>

Example COPY

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
```

To create a virtual server by using the configuration utility

Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and then create a virtual server.

Binding Services to the Virtual Server

Note: A service can be bound to a maximum of 500 virtual servers.

After you have created services and a virtual server, you must bind the services to the virtual server. In most cases, services are bound to virtual servers of the same type, but you can bind certain types of services to certain different types of virtual servers, as shown below.

Virtual Server Type	Service Type	Comment

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Virtual Server Type	Service Type	Comment
HTTP	SSL	You would normally bind an SSL service to an HTTP virtual server to do encryption.
SSL	HTTP	You would normally bind an HTTP service to an SSL virtual server to do SSL offloading.
SSL_TCP	TCP	You would normally bind a TCP service to an SSL_TCP virtual server to do SSL offloading for other TCP (SSL decryption without content awareness).

The state of the services bound to a virtual server determines the state of the virtual server: if all of the bound services are DOWN, the virtual server is marked DOWN, and if any of the bound services is UP or OUT OF SERVICE, the state of the virtual server is UP.

To bind a service to a load balancing virtual server by using the command line interface

At the command prompt, type:

bind lb vserver <name> <serviceName>

Example COPY

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind a service to a load balancing virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and select a virtual server.
2. Click in the **Service** section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Verifying the Configuration

After finishing your basic configuration, you should view the properties of each service and load balancing virtual server in your load balancing setup to verify that each is configured correctly. After the configuration is up and running, you should view the statistics for each service and load balancing virtual server to check for possible problems.

Viewing the Properties of a Server Object

You can view properties such as the name, state, and IP address of any server object in your NetScaler appliance configuration.

To view the properties of server objects by using the command line interface

At the command prompt, type:

show server <serverName>

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
show server server-1
```

To view the properties of server objects by using the configuration utility

Navigate to **Traffic Management** > **Load Balancing** > **Servers**. The parameter values of the available servers appear in the details pane.

Viewing the Properties of a Virtual Server

You can view properties such as the name, state, effective state, IP address, port, protocol, method, and number of bound services for your virtual servers. If you have configured more than the basic load balancing settings, you can view the persistence settings for your virtual servers, any policies that are bound to them, and any cache redirection and content switching virtual servers that have been bound to the virtual servers.

To view the properties of a load balancing virtual server by using the command line interface

At the command prompt, type:

show lb vserver <name>

Example COPY

```
show lb vserver Vserver-LB-1
```

To view the properties of a load balancing virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.

2. In the details pane, click a virtual server to display its properties at the bottom of the

details pane.

3. To view cache redirection and content switching virtual servers that are bound to this virtual server, click **Show CS/CR Bindings**.

Viewing the Properties of a Service

You can view the name, state, IP address, port, protocol, maximum client connection, maximum requests per connection, and server type of the configured services, and use this information to troubleshoot any mistake in the service configuration.

To view the properties of services by using the command line interface

At the command prompt, type:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

show service <name>

Example COPY

```
show service Service-HTTP-1
```

To view the properties of services by using the configuration utility

Navigate to **Traffic Management** > **Load Balancing** > **Services**. The details of the available services appear on the Services pane.

Viewing the Bindings of a Service

You can view the list of virtual servers to which the service is bound. The binding information also provides the name, IP address, port and state of the virtual servers to which the services are bound. You can use the binding information to troubleshoot any problem with binding the services to virtual servers.

To view the bindings of a service by using the command line

At the command prompt, type:

show service bindings <name>

Example COPY

```
show service bindings Service-HTTP-1
```

To view the bindings of a service by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.

2. In the details pane, select the service whose binding information you want to view.

3. In the **Action** tab, click **Show Bindings**.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the command prompt, type:

stat lb vserver [-detail] [<name>]

Example COPY

```
>stat lb vserver -detail

Virtual Server(s) Summary

vsvrIP port Protocol State Req/s Hits/s

One * 80 HTTP UP 5/s 0/s

Two * 0 TCP DOWN 0/s 0/s

Three * 2598 TCP DOWN 0/s 0/s

dnsVirtualNS 10.102.29.90 53 DNS DOWN 0/s 0/s

BRVSERV 10.10.1.1 80 HTTP DOWN 0/s 0/s

LBVIP 10.102.29.66 80 HTTP UP 0/s 0/s

Done
```

To display virtual server statistics by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.

2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server whose statistics you want to display.

3. In the details pane, click **Statistics**.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the command line interface

At the command prompt, type:

stat service <name>

Example COPY

```
stat service Service-HTTP-1
```

To view the statistics of a service by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.

2. In the details pane, select the service whose statistics you want to view (for example, Service-HTTP-1).

3. Click **Statistics**. The statistics appear in a new window.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing Virtual Server and Service States

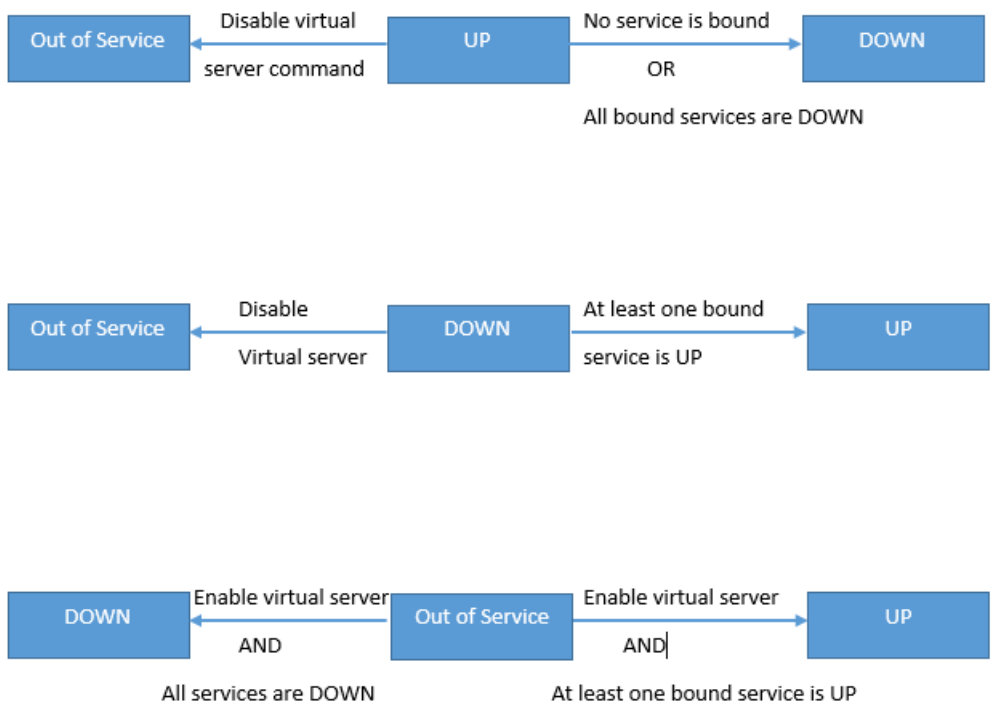
Jul 28, 2016

A load balancing virtual server that does not have a backup virtual server can take the following states, depending on the states of the service(s) bound to it and whether it is administratively disabled:

- **UP**: At least one of the services bound to the virtual server is UP.
- **DOWN**: All the services bound to the virtual server are DOWN, or the load balancing feature is not enabled.
- **Out of Service (OFS)**: If you administratively disable the virtual server, it enters the OFS state but its effective state is DOWN. Transitioning to the OFS state from the DOWN or UP state, or to the DOWN or UP state from the OFS state, is controlled by the administrator.

The state and effective state of a virtual server are the same if a backup virtual server is not configured. However, if a backup virtual server or a chain of backup virtual servers is configured, the effective state is derived from the states of the services that are bound to the primary virtual server and the backup virtual server(s). If any of the backup virtual servers in the chain is UP, the effective state of the primary virtual server is UP, even if all the services bound to the primary virtual server are DOWN.

The following diagrams show the conditions under which a virtual server transitions from one state to another.



A service can take the following states:

- **UP:** If probes from all the monitors bound to the service are successful.
- **DOWN**: If monitoring probes to the service are not answered within the configured time limit.
- **OUT OF SERVICE**: If you administratively disable the service, or if you gracefully shut down the service and there are no active transactions to the service

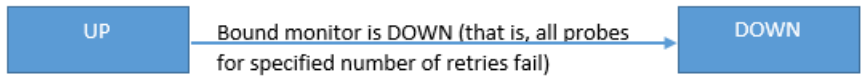
© 1999-2017 Citrix Systems, Inc. All rights reserved.

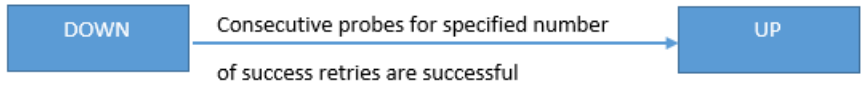
- **GOING OUT OF SERVICE (TROFS)**: If you administratively disable the service with delay, or gracefully shut down the service and there are active transactions to the service. For more information, see http://docs.citrix.com/en-us/netscaler/11/traffic-management/load-balancing/load-balancing-advanced-settings/graceful-shutdown.html.
- **DOWN WHEN GOING OUT OF SERVICE (TROFS_DOWN)**: A monitoring probe fails while the service is in the GOING OUT OF SERVICE state.

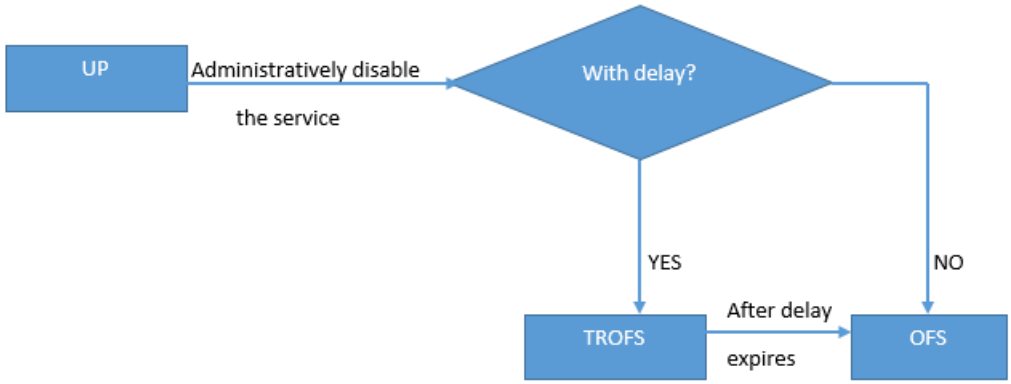
A service in the process of transitioning from UP to OFS is in the GOING OUT OF SERVICE state. A service transitioning from DOWN to OFS is in the DOWN WHEN GOING OUT OF SERVICE state. For example, if a service is DOWN and you disable it with delay, the service transitions to DOWN WHEN GOING OUT OF SERVICE and then to the OUT OF SERVICE state. If a service is UP and you disable it with delay, the service transitions to GOING OUT OF SERVICE. During this time, if a monitoring probe to the server fails, the service transitions to DOWN WHEN GOING OUT OF SERVICE and, after the delay time expires, enters the OFS state.

> ## Note
>
> You can configure spillover to a backup virtual server by setting the "healthThreshold" parameter to a non-zero positive value. Then, if a single service bound to the primary virtual server transitions to the DOWN WHEN GOING OUT OF SERVICE state and the health threshold is not reached, the primary virtual server is marked DOWN and new connections are directed to the backup virtual server.

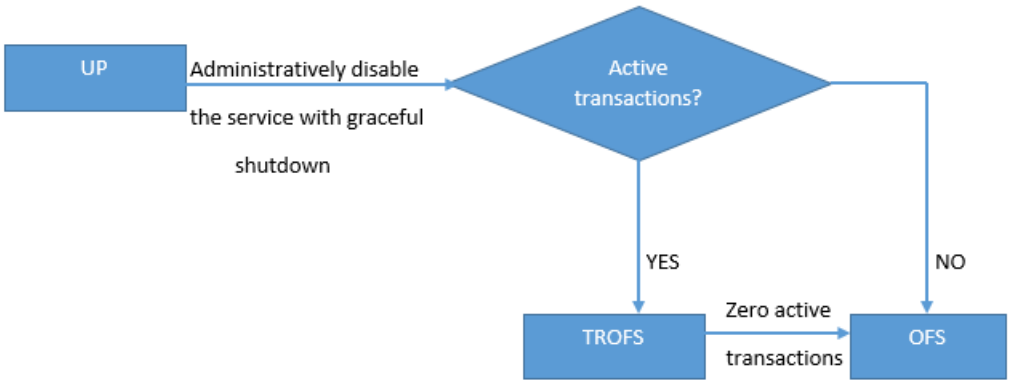
The following diagrams show the conditions under which a service transitions from one state to another.

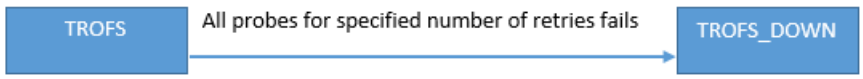






© 1999-2017 Citrix Systems, Inc. All rights reserved.





© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing Algorithms

Aug 29, 2013

The load balancing algorithm defines the criteria that the NetScaler appliance uses to select the service to which to redirect each client request. Different load balancing algorithms use different criteria. For example, the least connection algorithm selects the service with the fewest active connections, while the round robin algorithm maintains a running queue of active services, distributes each connection to the next service in the queue, and then sends that service to the end of the queue.

Some load balancing algorithms are best suited to handling traffic on websites, others to managing traffic to DNS servers, and others to handling complex web applications used in e-commerce or on company LANs or WANs. The following table lists each load balancing algorithm that the NetScaler appliance supports, with a brief description of how each operates.

Name	Server Selection Based On
LEASTCONNECTION	Which service currently has the fewest client connections. This is the default load balancing algorithm.
ROUNDROBIN	Which service is at the top of a list of services. After that service is selected for a connection, it moves to the bottom of the list.
LEASTRESPONSETIME	Which load balanced server currently has the quickest response time.
URLHASH	A hash of the destination URL.
DOMAINHASH	A hash of the destination domain.
DESTINATIONIPHASH	A hash of the destination IP address.
SOURCEIPHASH	A hash of the source IP address.
SRCIPDESTIPHASH	A hash of the source and destination IP addresses.
CALLIDHASH	A hash of the call ID in the SIP header.
SRCIPSRCPORTHASH	A hash of the client's IP address and port.
LEASTBANDWIDTH	Which service currently has the fewest bandwidth constraints.
LEASTPACKETS	Which service currently is receiving the fewest packets.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Server Selection Based On
CUSTOMLOAD	Data from a load monitor.
TOKEN	The configured token.
LRTM	Fewest active connections and the lowest average response time.

Depending on the protocol of the service that it is load balancing, the NetScaler appliance sets up each connection between client and server to last for a different time interval. This is called load balancing granularity, of which are three types: request-based, connection-based, and time-based granularity. The following table describes each type of granularity and when each is used.

Granularity	Types of Load Balanced Service	Specifies
Request - based	HTTP or HTTPS	A new service is chosen for each HTTP request, independent of TCP connections. As with all HTTP requests, after the Web server fulfills the request, the connection is closed.
Connection-based	TCP and TCP-based protocols other than HTTP	A service is chosen for every new TCP connection. The connection persists until terminated by either the service or the client.
Time-based	UDP and other IP protocols	A new service is chosen for each UDP packet. Upon selection of a service, a session is created between the service and a client for a specified period of time. When the time expires, the session is deleted and a new service is chosen for any additional packets, even if those packets come from the same client.

During startup of a virtual server, or whenever the state of a virtual server changes, the virtual server can initially use the round robin method to distribute the client requests among the physical servers. This type of distribution, referred to as *startup round robin,* helps prevent unnecessary load on a single server as the initial requests are served. After using the round robin method at the startup, the virtual server switches to the load balancing method specified on the virtual server.

The Startup RR Factor works in the following manner:

- If the Startup RR Factor is set to zero, the NetScaler switches to the specified load balancing method depending on the request rate.
- If the Startup RR Factor is any number other than zero, NetScaler uses the round robin method for the specified number of requests before switching to the specified load balancing method.
- By default, the Startup RR Factor is set to zero.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: You cannot set the startup RR Factor for an individual virtual server. The value you specify applies to all the virtual servers on the NetScaler appliance.

To set the startup round-robin factor by using the command line interface

At the command prompt, type:

set lb parameter -startupRRFactor <positive_integer>

Example

set lb parameter -startupRRFactor 25000

To set the startup round-robin factor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing Parameters, and set the Startup RR Factor.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The Least Connection Method

Nov 07, 2017

When a virtual server is configured to use the least connection load balancing algorithm (or method), it selects the service with the fewest active connections. This is the default method, because, in most circumstances, it provides the best performance.

For TCP, HTTP, HTTPS, and SSL_TCP services, the NetScaler appliance includes the following connection types in its list of existing connections:

- **Active connections to a service.** Connections representing requests that a client has sent to the virtual server and that the virtual server has forwarded to a service. For HTTP and HTTPS services, active connections represent only those HTTP or HTTPS requests that have not yet received a response.
- **Waiting connections in the surge queue.** Any connections to the virtual server that are waiting in a surge queue and have not yet been forwarded to a service. Connections can build up in the surge queue at any time, for any of the following reasons:
  - Your services have connection limits, and all services in your load balancing configuration are at that limit.
  - The surge protection feature is configured and has been activated by a surge in requests to the virtual server.
  - The load-balanced server has reached an internal limit and therefore does not open any new connections. (For example, an Apache server's connection limit is reached.)

When a virtual server uses the least connection method, it considers the waiting connections as belonging to the specific service. Therefore, it does not open new connections to those services.

For UDP services, the connections that the least connection algorithm considers include all sessions between the client and a service. These sessions are logical, time-based entities. When the first UDP packet in a session arrives, the NetScaler appliance creates a session between the source IP address and port and the destination IP address and port.

For Real-Time Streaming Protocol (RTSP) connections, the NetScaler appliance uses the number of active control connections to determine the lowest number of connections to an RTSP service.

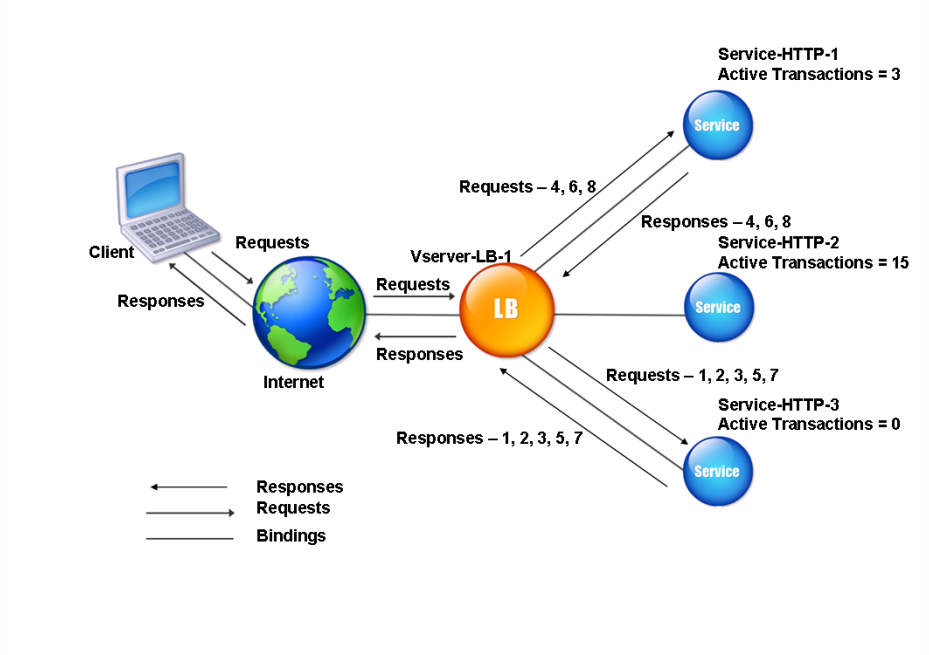
The following example shows how a virtual server selects a service for load balancing by using the least connection method. Consider the following three services:

- Service-HTTP-1 is handling 3 active transactions.
- Service-HTTP-2 is handling 15 active transactions.
- Service-HTTP-3 is not handling any active transactions.

The following diagram illustrates how the NetScaler appliance forwards incoming requests when using the least connection method.

Figure 1. Mechanism of the Least Connections Load Balancing Method

© 1999-2017 Citrix Systems, Inc. All rights reserved.



In this diagram, the virtual server selects the service for each incoming connection by choosing the server with the fewest active transactions.

Connections are forwarded as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.

  Note: The service with no active transaction is selected first.
- Service-HTTP-3 receives the second and third requests because the service has the next least number of active transactions.
- Service-HTTP-1 receives the fourth request Because Service-HTTP-1 and Service-HTTP-3 have same number of active transactions, the virtual server uses the round robin method to choose between them.
- Service-HTTP-3 receives the fifth request.
- Service-HTTP-1 receives the sixth request, and so on, until both Service-HTTP-1 and Service-HTTP-3 are handling the same number of requests as Service-HTTP-2. At that time, the NetScaler appliance starts forwarding requests to Service-HTTP-2 when it is the least loaded service or its turn comes up in the round robin queue.

  Note: If connections to Service-HTTP-2 close, it might get new connections before each of the other two services has 15 active transactions.

The following table explains how connections are distributed in the three-service load balancing setup described above.

Incoming Connection	Service Selected	Current Number of Active Connections	Remarks
Request-1	Service-HTTP-3 (N = 0)	1	Service-HTTP-3 has the fewest active connections.
Request-2	Service-	2	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Incoming Connection	Service Selected	Current Number of Active Connections	Remarks
	HTTP-3 (N = 1)		
Request-3	Service-HTTP-3 (N = 2)	3	
Request-4	Service-HTTP-1 (N = 3)	4	Service-HTTP-1 and Service-HTTP-3 have the same number of active connections.
Request-5	Service-HTTP-3 (N = 3)	4	
Request-6	Service-HTTP-1 (N = 4)	5	
Request-7	Service-HTTP-3 (N = 4)	5	
Request-8	Service-HTTP-1 (N = 5)	6	
Service-HTTP-2 is selected for load balancing when it completes its active transactions and the current connections to it close, or when the other services (Service-HTTP-1 and Service-HTTP-3) have 15 or more connections each.			

The NetScaler appliance can also use the least connection method when weights are assigned to services. It selects a service by using the value (Nw) of the following expression:

Nw = (Number of active transactions) * (10000 / weight)

The following example shows how the NetScaler appliance selects a service for load balancing by using the least connection method when weights are assigned to services. In the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. Connections are forwarded as follows:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Service-HTTP-3 receives the first because the service is not handling any active transactions.

  Note: If services are not handling any active transactions, the NetScaler appliance uses the round robin method regardless of the weights assigned to each of the services.
- Service-HTTP-3 receives the second, third, fourth, fifth, sixth, and seventh requests because the service has lowest Nw value.
- Service-HTTP-1 receives the eighth request. Because Service-HTTP-1 and Service-HTTP-3 now have same Nw value, the NetScaler performs load balancing in a round robin manner. Therefore, Service-HTTP-3 receives the ninth request.

The following table explains how connections are distributed on the three-service load balancing setup that is described above.

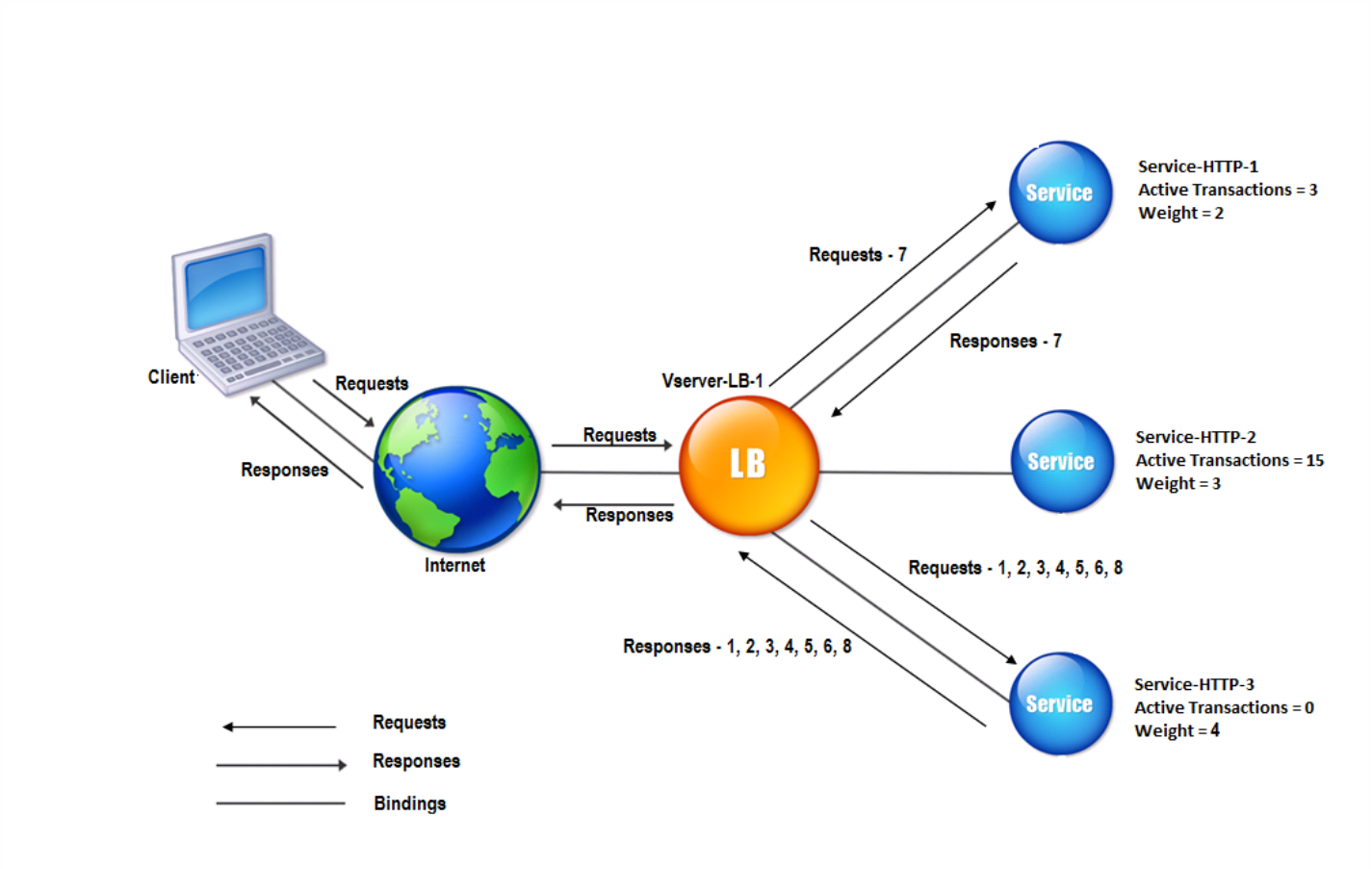
Request Received	**Service Selected**	**Current Nw (Number of active transactions) * (10000 / weight) value**	**Remarks**
Request-1	Service-HTTP-3 (Nw = 0)	Nw = 2500	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3 (Nw = 2500)	Nw = 5000	
Request-3	Service-HTTP-3 (Nw = 5000)	Nw = 7500	
Request-4	Service-HTTP-3 (Nw = 7500)	Nw = 10000	
Request-5	Service-HTTP-3 (Nw = 10000)	Nw = 12500	
Request-6	Service-HTTP-3	Nw = 15000	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Received	Service Selected	Current Nw (Number of active transactions) * (10000 / weight) value	Remarks
Request-7	Service-HTTP-1 (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw values
Request-8	Service-HTTP-3 (Nw = 15000)	Nw = 17500	
Service-HTTP-2 is selected for load balancing when it completes its active transactions or when the Nw value of other services (Service-HTTP-1 and Service-HTTP-3) is equal to 50000.			

The following diagram illustrates how the NetScaler appliance uses the least connection method when weights are assigned to the services.

Figure 2. Mechanism of the Least Connections Load Balancing Method when Weights are Assigned



© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure the least connection method, see Configuring a Load Balancing Method that Does Not Include a Policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

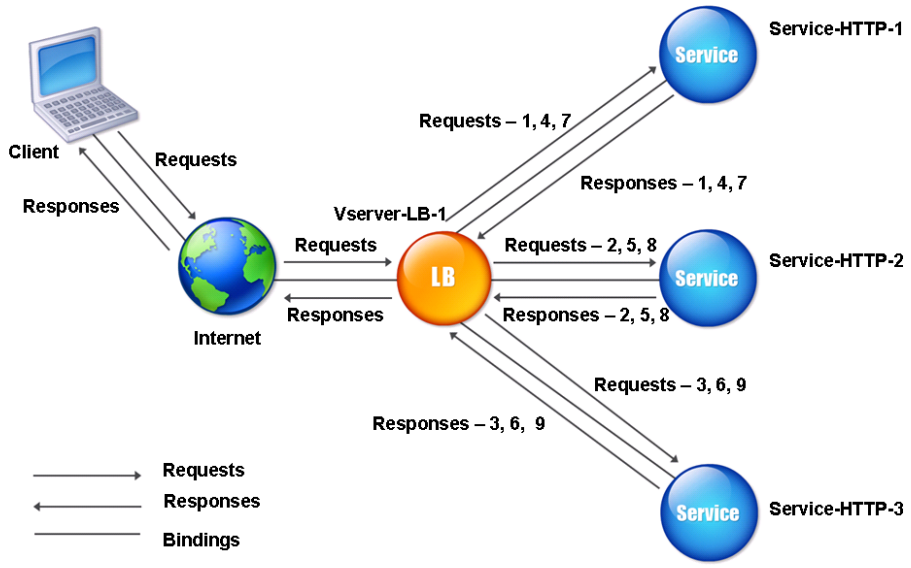
The Round Robin Method

Nov 07, 2017

When a load balancing virtual server is configured to use the round robin method, it continuously rotates a list of the services that are bound to it. When the virtual server receives a request, it assigns the connection to the first service in the list, and then moves that service to the bottom of the list.

The following diagram illustrates how the NetScaler appliance uses the round robin method with a load balancing setup that contains three load-balanced servers and their associated services.

Figure 1. How the Round Robin Load Balancing Method Works



If you assign a different weight to each service, the NetScaler appliance performs weighted round robin distribution of incoming connections. It does this by skipping the lower-weighted services at appropriate intervals.

For example, assume that you have a load balancing setup with three services. You set Service-HTTP-1 to a weight of 2, Service-HTTP-2 to a weight of 3, and Service-HTTP-3 to a weight of 4. The services are bound to Vserver-LB-1, which is configured to use the round robin method. With this setup, incoming requests are delivered as follows:

- Service-HTTP-1 receives the first request.
- Service-HTTP-2 receives the second request.
- Service-HTTP-3 receives the third request.
- Service-HTTP-1 receives the fourth request.
- Service-HTTP-2 receives the fifth request.
- Service-HTTP-3 receives the sixth request.
- Service-HTTP-2 receives the seventh request.
- Service-HTTP-3 receives both the eighth and the ninth requests.

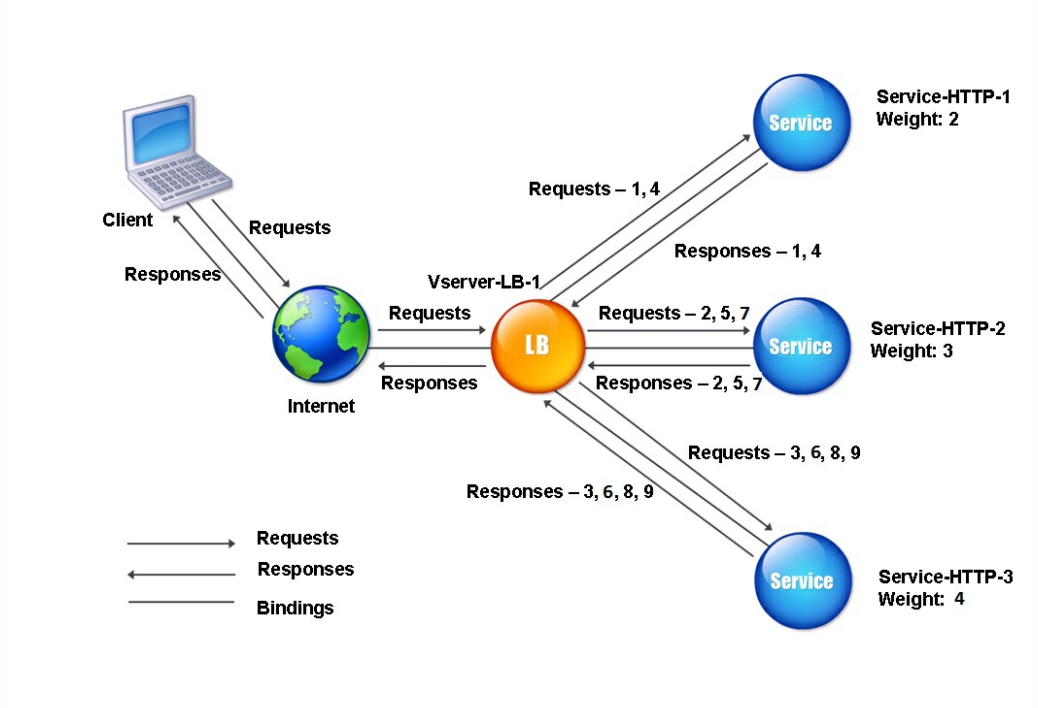
Note: You can also configure weights on services to prevent multiple services from using the same server and overloading

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the server.
A new cycle then begins, using the same pattern.

The following diagram illustrates the weighted round robin method.

Figure 2. How the Round Robin Load Balancing Method Works with Weighted Services



To configure the round robin method, see Configuring a Load Balancing Method that Does Not Include a Policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The Least Response Time Method

Nov 07, 2017

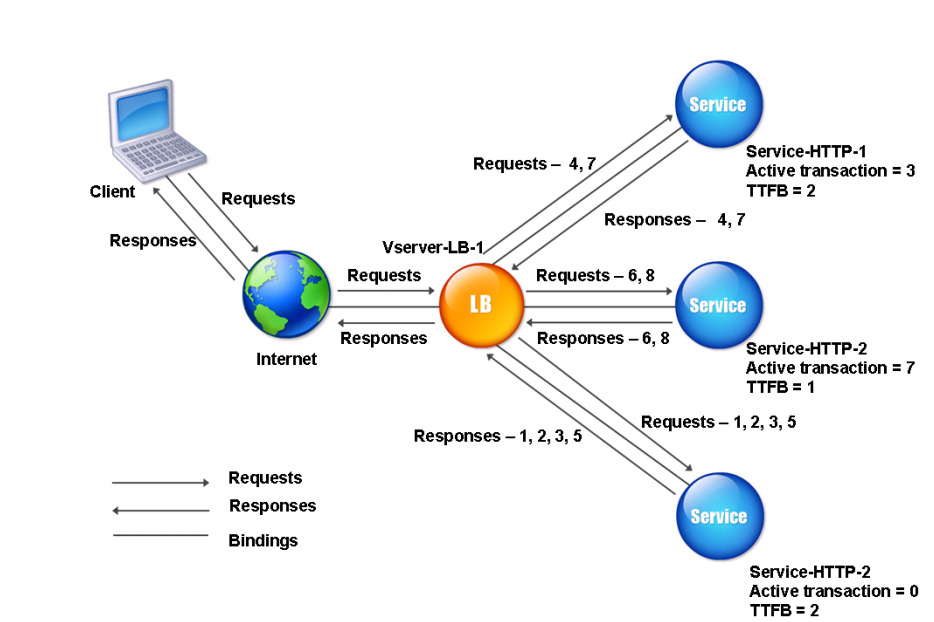
When the load balancing virtual server is configured to use the least response time method, it selects the service with the fewest active connections and the lowest average response time. You can configure this method for HTTP and Secure Sockets Layer (SSL) services only. The response time (also called Time to First Byte, or TTFB) is the time interval between sending a request packet to a service and receiving the first response packet from the service. The NetScaler appliance uses response code 200 to calculate TTFB.

The following example shows how a virtual server selects a service for load balancing by using the least response time method. Consider the following three services:

- Service-HTTP-1 is handling three active transactions and TTFB is two seconds.
- Service-HTTP-2 is handling seven active transactions and TTFB is one second.
- Service-HTTP-3 is not handling any active transactions and TTFB is two seconds.

The following diagram illustrates how the NetScaler appliance uses the least response time method to forward the connections.

Figure 1. How the Least Response Time Load Balancing Method Works



The virtual server selects a service by multiplying the number of active transactions by the TTFB for each service and then selecting the service with the lowest result. For the example shown above, the virtual server forwards requests as follows:

- Service-HTTP-3 receives the first request, because the service is not handling any active transactions.
- Service-HTTP-3 also receives the second and third requests, because the result is lowest of the three services.
- Service-HTTP-1 receives the fourth request. Since Service-HTTP-1 and Service-HTTP-3 have the same result, the NetScaler appliance chooses between them by applying the Round Robin method.
- Service-HTTP-3 receives the fifth request.
- Service-HTTP-2 receives the sixth request, because at this point it has the lowest result.
- Because Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all have the same result at this point, the NetScaler

© 1999-2017 Citrix Systems, Inc. All rights reserved.

switches to the round robin method, and continues to distribute connections using that method.

The following table explains how connections are distributed in the three-service load balancing setup described above.

Request Received	Service Selected	Current N Value (Number of Active Transactions * TTFB)	Remarks
Request-1	Service-HTTP-3 (N = 0)	N = 2	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-3 (N = 2)	N = 4	
Request-3	Service-HTTP-3 (N = 3)	N = 6	
Request-4	Service-HTTP-1 (N = 6)	N = 8	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-5	Service-HTTP-3 (N = 6)	N = 8	
Request-6	Service-HTTP-2 (N = 7)	N = 8	Service-HTTP-2 has the lowest N value.
Request-7	Service-HTTP-1 (N = 8)	N = 15	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-8	Service-HTTP-2 (N = 8)	N = 9	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Received	Service Selected	Current N Value (Number of Active Transactions * TFB)	Remarks

The virtual server selects a service based on the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

Suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned weight of 3, and Service-HTTP-3 is assigned weight of 4.

The NetScaler appliance distributes requests as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.

  If services are not handling any active transactions, the NetScaler selects them regardless of the weights assigned to them.

- Service-HTTP-3 receives the second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

Service-HTTP-1 has the lowest weight and therefore the highest Nw value, so the virtual server does not select it for load balancing.

The following table explains how connections are distributed in the three-service load balancing setup described above.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3 (Nw = 0)	Nw = 2500	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3 (Nw = 2500	Nw = 5000	
Request-3	Service-HTTP-3 (Nw = 5000)	Nw = 15000	
Request-4	Service-HTTP-3 (Nw = 15000)	Nw = 20000	

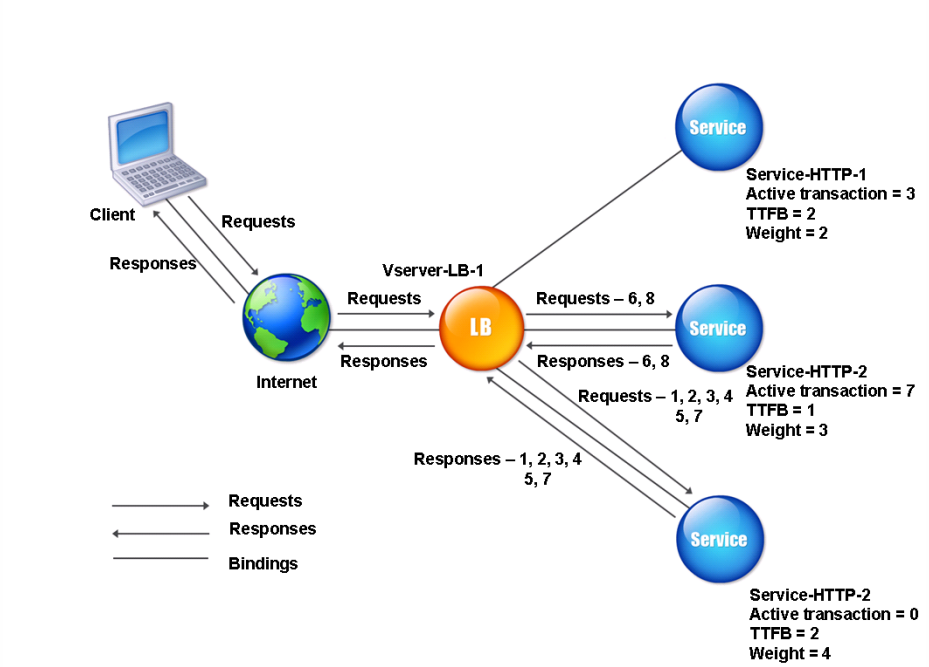
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-5	Service-HTTP-3 (Nw = 20000)	Nw = 25000	
Request-6	Service-HTTP-2 (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-7	Service-HTTP-3 (Nw = 25000)	Nw= 30000	Service-HTTP-3 has the lowest Nw value.
Request-8	Service-HTTP-2 (Nw = 26666.67)	Nw = 33333.34	Service-HTTP-2 has the lowest Nw value.
Service-HTTP-1 is selected for load balancing when it completes its active transactions or when the Nw values of other services (Service-HTTP-2 and Service-HTTP-3) are equal to 105000.			

The following diagram illustrates how the NetScaler appliance uses the least response time method when weights are assigned.

Figure 2. How the Least Response Time Load Balancing Method Works When Weights Are Assigned

© 1999-2017 Citrix Systems, Inc. All rights reserved.



To configure the least response time method, see Configuring a Load Balancing Method that Does Not Include a Policy.

When a load balancing virtual server is configured to use the least response time method with monitors, it uses the existing monitoring infrastructure to select the service with the smallest number of active transactions and the fastest average response time. Before you use the least response time method with monitoring, you must bind application-specific monitors to each service and enable least response time method mode on these monitors. The NetScaler appliance then makes load balancing decisions based on the response times it calculates from monitoring probes. For more information about configuring monitors, see Configuring Monitors in a Load Balancing Setup.

You can use the least response time method with monitors to select non-HTTP and non-HTTPS services. You can also use this method when several monitors are bound to a service. Each monitor determines the response time by using the protocol that it measures for the service that it is bound to. The virtual server then calculates an average response time for that service by averaging the results.

The following table summarizes how response times are calculated for various monitors.

Monitor	Response Time Calculation
PING	Time difference between the ICMP ECHO request and the ICMP ECHO response.
TCP	Time difference between the SYN request and the SYN+ACK response.
HTTP	Time difference between the HTTP request (after the TCP connection is established) and the HTTP response.
TCP-ECV	Time difference between the time the data send string is sent and the data receive string is returned. A tcp-ecv monitor without the send and receive strings is considered to have an incorrect

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitor	Response Time Calculation
HTTP-ECV	Time difference between the HTTP request and the HTTP response.
UDP-ECV	Time difference between the UDP send string and the UDP receive string. A udp-ecv monitor without the receive string is considered to have an incorrect configuration.
DNS	Time difference between a DNS query and the DNS response.
TCPS	Time difference between a SYN request and the SSL handshake completion.
FTP	Time difference between the sending of the user name and the completion of user authentication.
HTTPS (monitors HTTPS requests)	Time difference is same as for the HTTP monitor.
HTTPS-ECV (monitors HTTPS requests)	Time difference is same as for the HTTP-ECV monitor
USER	Time difference between the time when a request is sent to the dispatcher and the time when the dispatcher response is received.

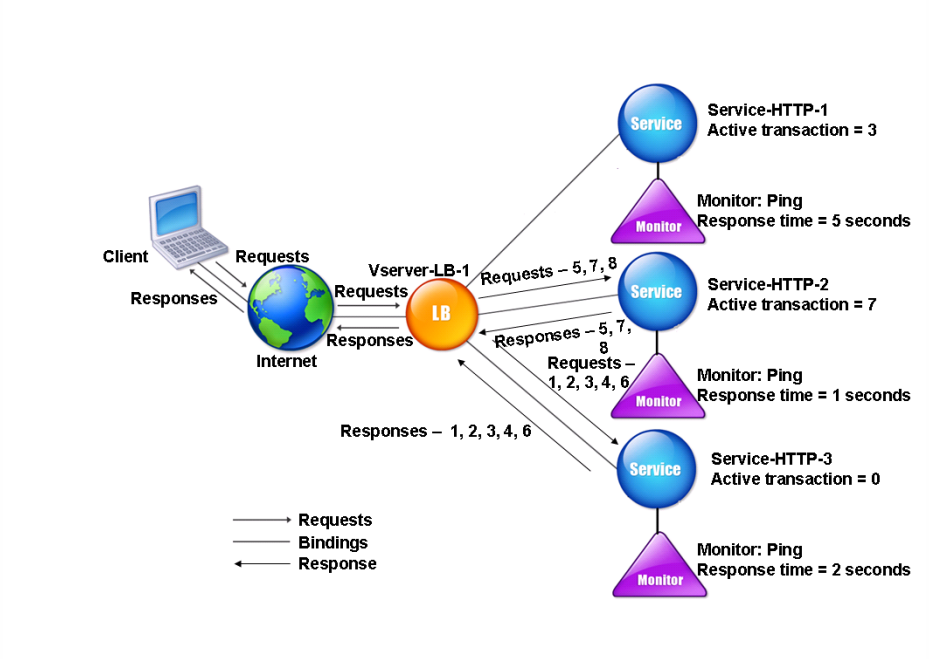
The following example shows how the NetScaler appliance selects a service for load balancing by using the least response time method with monitors. Consider the following three services:

- Service-HTTP-1 is handling 3 active transactions and the response time is five seconds.
- Service-HTTP-2 is handling 7 active transactions and the response time is one second.
- Service-HTTP-3 is not handling any active transactions and the response time is two seconds.

The following diagram illustrates the process that the NetScaler appliance follows when it forwards requests.

Figure 3. How the Least Response Time Load Balancing Method Works When Using Monitors

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The virtual server selects a service by using the value (N) in the following expression:

N = Number of active transactions * Response time that is determined by the monitor

The virtual server delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service is not handling any active transaction.
- Service-HTTP-3 receives the second, third, and fourth requests, because this service has the lowest N value.
- Service-HTTP-2 receives the fifth request, because this service has the lowest N value.
- Since both Service-HTTP-2 and Service-HTTP-3 currently have the same N value, the NetScaler appliance switches to the round robin method Therefore, Service-HTTP-3 receives the sixth request.
- Service-HTTP-2 receives the seventh and eighth requests, because this service has the lowest N value.

Service-HTTP-1 is not considered for load balancing, because it is more heavily loaded (has the highest N value) when compared to the other two services. However, if Service-HTTP-1 completes its active transactions, the NetScaler appliance again considers that service for load balancing.

The following table summarizes how N is calculated for the services.

Request Received	Service Selected	Current N Value (Number of Active Transactions)	Remarks
Request-1	Service-HTTP-3 (N = 0)	N = 2	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-3 (N = 2)	N = 4	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Received	Service Selected	Current N Value (Number of Active Transactions)	Remarks
Request-3	Service-HTTP-3 (N = 4)	N = 6	
Request-4	Service-HTTP-3 (N = 6)	N = 8	
Request-5	Service-HTTP-2 (N = 7)	N = 8	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-6	Service-HTTP-3 (N = 8)	N = 10	
Request-7	Service-HTTP-2 (N = 8)	N = 9	Service-HTTP-2 has the lowest N value.
Request-8	Service-HTTP-1 (N = 9)	N = 10	
Service-HTTP-1 is again selected for load balancing when it completes its active transactions or when the N value of the other services (Service-HTTP-2 and Service-HTTP-3) is equal to 15.			

The NetScaler appliance also performs load balancing by using the number of active transactions, response time, and weights if different weights are assigned to services. The NetScaler appliance selects the service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4.

The NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.
- Service-HTTP-3 receives the second, third, and fourth requests, because this service has the lowest Nw value.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Service-HTTP-2 receives the fifth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the seventh and the eighth requests, because this service has the lowest Nw value.

Service-HTTP-1 has the lowest weight and the highest Nw value, so the NetScaler appliance does not select it for load balancing.

The following table summarizes how Nw is calculated for various monitors.

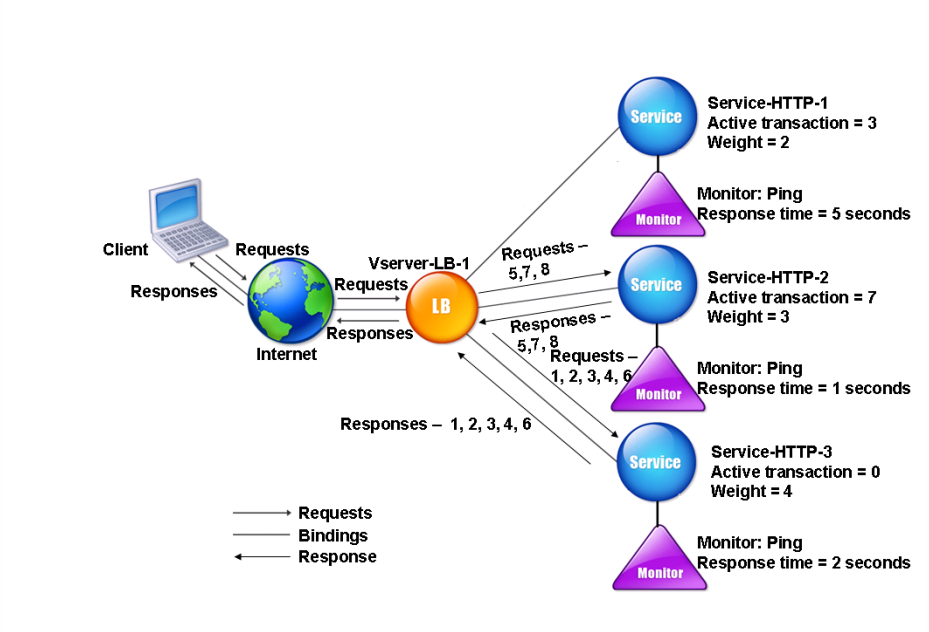
Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3 (Nw = 0)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3 (Nw = 5000	Nw = 10000	
Request-3	Service-HTTP-3 (Nw = 15000)	Nw = 20000	
Request-4	Service-HTTP-3 (Nw = 20000)	Nw = 25000	
Request-5	Service-HTTP-2 (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-6	Service-HTTP-3 (Nw = 25000)	Nw = 30000	Service-HTTP-3 has the lowest Nw value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-7	Service-HTTP-2 (Nw = 23333.34)	Nw= 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-8	Service-HTTP-2 (Nw =25000)	Nw = 30000	Service-HTTP-2 has the lowest Nw value.
Service-HTTP-1 is selected for load balancing when it completes its active transactions or when the Nw value of the other services (Service-HTTP-2 and Service-HTTP-3) is equal to 75000.			

The following diagram illustrates how the virtual server uses the least response time method when weights are assigned.

Figure 4. How the Least Response Time Load Balancing Method with Monitors Works When Weights Are Assigned



To configure the least response time method using monitors, see Configuring a Load Balancing Method that Does Not Include a Policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

About Hashing Methods

Oct 21, 2015

Load balancing methods based on hashes of certain connection information or header information constitute the majority of the NetScaler appliance's load balancing methods. Hashes are shorter and easier to use than the information that they are based on, while retaining enough information to ensure that no two different pieces of information generate the same hash and are therefore confused with one another.

You can use the hashing load balancing methods in an environment where a cache serves a wide range of content from the Internet or specified origin servers. Caching requests reduces request and response latency, and ensures better resource (CPU) utilization, making caching popular on heavily used Web sites and application servers. Since these sites also benefit from load balancing, hashing load balancing methods are widely useful.

The NetScaler provides the following hashing methods:

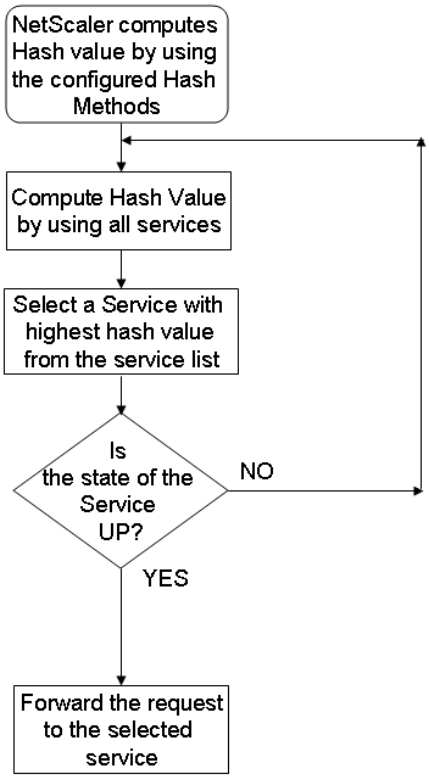
- URL hash method
- Domain hash method
- Destination IP hash method
- Source IP hash method
- Source IP Destination IP hash method
- Source IP Source Port hash method
- Call ID hash method
- Token method

These hashing algorithms ensure minimal disruption when services are added to or deleted from your load balancing setup. Most of them calculate two hash values:

- A hash of the service's IP address and port.
- A hash of the incoming URL, the domain name, the source IP address, the destination IP address, or the source and destination IP addresses, depending on the configured hash method.

The NetScaler appliance then generates a new hash value by using both of those hash values. Finally, it forwards the request to the service with highest hash value. As the appliance computes a hash value for each request and selects the service that will process the request, it populates a cache. Subsequent requests with the same hash value are sent to the same service. The following flow chart illustrates this process.

Figure 1. How the Hashing Methods Distribute Requests

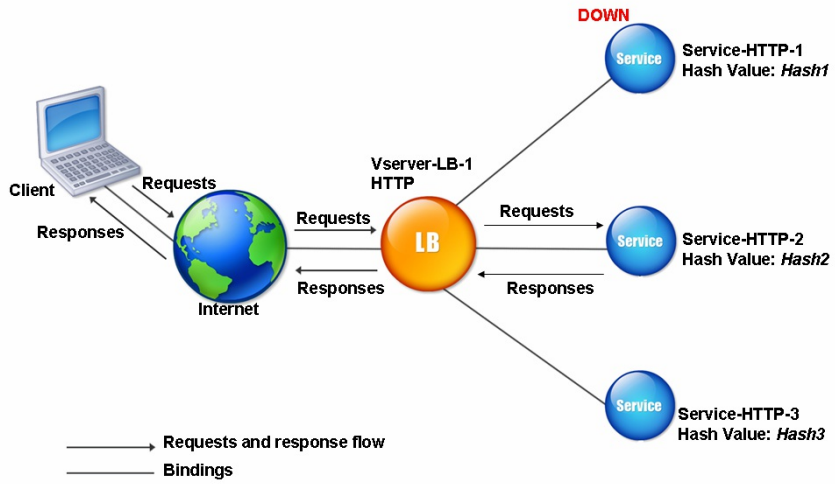


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Hashing methods can be applied to IPv4 and IPv6 addresses.

Consider a scenario where three services (Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3) are bound to a virtual server, any hash method is configured, and the hash value is Hash1. When the configured services are UP, the request is sent to Service-HTTP-1. If Service-HTTP-1 is down, the NetScaler appliance calculates the hash value for the last log of the number of services. The NetScaler then selects the service with the highest hash value, such as Service-HTTP-2. The following diagram illustrates this process.

Figure 2. Entity Model for Hashing Methods



Note: If the NetScaler appliance fails to select a service by using a hashing method, it defaults to the least connection method to select a service for the incoming request. You should adjust server pools by removing services during periods of low traffic to enable the caches to repopulate without affecting performance on your load balancing setup.

The URL Hash Method

When you configure the NetScaler appliance to use the URL hash method for load balancing the services, for selecting a service, the NetScaler generates a hash value of the HTTP URL present in the incoming request. If the service selected by the hash value is DOWN, the algorithm has a method to select another service from the list of active services. The NetScaler caches the hashed value of the URL, and when it receives subsequent requests that use the same URL, it forwards them to the same service. If the NetScaler cannot parse an incoming request, it uses the round robin method for load balancing instead of the URL hash method.

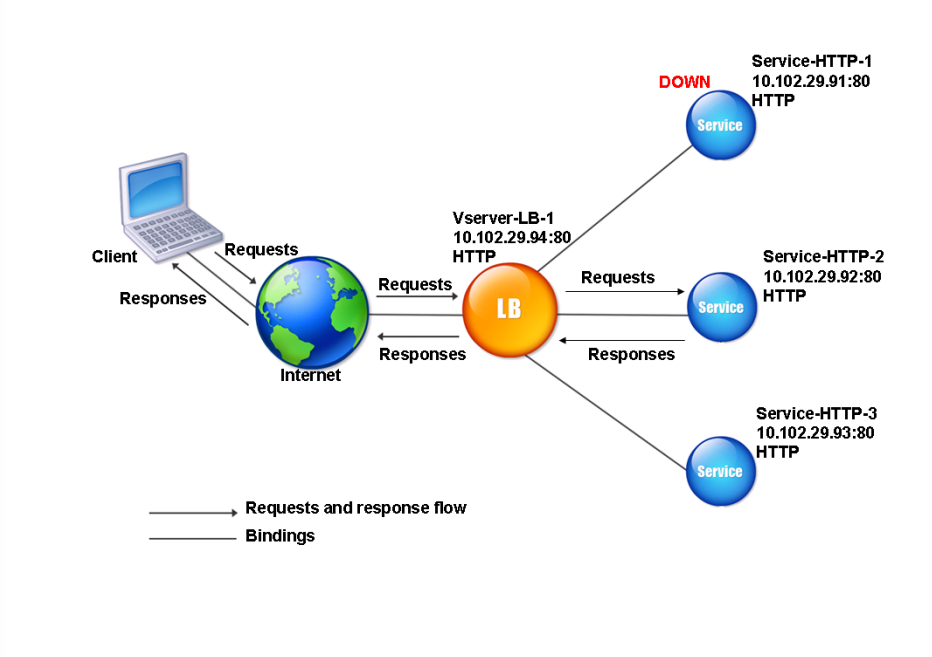
For generating the hash value, NetScaler uses a specific algorithm and considers a part of the URL. By default, the NetScaler considers the first 80 bytes of the URL. If the URL is of less than 80 bytes, the complete URL is used. You can specify a different length. The hash length can be from 1 to 4096 bytes. Generally, if long URLs are used where only a small number of characters are different, it is a good idea to make the hash length as high as possible to try to ensure a more even load distribution.

Consider a scenario where three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3, are bound to a virtual server, and the load balancing method configured on the virtual server is the URL hash method. The virtual server receives a request and the hash value of the URL is U1. NetScaler selects Service-HTTP-1. If Service-HTTP-1 is DOWN, the NetScaler selects Service-HTTP-2.

The following diagram illustrates this process.

Figure 3. How URL Hashing Operates

© 1999-2017 Citrix Systems, Inc. All rights reserved.



If both Service-HTTP-1 and Service-HTTP-2 are DOWN, NetScaler sends requests with hash value U1 to Service-HTTP-3.

If Service-HTTP-1 and Service-HTTP-2 are down, requests that generate the hash URL1 are sent to Service-HTTP-3. If these services are UP, requests that generate the hash URL1 are distributed in the following manner:

- If the Service-HTTP-2 is up, the request is sent to Service-HTTP-2.
- If the Service-HTTP-1 is up, the request is sent to Service-HTTP-1.
- If Service-HTTP-1 and Service-HTTP-2 are up at the same time, the request is sent to Service-HTTP-1.

To configure the URL hash method, see Configuring a Load Balancing Method that Does Not Include a Policy. Select the load balancing method as URL Hash, and set the hash length to the number of bytes to be used for generating the hash value.

The Domain Hash Method

A load balancing virtual server configured to use the domain hash method uses the hashed value of the domain name in the HTTP request to select a service. The domain name is taken from either the incoming URL or the Host header of the HTTP request. If the domain name appears in both the URL and the Host header, the NetScaler gives preference to the URL.

If you configure domain name hashing, and an incoming HTTP request does not contain a domain name, the NetScaler appliance defaults to the round robin method for that request.

The hash-value calculation uses the name length or hash length value, whichever is smaller. By default, the NetScaler appliance calculates the hash value from the first 80 bytes of the domain name. To specify a different number of bytes in the domain name when calculating the hash value, you can set the hashLength parameter (Hash Length in the configuration utility) to a value of from 1 to 4096 (bytes).

To configure the domain hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Destination IP Hash Method

A load balancing virtual server configured to use the destination IP hash method uses the hashed value of the destination IP address to select a server. You can mask the destination IP address to specify which part of it to use in the hash value

© 1999-2017 Citrix Systems, Inc. All rights reserved.

calculation, so that requests that are from different networks but destined for the same subnet are all directed to the same server. This method supports IPv4 and IPv6-based destination servers.

This load balancing method is appropriate for use with the cache redirection feature.

To configure the destination IP hash method for an IPv4 destination server, you set the netMask parameter. To configure this method for an IPv6 destination server, you use the v6NetMaskLen parameter. In the configuration utility, text boxes for setting these parameters appear when you select the Destination IP Hash method.

To configure the destination IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Hash Method

A load balancing virtual server configured to use the source IP hash method uses the hashed value of the client IPv4 or IPv6 address to select a service. To direct all requests from source IP addresses that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the netMask parameter. For IPv6 addresses, use the v6NetMaskLength parameter.

To configure the source IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Destination IP Hash Method

A load balancing virtual server configured to use the source IP destination IP hash method uses the hashed value of the source and destination IP addresses (either IPv4 or IPv6) to select a service. Hashing is symmetric; the hash-value is the same regardless of the order of the source and destination IPs. This ensures that all packets flowing from a particular client to the same destination are directed to the same server.

To direct all requests that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the netMask parameter. For IPv6 addresses, use the v6NetMaskLength parameter.

To configure the source IP destination IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Source Port Hash Method

A load balancing virtual server configured to use the source IP source port hash method uses the hash value of the source IP (either IPv4 or IPv6) and source port to select a service. This ensures that all packets on a particular connection are directed to the same service.

This method is used in connection mirroring and firewall load balancing. For more information about connection mirroring, see Connection Failover.

To direct all requests that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the netMask parameter. For IPv6 addresses, use the v6NetMaskLength parameter.

To configure the source IP source port hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Call ID Hash Method

A load balancing virtual server configured to use the call ID hash method uses the hash value of the call ID in the SIP header to select a service. Packets for a particular SIP session are therefore always directed to the same proxy server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

This method is applicable to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

To configure the call ID hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The Least Bandwidth Method

Nov 07, 2017

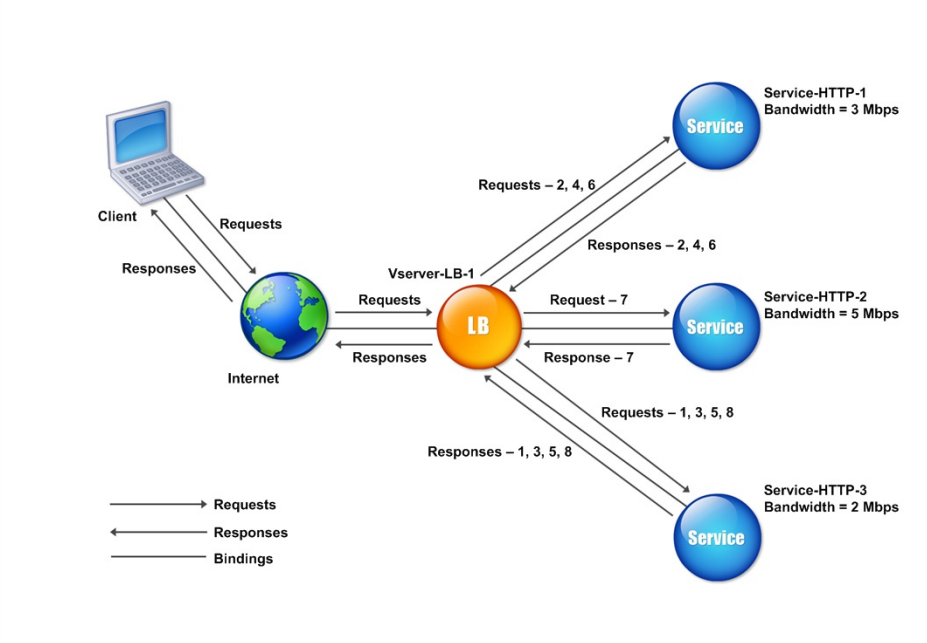
A load balancing virtual server configured to use the least bandwidth method selects the service that is currently serving the least amount of traffic, measured in megabits per second (Mbps). The following example shows how the virtual server selects a service for load balancing by using the least bandwidth method.

Consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 has 3 Mbps bandwidth.
- Service-HTTP-2 has 5 Mbps bandwidth.
- Service-HTTP-3 has 2 Mbps bandwidth.

The following diagram illustrates how the virtual server uses the least bandwidth method to forward requests to the three services.

Figure 1. How the Least Bandwidth Load Balancing Method Works



The virtual server selects the service by using the bandwidth value (N), which is the sum of the number of bytes transmitted and received over the previous 14 seconds. If each request requires 1 Mbps bandwidth, the NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service has the lowest N value.
- Since Service-HTTP-1 and Service-HTTP-3 now have same N value, the virtual server switches to the round robin method for these servers, alternating between them. Service-HTTP-1 receives the second request, Service-HTTP-3 receives the third request, Service-HTTP-1 receives the fourth request, Service-HTTP-3 receives the fifth request, and Service-HTTP-1 receives the sixth request.
- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 now all have same N value, the virtual server includes Service-HTTP-2 in the round robin list. Therefore, Service-HTTP-2 receives the seventh request, Service-HTTP-3 receives the eighth request, and so on.

The following table summarizes how N is calculated.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Received	Service Selected	Current N Value	Remarks
Request-1	Service-HTTP-3 (N = 2)	N = 3	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-1 (N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-3	Service-HTTP-3 (N = 3)	N = 4	
Request-4	Service-HTTP-1 (N = 4)	N = 5	
Request-5	Service-HTTP-3 (N = 4)	N = 5	
Request-6	Service-HTTP-1 (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2 (N = 5)	N = 6	
Request-8	Service-HTTP-3 (N = 5)	N = 6	

Note: If you enable the RTSP NAT option on the virtual server, the NetScaler appliance uses the number of data and control bytes exchanged to determine the bandwidth usage for RTSP services. For more information about RTSP NAT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

option, see Managing RTSP Connections.
The NetScaler appliance also performs load balancing by using the bandwidth and weights if different weights are assigned to the services. It selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. The NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-1 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

The following table summarizes how Nw is calculated.

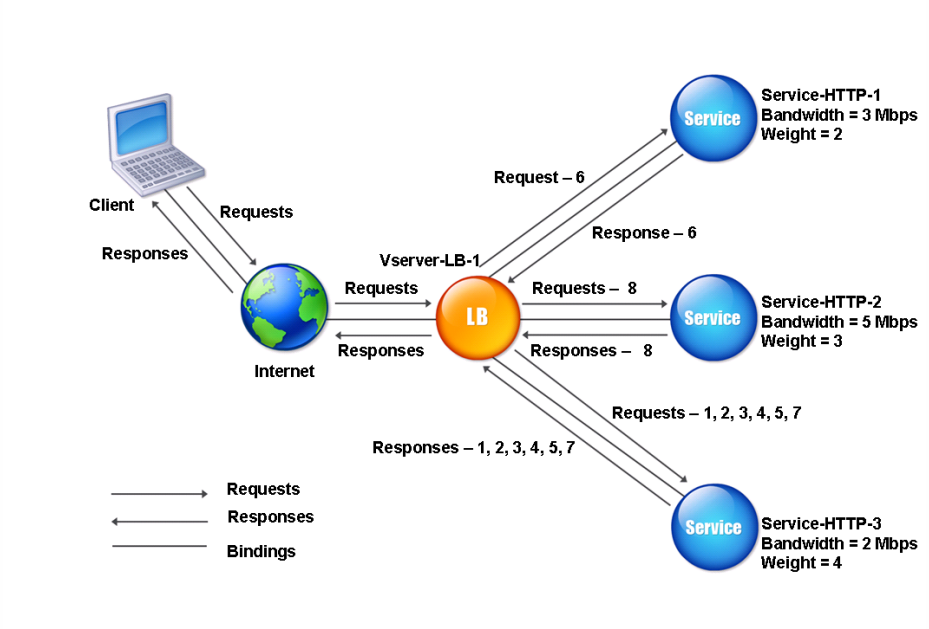
Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3 (Nw = 5000)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3 (Nw = 5000)	Nw = 7500	
Request-3	Service-HTTP-3 (Nw = 7500)	Nw = 10000	
Request-4	Service-HTTP-3 (Nw = 10000)	Nw = 12500	
Request-5	Service-HTTP-3 (Nw =	Nw = 15000	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-6	Service-HTTP-1 (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-7	Service-HTTP-3 (Nw = 15000)	Nw = 17500	
Request-8	Service-HTTP-2 (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 has the lowest Nw value.

The following diagram illustrates how the virtual server uses the least bandwidth method when weights are assigned to the services.

Figure 2. How the Least Bandwidth Load Balancing Method Works When Weights Are Assigned



To configure the least bandwidth method, see Configuring a Load Balancing Method that Does Not Include a Policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The Least Packets Method

Nov 07, 2017

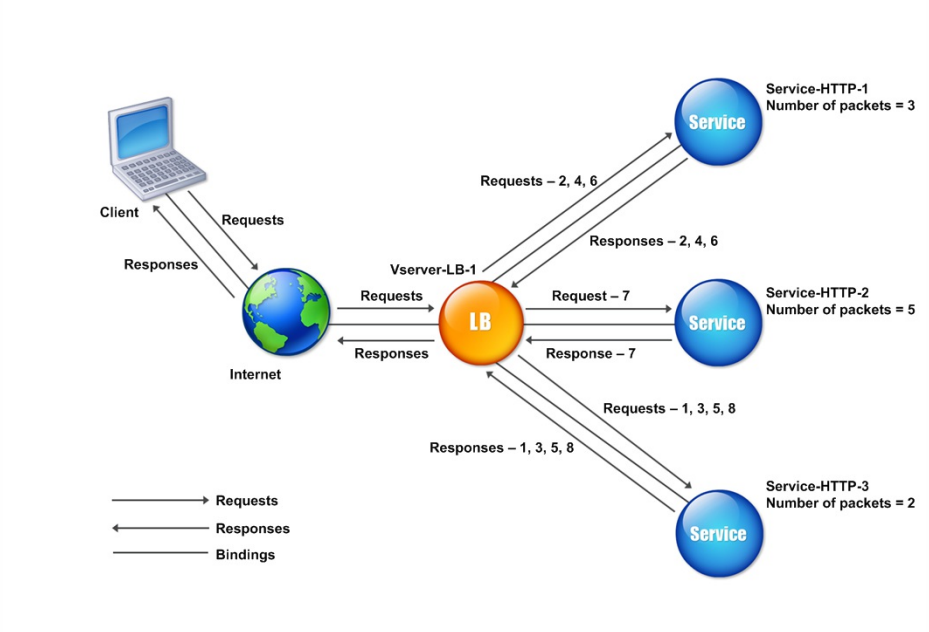
A load balancing virtual server configured to use the least packets method selects the service that has received the fewest packets in the last 14 seconds.

For example, consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 has handled three packets in last 14 seconds.
- Service-HTTP-2 has handled five packets in last 14 seconds.
- Service-HTTP-3 has handled two packets in last 14 seconds.

The following diagram illustrates how the NetScaler appliance uses the least packets method to choose a service for each request that it receives.

Figure 1. How the Least Packets Load Balancing Method Works



The NetScaler appliance selects a service by using the number of packets (N) transmitted and received by each service in the last 14 seconds. Using this method, it delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service has the lowest N value.
- Since Service-HTTP-1 and Service-HTTP-3 now have the same N value, the virtual server switches to the round robin method. Service-HTTP-1 therefore receives the second request, Service-HTTP-3 receives the third request, Service-HTTP-1 receives the fourth request, Service-HTTP-3 receives the fifth request, and Service-HTTP-1 receives the sixth request.
- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all now have same N value, the virtual server switches to the round robin method for Service-HTTP-2 as well, including it in the round robin list. Therefore, Service-HTTP-2 receives the seventh request, Service-HTTP-3 receives the eighth request, and so on.

The following table summarizes how N is calculated.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Received	Service Selected	Current N Value	Remarks
Request-1	Service-HTTP-3 (N = 2)	N = 3	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-1 (N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-3	Service-HTTP-3 (N = 3)	N = 4	
Request-4	Service-HTTP-1 (N = 4)	N = 5	
Request-5	Service-HTTP-3 (N = 4)	N = 5	
Request-6	Service-HTTP-1 (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2 (N = 5)	N = 6	
Request-8	Service-HTTP-3 (N = 5)	N = 6	

Note: If you enable the RTSP NAT option on the virtual server, the NetScaler uses the number of data and control packets to calculate the number of packets for RTSP services. For more information about RTSP NAT option, see Managing RTSP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Connections.

The NetScaler appliance also performs load balancing by using the number of packets and weights when a different weight is assigned to each service. It selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. The NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-1 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

The following table summarizes how Nw is calculated.

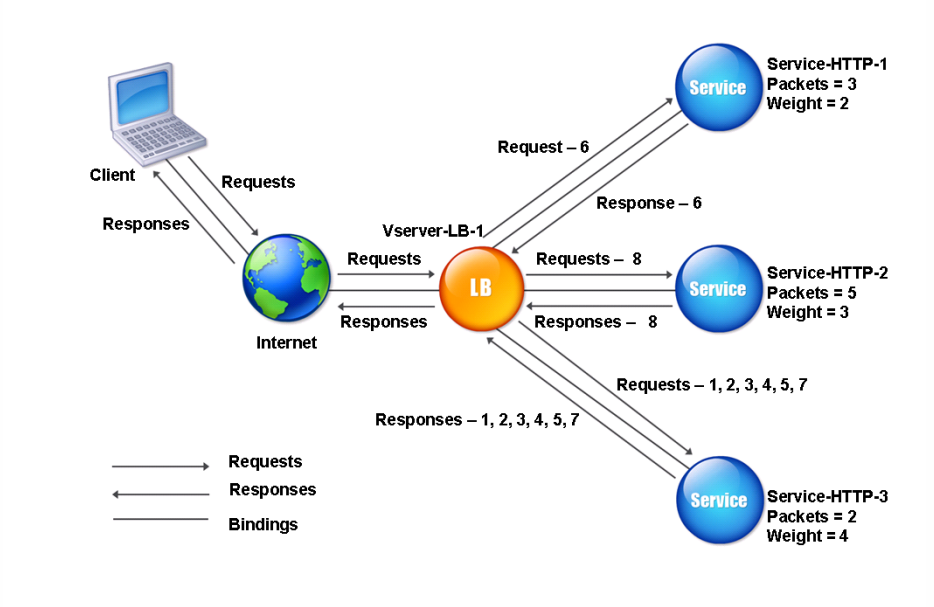
Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / weight)	Remarks
Request-1	Service-HTTP-3 (Nw = 5000)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3 (Nw = 5000)	Nw = 7500	
Request-3	Service-HTTP-3 (Nw = 7500)	Nw = 10000	
Request-4	Service-HTTP-3 (Nw = 10000)	Nw = 12500	
Request-5	Service-HTTP-3 (Nw =	Nw = 15000	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / weight)	Remarks
Request-6	Service-HTTP-1 (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-7	Service-HTTP-3 (Nw = 15000)	Nw = 17500	
Request-8	Service-HTTP-2 (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 has the lowest Nw value.

The following diagram illustrates how the virtual server uses the least packets method when weights are assigned.

Figure 2. How the Least Packets Method Works When Weights Are Assigned



To configure the least packets method, see Configuring a Load Balancing Method that Does Not Include a Policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

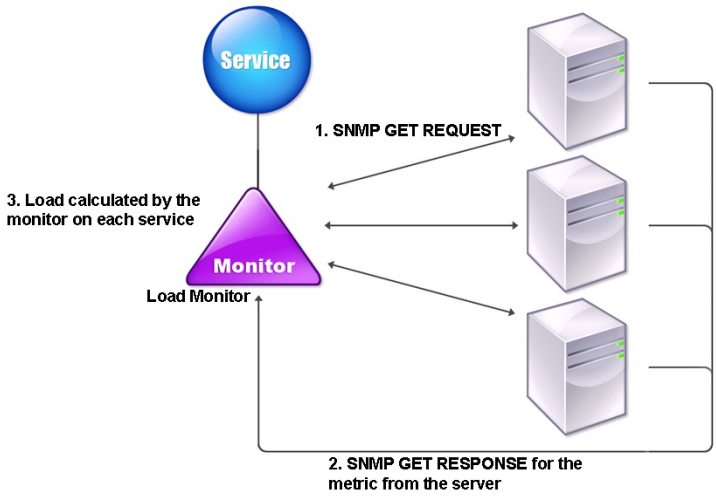
The Custom Load Method

Jul 03, 2017

Custom load balancing is performed on server parameters such as CPU usage, memory, and response time. When using the custom load method, the NetScaler appliance usually selects a service that is not handling any active transactions. If all of the services in the load balancing setup are handling active transactions, the appliance selects the service with the smallest load. A special type of monitor, known as a load monitor, calculates the load on each service in the network. The load monitors do not mark the state of a service, but they do take services out of the load balancing decision when those services are not UP.

For more information about load monitors, see Understanding Load Monitors. The following diagram illustrates how a load monitor operates.

Figure 1. How Load Monitors Operate



The load monitor uses Simple Network Management Protocol (SNMP) probes to calculate load on each service by sending an SNMP GET request to the service. This request contains one or more object IDs (OIDs). The service responds with an SNMP GET response, with metrics corresponding to the SNMP OIDs. The load monitor uses the response metrics, described below, to calculate the load on the service.

The load monitor calculates the load on a service by using the following parameters:

- Metrics values retrieved through SNMP probes that exist as tables in the NetScaler.
- Threshold value set for each metric.
- Weight assigned to each metric.

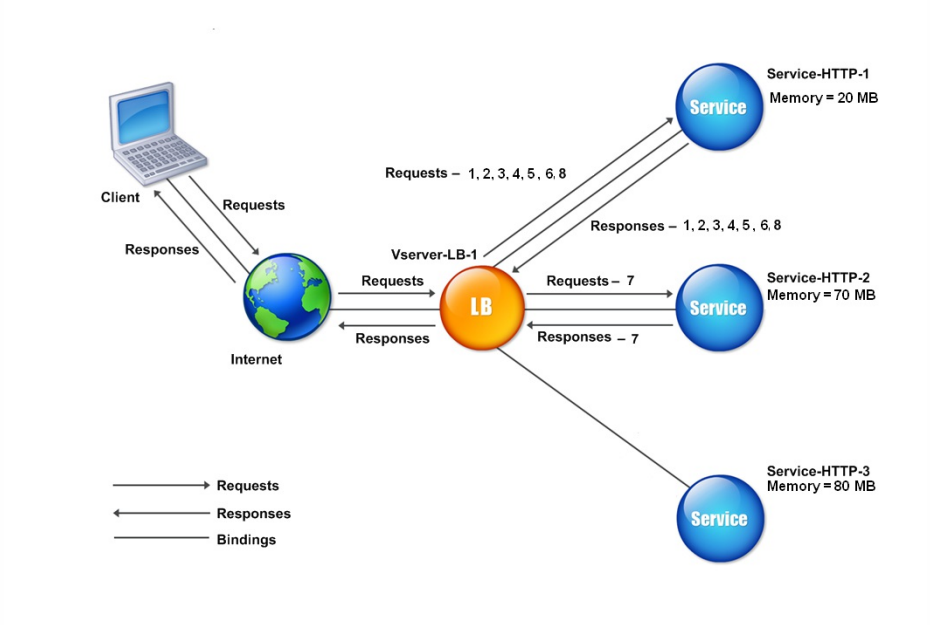
For example, consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 is using 20 megabytes (MB) of memory.
- Service-HTTP-2 is using 70 MB of memory.
- Service-HTTP-3 is using 80 MB of memory.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The load balanced servers can export metrics such as CPU and memory usage to the services, which can in turn provide them to the load monitor. The load monitor sends an SNMP GET request containing the OIDs 1.3.6.1.4.1.5951.4.1.1.41.1.5, 1.3.6.1.4.1.5951.4.1.1.41.1.4, and 1.3.6.1.4.1.5951.4.1.1.41.1.3 to the services. SNMP OIDs of type STRING are not supported, because you cannot calculate the load by using a STRING OID. Loads can be calculated by using other data types, such as INT and gauge32. The three services respond to the request. The NetScaler appliance compares the exported metrics, and then selects Service-HTTP-1 because it has more available memory. The following diagram illustrates this process.

Figure 2. How the Custom Load Method Works



If each request uses 10 MB memory, the NetScaler appliance delivers requests as follows:

- Service-HTTP-1 receives the first, second, third, fourth, and fifth requests, because this service has the lowest N value.
- Service-HTTP-1 and Service-HTTP-2 now have the same load, so the virtual server reverts to the round robin method for these servers. Therefore, Service-HTTP-2 receives the sixth request, and Service-HTTP-1 receives the seventh request.
- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all now have same load, the virtual server reverts to the round robin method for Service-HTTP-3 as well. Therefore, Service-HTTP-3 receives the eighth request.

The following table summarizes how N is calculated.

Request received	Service selected	Current N Value (Number of Active Transactions)	Remarks
Request-1	Service-HTTP-1 (N = 20)	N = 30	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-1 (N = 30)	N = 40	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request received	Service selected	Current N Value (Number of Active Transactions)	Remarks
Request-3	Service-HTTP-1 (N = 40)	N = 50	
Request-4	Service-HTTP-1 (N = 50)	N = 60	
Request-5	Service-HTTP-1 (N = 60)	N = 70	
Request-6	Service-HTTP-1 (N = 70)	N = 80	Service-HTTP-2 and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2 (N = 70)	N = 80	
Request-8	Service-HTTP-1 (N = 80)	N = 90	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.

If different weights are assigned to the services, the custom load algorithm considers both the load on each service and the weight assigned to each service. It selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 4, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 2. If each request uses 10 MB memory, the NetScaler appliance delivers requests as follows:

- Service-HTTP-1 receives the first, second, third, fourth, fifth, sixth, seventh, and eighth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the ninth request, because this service has the lowest Nw value.

Service-HTTP-3 has the highest Nw value, and is therefore not considered for load balancing.

The following table summarizes how Nw is calculated.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

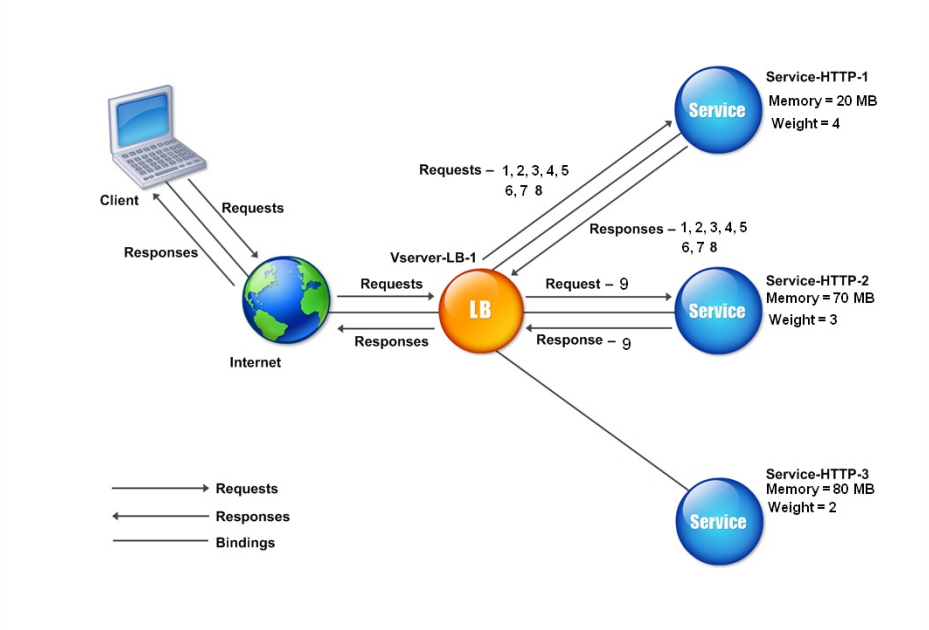
Request received	Service selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-1 (Nw = 50000)	Nw = 75000	Service-HTTP-1 has the lowest Nw value.
Request-2	Service-HTTP-1 (Nw = 5000)	Nw = 100000	
Request-3	Service-HTTP-1 (Nw = 15000)	Nw = 125000	
Request-4	Service-HTTP-1 (Nw = 20000)	Nw = 150000	
Request-5	Service-HTTP-1 (Nw = 23333.34))	Nw = 175000	
Request-6	Service-HTTP-1 (Nw = 25000)	Nw = 200000	
Request-7	Service-HTTP-1 (Nw = 23333.34)	Nw = 225000	
Request-8	Service-	Nw = 250000	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request received	Service selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
	HTTP-1 (Nw = 25000)		
Request-9	Service-HTTP-2 (Nw = 233333.34)	Nw = 266666.67	Service-HTTP-2 has the lowest Nw value.
Service-HTTP-1 is selected for load balancing when it completes its active transactions or when the Nw value of other services (Service-HTTP-2 and Service-HTTP-3) is equal to 400,000.			

The following diagram illustrates how the NetScaler appliance uses the custom load method when weights are assigned.

Figure 3. How the Custom Load Method Works When Weights Are Assigned



To configure the custom load method, see Configuring a Load Balancing Method that Does Not Include a Policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The Static Proximity Method

Jul 07, 2017

When a virtual server is configured to use the static proximity method, it selects the service that best matches the proximity criteria.

For the static proximity method to work, you must either configure the NetScaler appliance to use an existing static proximity database populated through a location file or add custom entries to the static proximity database. After adding custom entries, you can set their location qualifiers. After configuring the database, you are ready to specify static proximity as the load balancing method.

For more details, see the following topics.

- Adding a Location File to Create a Static Proximity Database
- Adding Custom Entries to a Static Proximity Database
- Setting the Location Qualifiers
- Specifying the static proxmity method

Specifying the Static Proximity Method

When you have configured the static proximity database, you are ready to specify static proximity as the GLSB method.

To specify static proximity by using the command line interface

At the command prompt, type the following commands to configure static proximity and verify the configuration:

- set lb vserver <name> -lbMethod STATICPROXIMITY
- show lb vserver <name>

Example

```
set lb vserver Vserver-LB-1 -lbMethod STATICPROXIMITY

show lb vserver
```

To specify static proximity by using the configuration utility

1. Navigate to **Traffic Managemen**t > **Load Balancing** > **Virtual Servers.**
2. Select the virtual server, click **Edit** and expand the **Method** section.
3. In the **Load Balancing Method** list, select **STATICPROXIMITY**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Token Method

Nov 07, 2017

A load balancing virtual server configured to use the token method bases its selection of a service on the value of a data segment extracted from the client request. The data segment is called the token. You configure the location and size of the token. For subsequent requests with the same token, the virtual server chooses the same service that handled the initial request.

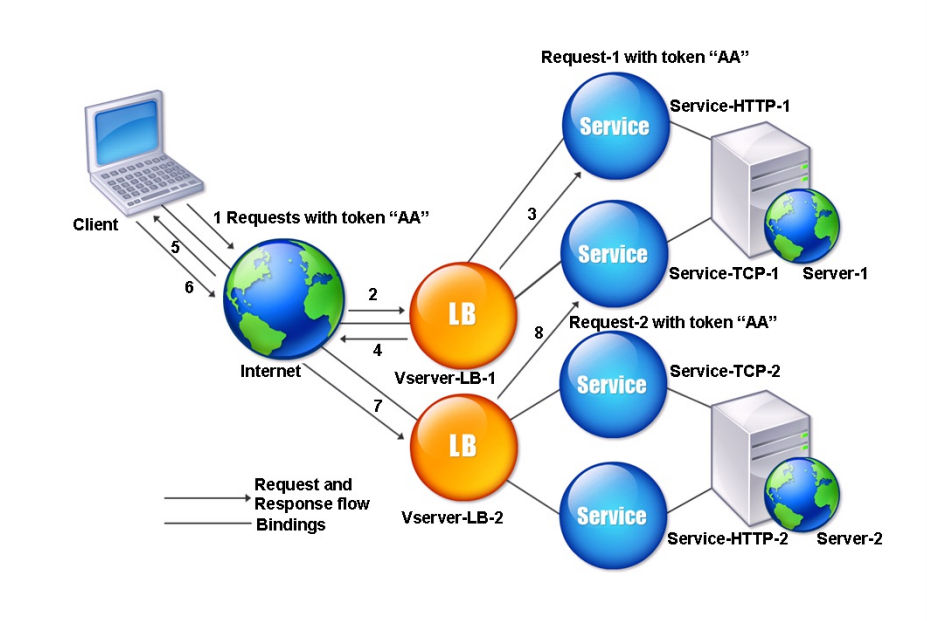
This method is content aware; it operates differently for TCP, HTTP, and HTTPS connections. For HTTP or HTTPS services, the token is found in the HTTP headers, the URL, or the BODY. To locate the token, you specify or create a classic or advanced expression. For more information on classic or advanced expressions, see Policies and Expressions.

For HTTP services, the virtual server searches for the configured token in the first 24 kilobytes (KB) of the TCP payload. For non-HTTP (TCP, SSL, and SSL_TCP) services, the virtual server searches for the configured token in the first 16 packets if the total size of the 16 packets is less than 24 KB. But if the total size of the 16 packets is greater than 24 KB, the NetScaler searches for the token in the first 24 KB of payload. You can use this load balancing method across virtual servers of different types to make sure that requests presenting the same token are directed to appropriate services, regardless of the protocol used.

For example, consider a load balancing setup consisting of servers that contain Web content. You want to configure the NetScaler appliance to search for a specific string (the token) inside the URL query portion of the request. Server-1 has two services, Service-HTTP-1 and Service-TCP-1, and Server-2 has two services, Service-HTTP-2 and Service-TCP-2. The TCP services are bound to Vserver-LB-2, and the HTTP services are bound to Vserver-LB-1.

If Vserver-LB-1 receives a request with the token AA, it selects the service Service-HTTP-1 (bound to server-1) to process the request. If Vserver-LB-2 receives a different request with the same token (AA), it directs this request to the service Service-TCP-1. The following diagram illustrates this process.

Figure 1. How the Token Method Works



To configure the Token load balancing method by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands to configure the token load balancing method and verify the configuration:

- set lb vserver <name> -lbMethod TOKEN -rule <rule> -datalength <length> -dataoffset <offset>
- show lb vserver <name>

Example

```
set lb vserver LB-VServer-1 -lbMethod TOKEN -rule 'AA' -datalength 2 -dataoffset 25

show lb vserver LB-VServer-1
```

To configure the token load balancing method by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, click Method
3. In the Load Balancing Method list, select Token, and specify an expression.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Load Balancing Method That Does Not Include a Policy

Nov 07, 2017

After you select a load balancing algorithm for your load balancing setup, you must configure the NetScaler appliance to use that algorithm. You can configure it by using the NetScaler command line or by using the configuration utility.

Note:
The token method is policy based and requires more configuration than is described here. To configure the token method, see Configuring the Token Method.

For some hash-based methods, you can mask an IP address to direct requests belonging to the same subnet to the same server. For more information, see the sections The Destination IP Hash Method, The Source IP Hash Method, The Source IP Destination IP Hash Method, and The Source IP Source Port Hash Method in About Hashing Methods.

To set the load balancing method by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -lbMethod <method>
```

Example

```
set lb vserver Vserver-LB-1 -lbMethod LeastConnection
```

To set the load balancing method by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, click Method, and in the Load Balancing Method list, select a method.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence and Persistent Connections

Jan 01, 2018

Unless you configure persistence, a load balancing stateless protocol, such as HTTP, disrupts the maintenance of state information about client connections. Different transmissions from the same client might be directed to different servers even though all of the transmissions are part of the same session. You must configure persistence on a load balancing virtual server that handles certain types of Web applications, such as shopping cart applications.

Before you can configure persistence, you need to understand the different types of persistence, how they are used, and what the implications of each type is. You then need to configure the NetScaler appliance to provide persistent connections for those Web sites and Web applications that require them.

You can also configure backup persistence, which takes effect in the event that the primary type of persistence configured for a load balancing virtual server fails. You can configure persistence groups, so that a client transmission to any virtual server in a group can be directed to a server that has received previous transmissions from the same client.

For information about persistence with RADIUS load balancing, see Configuring RADIUS Load Balancing with Persistence.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

About Persistence

Mar 16, 2012

You can choose from among any of several types of persistence for a given load balancing virtual server, which then routes to the same service all connections from the same user to your shopping cart application, Web-based email, or other network application. The persistence session remains in effect for a period of time, which you specify.

If a server participating in a persistence session goes DOWN, the load balancing virtual server uses the configured load balancing method to select a new service, and establishes a new persistence session with the server represented by that service. If the server goes OUT OF SERVICE, it continues to process existing persistence sessions, but the virtual server does not direct any new traffic to it. After the shutdown period elapses, the virtual server ceases to direct connections from existing clients to the service, closes existing connections, and redirects those clients to new services if necessary.

Depending on the persistence type you configure, the NetScaler appliance might examine the source IPs, destination IPs, SSL session IDs, Host or URL headers, or some combination of these things to place each connection in the proper persistence session. It might also base persistence on a cookie issued by the Web server, on an arbitrarily assigned token, or on a logical rule. Almost anything that allows the appliance to match connections with the proper persistence session and be used as the basis for persistence.

The following table summarizes the persistence types available on the NetScaler appliance.

Table 1. Types of Persistence

Persistence Type	Description
Source IP	SOURCEIP. Connections from the same client IP address are parts of the same persistence session.
HTTP Cookie	COOKIEINSERT. Connections that have the same HTTP Cookie header are parts of the same persistence session.
SSL Session ID	SSLSESSION. Connections that have the same SSL Session ID are parts of the same persistence session.
URL Passive	URLPASSIVE. Connections to the same URL are treated as parts of the same persistence session.
Custom Server ID	CUSTOMSERVERID. Connections with the same HTTP HOST header are treated as parts of the same persistence session.
Destination IP	DESTIP. Connections to the same destination IP are treated as parts of the same persistence session.
Source and	SRCIPDESTIP. Connections that are both from the same source IP and to the same destination IP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Type	Description
Destination IPs	are treated as parts of the same persistence session.
SIP Call ID	CALLID. Connections that have the same call ID in the SIP header are treated as parts of the same persistence session.
RTSP Session ID	RTSPSID. Connections that have the same RTSP Session ID are treated as parts of the same persistence session.
User-Defined Rule	RULE. Connections that match a user-defined rule are treated as parts of the same persistence session.

Depending on the type of persistence that you have configured, the virtual server can support either 250,000 simultaneous persistent connections or any number of persistent connections up to the limits imposed by the amount of RAM on your NetScaler appliance. The following table shows which types of persistence fall into each category.

Table 2. Persistence Types and Numbers of Simultaneous Connections Supported

Persistence Type	Number of Simultaneous Persistent Connections Supported
Source IP, SSL Session ID, Rule, destination IP, source IP/destination IP, SIP Call ID, RTSP Session ID	250 K
Cookie, URL Server ID, Custom Server ID	Memory limit. In case of CookieInsert, if timeout is not 0, the number of connections is limited by memory.

Some types of persistence are specific to particular types of virtual server. The following table lists each type of persistence and indicates which types of persistence are supported on which types of virtual server.

Table 3. Relationship of Persistence Type to Virtual Server Type

Persistence Type	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
SOURCEIP	YES	YES	YES	YES	YES	YES	NO	NO
COOKIEINSERT	YES	YES	NO	NO	NO	NO	NO	NO
SSLSESSION	NO	YES	NO	NO	YES	YES	NO	NO
URLPASSIVE	YES	YES	NO	NO	NO	NO	NO	NO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Type	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
CUSTOMSERVERID	YES	YES	NO	NO	NO	NO	NO	NO
RULE	YES	YES	YES	NO	NO		NO	NO
SRCIPDESTIP	YES	YES	YES	YES	YES	YES	NO	NO
DESTIP	YES	YES	YES	YES	YES	YES	NO	NO
CALLID	NO	NO	NO	NO	NO	NO	NO	YES
RTSPID	NO	NO	NO	NO	NO	NO	YES	NO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Based on Source IP Address

Oct 21, 2015

When source IP persistence is configured, the load balancing virtual server uses the configured load balancing method to select a service for the initial request, and then uses the source IP address (client IP address) to identify subsequent requests from that client and send them to the same service. You can set a time-out value, which specifies the maximum inactivity period for the session. When the time-out value expires, the session is discarded, and the configured load balancing algorithm is used to select a new server.

Caution: In some circumstances, using persistence based on source IP address can overload your servers. All requests to a single Web site or application are routed through the single gateway to the NetScaler appliance, even though they are then redirected to multiple locations. In multiple proxy environments, client requests frequently have different source IP addresses even when they are sent from the same client, resulting in rapid multiplication of persistence sessions where a single session should be created. This issue is called the "Mega Proxy problem." You can use HTTP cookie-based persistence instead of Source IP-based persistence to prevent this from happening.
To configure persistence based on Source IP Address, see Configuring Persistence Types That Do Not Require a Rule.

Note: If all incoming traffic comes from behind a Network Address Translation (NAT) device or proxy, the traffic appears to the NetScaler appliance to come from a single source IP address. This prevents Source IP persistence from functioning properly. Where this is the case, you must select a different persistence type.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Based on HTTP Cookies

Oct 21, 2015

When HTTP cookie persistence is configured, the NetScaler appliance sets a cookie in the HTTP headers of the initial client request. The cookie contains the IP address and port of the service selected by the load balancing algorithm. As with any HTTP connection, the client then includes that cookie with any subsequent requests.

When the NetScaler appliance detects the cookie, it forwards the request to the service IP and port in the cookie, maintaining persistence for the connection. You can use this type of persistence with virtual servers of type HTTP or HTTPS. This persistence type does not consume any NetScaler resources and therefore can accommodate an unlimited number of persistent clients.

Note: If the client's Web browser is configured to refuse cookies, HTTP cookie-based persistence will not work. It might be advisable to configure a cookie check on the Web site, and warn clients that do not appear to be storing cookies properly that they will need to enable cookies for the Web site if they want to use it.
The format of the cookie that the NetScaler appliance inserts is:

NSC_XXXX=<ServiceIP ><ServicePort>

where:

- NSC_XXXX is the virtual server ID that is derived from the virtual server name.
- ServiceIP and ServicePort are encoded representations of the service IP address and service port, respectively. The IP address and port are encoded separately.

You can set a time-out value for this type of persistence to specify an inactivity period for the session. When the connection has been inactive for the specified period of time, the NetScaler appliance discards the persistence session. Any subsequent connection from the same client results in a new server being selected based on the configured load balancing method, and a new persistence session being established.

Note: If you set the time-out value to 0, the NetScaler appliance does not specify an expiration time, but sets a session cookie that is not saved when the client's browser is shut down.
By default, the NetScaler appliance sets HTTP version 0 cookies for maximum compatibility with client browsers. (Only certain HTTP proxies understand version 1 cookies; most commonly used browsers do not.) You can configure the appliance to set HTTP version 1 cookies, for compliance with RFC2109. For HTTP version 0 cookies, the appliance inserts the cookie expiration date and time as an absolute Coordinated Universal Time (GMT). It calculates this value as the sum of the current GMT time on the appliance and the time-out value. For HTTP version 1 cookies, the appliance inserts a relative expiration time by setting the "Max-Age" attribute of the HTTP cookie. In this case, the client's browser calculates the actual expiration time.

To configure persistence based on a cookie inserted by the appliance, see Configuring Persistence Types That Do Not Require a Rule.

In the HTTP cookie, the appliance by default sets the httponly flag to indicate that the cookie is nonscriptable and should not be revealed to the client application. Therefore, a client-side script cannot access the cookie, and the client is not susceptible to cross-site scripting.

Certain browsers, however, do not support the httponly flag and, therefore, might not return the cookie. As a result, persistence is broken. For browsers that do not support the flag, you can omit the httponly flag in the persistence cookie.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To change the httponly flag setting by using the command line interface

At the command prompt, type:

set lb parameter -httpOnlyCookieFlag (ENABLED | DISABLED)

Example

```
> set lb parameter -httpOnlyCookieFlag disabled
  Done
> show lb parameter
 Global LB parameters:
     Persistence Cookie HttpOnly Flag: DISABLED
     Use port for hash LB: YES
 Done
```

To change the httponly flag setting by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing Parameters, and select or clear the Persistence Cookie HTTPOnly Flag.

Encrypting the Cookie

From release 10.5 build 55.8, you can encrypt the cookie in addition to any SSL encryption.

To encrypt the cookie by using the command line interface, at the command prompt, type:

set lb parameter -useSecuredPersistenceCookie ENABLED -cookiePassphrase test

To encrypt the cookie by using the configuration utility, navigate to Traffic Management > Change Load Balancing Parameters, and select Use Secured Persistence Cookie and Cookie Passphrase and enter a passphrase.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Based on SSL Session IDs

Aug 22, 2016

When SSL Session ID persistence is configured, the NetScaler appliance uses the SSL Session ID, which is part of the SSL handshake process, to create a persistence session before the initial request is directed to a service. The load balancing virtual server directs subsequent requests that have the same SSL session ID to the same service. This type of persistence is used for SSL bridge services.

Note:

There are two issues that users should consider before choosing this type of persistence. First, this type of persistence consumes resources on the NetScaler appliance, which limits the number of concurrent persistence sessions that it can support. If you expect to support a very large number of concurrent persistence sessions, you might want to choose another type of persistence.

Second, if the client and the load-balanced server should renegotiate the session ID during their transactions, persistence is not maintained, and a new persistence session is created when the client's next request is received. This may result in the client's activity on the Web site being interrupted and the client being required to reauthenticate or restart the session. It may also result in large numbers abandoned sessions if the timeout is set to too large a value.

To configure persistence based on SSL session ID, see Configuring Persistence Types That Do Not Require a Rule.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Based on Diameter AVP Number

Aug 25, 2016

You can use persistence based on the Attribute-Value Pair (AVP) number of a Diameter message to create persistent Diameter sessions. When the NetScaler appliance finds the AVP in the Diameter message, it creates a persistence session based on the value of the AVP. All subsequent messages that match the value of the AVP are directed to the previously selected server. If the value of the AVP does not match the persistence session, a new session is created for the new value.

Note: If the AVP number is not defined in Diameter base-protocol RFC 6733, and if the number is nested inside a grouped AVP, you must define a sequence of AVP numbers (maximum of 3) in parent-to-child order. For example, if persist AVP number X is nested inside AVP Y, which is nested in Z, define the list as Z Y X.

To configure Diameter-based persistence on a virtual server by using the command line interface

At the command prompt, type the following command:

set lb vserver <name> -PersistenceType <type-> persistAVPno <positive_integer>

Example

set lb vserver diameter_vs -persistenceType DIAMETER -persistAVPno 263

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Custom Server ID Persistence

Jan 01, 2018

In the Custom Server ID persistence method, the Server ID specified in the client request is used to maintain persistence. For this type of persistence to work, you must first set a server ID on the services. The NetScaler appliance checks the URL of the client request and connects to the server associated with the specified server ID. The service provider should make sure that the users are aware of the server IDs to be provided in their requests for specific services.

For example, if your site provides different types of data, such as images, text, and multimedia, from different servers, you can assign each server a server ID. On the NetScaler appliance, you specify those server IDs for the corresponding services, and you configure custom server ID persistence on the corresponding load balancing virtual server. When sending a request, the client inserts the server ID into the URL indicating the required type of data.

To configure custom server ID persistence:

- In your load balancing setup, assign a server ID to each service for which you want to use the user-defined server ID to maintain persistence. Alphanumeric server IDs are allowed.
- Specify rules, in the default-syntax expression language, to examine the URL queries for the server ID and forward traffic to the corresponding server.
- Configure custom server ID persistence.

Note: The persistence time-out value does not affect the Custom Server ID persistence type. There is no limit on the maximum number of persistent clients because this persistence type does not store any client information.

Example

In a load balancing setup with two services, assign server ID 2345-photo-56789 to Service-1, and server ID 2345-drawing-abb123 to Service-2. Bind these services to a virtual server named Web11.

set service Service-1 10.102.29.5 -CustomServerID 2345-photo-56789

set service Service-2 10.102.29.6 -CustomServerID 2345-drawing-abb123

On virtual server Web11, enable Custom Server ID persistence.

Create the following expression so that all URL queries containing the string "sid=" are examined.

HTTP.REQ.URL.AFTER_STR("sid=")

Example

set lb vserver Web11 -persistenceType customserverID -rule "HTTP.REQ.URL.AFTER_STR(\"sid=\")"

bind lb vserver Web11 Service-[1-2]

When a client sends a request with the following URL to the IP address of Web11, the NetScaler directs the request to Service-2 and honors persistence.

Example

http://www.example.com/index.asp?&sid=2345-drawing-abb123

For more information about default-syntax policy expressions, see Policies and Expressions.

To configure custom server ID persistence by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service and set a server ID.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
4. In Advanced Settings, select Persistence.
5. Select CUSTOMESERVERID, and specify an expression.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Based on IP Addresses

Nov 23, 2015

You can base persistence on Destination IP addresses, or on both Source IP and Destination IP Addresses.

Persistence Based on Destination IP Addresses

With destination IP address-based persistence, when the NetScaler appliance receives a request from a new client, it creates a persistence session based on the IP address of the service selected by the virtual server (the destination IP address). Subsequently, it directs requests to the same destination IP to the same service. This type of persistence is used with link load balancing. For more information about link load balancing, see Link Load Balancing.

The time-out value for destination IP persistence is the same as that for source IP persistence, described in Persistence Based on Source IP Address.

To configure persistence based on the destination IP address, see Configuring Persistence Types That Do Not Require a Rule.

Persistence Based on Source and Destination IP Addresses

With source and destination IP address-based persistence, when the NetScaler appliance receives a request, it creates a persistence session based on both the IP address of the client (the source IP address) and the IP address of the service selected by the virtual server (the destination IP address). Subsequently, it directs requests from the same source IP and to the same destination IP to the same service.

The time-out value for destination IP persistence is the same as that for source IP persistence, described in Persistence Based on Source IP Address.

To configure persistence based on both source and destination IP addresses, see Configuring Persistence Types That Do Not Require a Rule.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Based on SIP Call ID

Nov 23, 2015

With SIP Call ID persistence, the NetScaler appliance chooses a service based on the call ID in the SIP header. This enables it to direct packets for a particular SIP session to the same service and, therefore, to the same load balanced server. This persistence type is applicable specifically to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

To configure persistence based on SIP Call ID, see Configuring Persistence Types That Do Not Require a Rule.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Persistence Based on RTSP Session IDs

Nov 23, 2015

With RTSP Session ID persistence, when the NetScaler appliance receives a request from a new client, it creates a new persistence session based on the Real-Time Streaming Protocol (RTSP) session ID in the RTSP packet header, and then directs the request to the RTSP service selected by the configured load balancing method. It directs subsequent requests that contain the same session ID to the same service. This persistence type is applicable specifically to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

Note: RTSP Session ID persistence is configured by default on RTSP virtual servers, and you cannot modify that setting. Sometimes different RTSP servers issue the same session IDs. When this happens, unique sessions cannot be created between the client and the RTSP server by using only the RTSP session ID. If you have multiple RTSP servers that may issue the same session IDs, you can configure the appliance to append the server IP address and port to the session ID, creating a unique token that can be used to establish persistence. This is called session ID mapping.

To configure persistence based on RTSP Session IDs, see Configuring Persistence Types That Do Not Require a Rule.

Important: If you need to use session ID mapping, you must set the following parameter when configuring each service within the load balancing setup. Also, make sure that no non-persistent connections are routed through the RTSP virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring URL Passive Persistence

Nov 23, 2015

With URL Passive persistence, when the NetScaler appliance receives a request from a client, it extracts the server IP address-port information (expressed as a single hexadecimal number) from the client request.

URL passive persistence requires configuring an advanced expression that specifies the query element that contains the server IP address-port information. For more information about classic and advanced policy expressions, see Policy Configuration and Reference.

The following expression configures the appliance to examine requests for URL queries that contain the string "urlp=", extract the server IP address-port information, convert it from a hexadecimal string to an IP and port number, and forward the request to the service configured with this IP address and port number.

HTTP.REQ.URL.AFTER_STR("urlp=")

If URL passive persistence is enabled and the above expression is configured, a request with the following URL and server IP address-port string is directed to 10.102.29.10:80.

http://www.example.com/index.asp?&urlp=0A661D0A0050

The persistence time-out value does not affect this persistence type; persistence is maintained as long as the server IP address-port information can be extracted from client requests. This persistence type does not consume any NetScaler resources, so it can accommodate an unlimited number of persistent clients.

To configure URL passive persistence, you first configure persistence as described in Configuring Persistence Types That Do Not Require a Rule. You set the persistence type to URLPASSIVE. You then perform the procedures provided below.

To configure URL passive persistence by using the command line interface

At the command prompt, type:

```
set lb vserver <vserverName>  [-persistenceType <persistenceType>]  [-rule <expression>]
```

Example

```
set lb vserver LB-VServer-1 -persistenceType URLPASSIVE –rule HTTP.REQ.URL.AFTER_STR("urlp=")
```

To configure URL passive persistence by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, select Persistence, and specify URLPASSIVE.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Persistence Based on User-Defined Rules

Nov 23, 2015

When rule based persistence is configured, the NetScaler appliance creates a persistence session based on the contents of the matched rule before directing the request to the service selected by the configured load balancing method. Subsequently, it directs all requests that match the rule to the same service. You can configure rule based persistence for services of type HTTP, SSL, RADIUS, ANY, TCP, and SSL_TCP.

Rule based persistence requires a classic or default syntax expression. You can use a classic expression to evaluate request headers, or you can use a default syntax expression to evaluate request headers, Web form data in a request, response headers, or response bodies. For example, you could use a classic expression to configure persistence based on the contents of the HTTP Host header. You could also use a default syntax expression to configure persistence based on application session information in a response cookie or custom header. For more information on creating and using classic and default syntax expressions, see [Policy Configuration and Reference].

The expressions that you can configure depends on the type of service for which you are configuring rule based persistence. For example, certain RADIUS-specific expressions are not allowed for protocols other than RADIUS, and TCP-option based expressions are not allowed for service types other than the ANY type. For TCP and SSL_TCP service types, you can use expressions that evaluate TCP/IP protocol data, Layer 2 data, TCP options, and TCP payloads.

Note: For a use case that involves configuring rule based persistence on the basis of Financial Information eXchange ("FIX") Protocol data transmitted over TCP, see Configuring Rule Based Persistence Based on a Name-Value Pair in a TCP Byte Stream.

Rule based persistence can be used for maintaining persistence with entities such as Branch Repeater appliances, Branch Repeater plug-ins, cache servers, and application servers.

Note: On an ANY virtual server, you cannot configure rule-based persistence for the responses.

To configure persistence based on a user-defined rule, you first configure persistence as described in Configuring Persistence Types That Do Not Require a Rule, and set the persistence type to RULE. You then perform the procedures provided below.

You can configure rule based persistence by using the configuration utility or the NetScaler command line.

To configure persistence based on user-defined rules by using the command line interface

At the command prompt, type:

```
set lb vserver <vserverName> [-rule <expression>][-resRule <expression>]
```

Example

```
set lb vserver vsvr_name –rule http.req.header("cookie").value(0).typecast_nvlist_t('=',';').value("server")

set lb vserver vsvr_name –resrule http.res.header("set-cookie").value(0).typecast_nvlist_t('=',';').value("server")
```

To configure persistence based on user-defined rules by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, select Persistence, select RULE, and specify an expression.

Example: Classic Expression for a Request Payload

The following classic expression creates a persistence session based on the presence of a User-Agent HTTP header that contains the string, “MyBrowser”, and directs any subsequent client requests that contain this header and string to the same server that was selected for the initial request.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http header User-Agent contains MyBrowser

Example: Default syntax Expression for a Request Header

The following default syntax expression does exactly the same thing as the previous classic expression.

HTTP.REQ.HEADER("User-Agent").CONTAINS ("MyBrowser")

Example: Default syntax Expression for a Response Cookie

The following expression examines responses for “server” cookies, and then directs any requests that contain that cookie to the same server that was selected for the initial request.

HTTP.RES.HEADER("SET-COOKIE").VALUE(0).TYPECAST_NVLIST_T('=',';').VALUE("server")

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Persistence Types That Do Not Require a Rule

Apr 28, 2016

To configure persistence, you must first set up a load balancing virtual server, as described in Setting Up Basic Load Balancing. You then configure persistence on the virtual server.

To configure persistence on a virtual server by using the command line interface

At the command prompt, type the following commands to configure persistence and verify the configuration:

- set lb vserver <name> -PersistenceType <type> [-timeout <integer>]
- show lb vserver

Timeout is the time period for which a persistence session is in effect. Default value: 2. Maximum value: 1440.

Example

```
set lb vserver Vserver-LB-1 -persistenceType SOURCEIP

show lb vserver
```

Note: For IP-based persistence, you can also set the persistMask parameter.

To configure persistence on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, select Persistence, and specify a persistence type other than RULE.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Backup Persistence

Nov 23, 2015

The NetScaler appliance uses backup persistence to choose a new type of persistence when the primary persistence type fails. For example, if the primary persistence type is set to Cookie Insert, and backup persistence is set to Source IP, the NetScaler appliance uses Source IP-based persistence when the cookie is missing from the HTTP header or when the client browser does not support cookies.

You can set a time-out value for backup persistence only when the primary persistence type is HTTP Cookie-based or RTSP session ID-based persistence, and the backup persistence type is Source IP-based.

To set backup persistence for a virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -persistenceType <PersistenceType> -persistenceBackup <BackupPersistenceType>
```

Example

```
set lb vserver Vserver-LB-1 -persistenceType CookieInsert -persistenceBackup SourceIP
```

To set backup persistence for a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, select Persistence, and specify a backup persistence type.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Persistence Groups

Nov 23, 2015

When you have load-balanced servers that handle several different types of connections (such as Web servers that host multimedia), you can configure a virtual server group to handle these connections. To create a virtual server group, you bind different types of virtual servers, one for each type of connection that your load balanced servers accept, into a single group. You then configure a persistence type for the entire group.

You can configure either source IP-based persistence or HTTP cookie-based persistence for persistence groups. After you set persistence for the entire group, you cannot change it for individual virtual servers in the group. If you configure persistence on a group and then add a new virtual server to the group, the persistence of the new virtual server is changed to match the persistence setting of the group.

When persistence is configured on a group of virtual servers, persistence sessions are created for initial requests, and subsequent requests are directed to the same service as initial request, regardless of the virtual server in the group that receives each client request.

If you configure HTTP cookie-based persistence, the domain attribute of the HTTP cookie is set. This setting causes the client software to add the HTTP cookie into client requests if different virtual servers have different public host names. For more information about CookieInsert persistence type, see Persistence Based on HTTP Cookies.

To create a virtual server persistency group by using the command line interface

At the command prompt, type:

bind lb group <vServerGroupName> <vServerName> -persistenceType <PersistenceType>

Example

bind lb group Vserver-Group-1 Vserver-LB-1 -persistenceType CookieInsert

To modify a virtual server group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Persistency Groups, create a persistency group, and specify the virtual servers that must be part of this group.

To modify a virtual server group by using the command line interface

At the command prompt, type:

set lb group <vServerGroupName> -PersistenceBackup <BackupPersistenceType> -persistMask <SubnetMaskAddress>

Example

set lb group vserver-Group-1 -PersistenceBackup SourceIP -persistMask 255.255.255.255

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring RADIUS Load Balancing with Persistence

Nov 23, 2015

Today's complex networking environment often requires coordinating a high-volume, high-capacity load balancing configuration with robust authentication and authorization. Application users may connect to a VPN through mobile access points such as consumer-grade DSL or Cable connections, WiFi, or even dial-up nodes. Those connections usually use dynamic IPs, which can change during the connection.

If you configure RADIUS load balancing on the NetScaler appliance to support persistent client connections to RADIUS authentication servers, the appliance uses the user logon or the specified RADIUS attribute instead of the client IP as the session ID, directing all connections and records associated with that user session to the same RADIUS server. Users are therefore able to log on to your VPN from mobile access locations without experiencing disconnections when the client IP or WiFi access point changes.

To configure RADIUS load balancing with persistence, you must first configure RADIUS authentication for your VPN. For information and instructions, see the Authentication, Authorization, Auditing (AAA) chapter in AAA Application Traffic. You must also choose either the Load Balancing or Content Switching feature as the basis for your configuration, and make sure that the feature you chose is enabled. The configuration process with either feature is almost the same.

Then, you configure either two load balancing, or two content switching, virtual servers, one to handle RADIUS authentication traffic and the other to handle RADIUS accounting traffic. Next, you configure two services, one for each load balancing virtual server, and bind each load balancing virtual server to its service. Finally, you create a load balancing persistency group and set the persistency type to RULE.

To configure RADIUS load balancing with persistence, see the following sections:

- Enabling the Load Balancing or Content Switching Feature
- Configuring Virtual Servers
- Configuring Services
- Binding Virtual Servers to Services
- Configuring a Persistency Group for Radius

Enabling the Load Balancing or Content Switching Feature

To use the Load Balancing or Content Switching feature, you must first ensure that the feature is enabled. If you are configuring a new NetScaler appliance that has not previously been configured, both of these features are already enabled, so you can skip to the next section. If you are configuring a NetScaler appliance with a previous configuration on it, and you are not certain that the feature you will use is enabled, you must do that now.

- For instructions on enabling the load balancing feature, see Enabling Load Balancing.
- For instructions on enabling the content switching feature, see Enabling Content Switching.

Configuring Virtual Servers

After enabling the load balancing or content switching feature, you must next configure two virtual servers to support RADIUS authentication:

- **RADIUS authentication virtual server.** This virtual server and its associated service will handle authentication traffic to your RADIUS server. Authentication traffic consists of connections associated with users logging onto your protected application or virtual private network (VPN).
- **RADIUS accounting virtual server.** This virtual server and its associated service will handle accounting connections to your RADIUS server. Accounting traffic consists of connections that track an authenticated user's activities on your protected application or VPN.

Important: You must create either a pair of load balancing virtual servers or a pair of content switching virtual servers to use in your RADIUS persistence configuration. You cannot mix virtual server types.

To configure a load balancing virtual server by using the command line interface

At the command prompt type the following commands to create a new load balancing virtual server and verify the configuration:

- add lb vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule <rule>
- show lb vserver <name>

To configure an existing load balancing virtual server, replace the above add lb virtual server command with the set lb vserver command, which takes the same arguments.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure a content switching virtual server by using the command line interface

At the command prompt type the following commands to create a new content switching virtual server and verify the configuration:

- add cs vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule <rule>
- show cs vserver <name>

To configure an existing content switching virtual server, replace the above add cs vserver command with the set cs vserver command, which takes the same arguments.

Example

```
add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813 -lbmethod TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
set lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
set lb vserver radius_auth_vs1 RADIUS 192.168.46.34 1813 -lbmethod TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
```

To configure a load balancing or content switching virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers or navigate to Traffic Management > Content Switching > Virtual Servers>>, and configure a virtual server.

Configuring Services

After configuring your virtual servers, you must next configure two services, one for each of the virtual servers that you created. For instructions, see Configuring Services.

Note: Once configured, these services are in the DISABLED state until the NetScaler appliance can connect to your RADIUS server's authentication and accounting IPs and monitor their status.

Binding Virtual Servers to Services

After configuring your services, you must next bind each of the virtual servers that you created to the appropriate service. For instructions, see Binding Services to the Virtual Server.

Configuring a Persistency Group for Radius

After binding your load balancing virtual servers to the corresponding services, you must set up your RADIUS load balancing configuration to support persistence. To do so, you configure a load balancing persistency group that contains your RADIUS load balancing virtual servers and services, and configure that load balancing persistency group to use rule-based persistence. For instructions, see Configuring Persistence Groups.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing Persistence Sessions

Sep 13, 2013

You can view the different persistence sessions that are in effect globally or for a particular virtual server.

Note: A NetScaler nCore appliance uses multiple CPU cores for packet handling. Every session on the appliance is owned by a CPU core. If the appliance receives a request for which a session does not already exist, a session is created, and one of the cores is designated as the owner of that session. Subsequent requests that belong to that session might not always arrive at and be handled by the owner core. In that case, inter-core messaging ensures that the session information on the owner core is always current. However, when a core receives a request that belongs to a persistence session owned by another core, the inter-core messaging does not refresh the timeout value for the persistence session. Consequently, in the output of successively executed show lb persistentSessions commands, which display timeout values from owner cores only, the timeout value for a persistence session might diminish to 0 (zero), even if the persistence session continues to be active.

To view a persistence session by using the command line interface

At the command prompt, to view all persistence sessions type:

show lb persistentSessions [<vServer>]

Example

```
show lb persistentSessions myVserver
```

To view persistence sessions by using the configuration utility

Navigate to Traffic Management > Virtual server persistent sessions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Clearing Persistence Sessions

Nov 23, 2015

You might need to clear persistence sessions from the NetScaler if sessions fail to time out. You can do one of the following:

- Clear all sessions for all virtual servers at once.
- Clear all sessions for a given virtual server at once.
- Clear a particular session that is associated with a given virtual server.

To clear a persistence session by using the command line interface

At the command prompt, type the following commands to clear persistence sessions and verify the configuration:

- clear lb persistentSessions [<vServer> [-persistenceParam <string>]]
- show persistentSessions <vServer>

Examples

Example 1 clears all persistence sessions for load balancing virtual server lbvip1. Example 2 first displays the persistence sessions for load balancing virtual server lbvip1, clears the session with persistence parameter xls, and then displays the persistence sessions to verify that the session was cleared.

Example

```
> clear persistentSessions lbvip1
 Done
> show persistentSessions
 Done
>
```

Example 2

```
> show persistentSessions lbvip1
Type      SRC-IP    ...   PERSISTENCE-PARAMETER
RULE       0.0.0.0  ...   xls
RULE       0.0.0.0  ...   txt
RULE       0.0.0.0  ...   html
 Done
> clear persistentSessions lbvip1 -persistenceParam xls
 Done
> show persistentSessions lbvip1
Type      SRC-IP    ...   PERSISTENCE-PARAMETER
RULE       0.0.0.0  ...   txt
RULE       0.0.0.0  ...   html
 Done
>
```

To clear persistence sessions by using the configuration utility

1. Navigate to Traffic Management > Clear Persistent Sessions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Overriding Persistence Settings for Overloaded Services

Jan 01, 2018

When a service is loaded or is otherwise unavailable, service to clients is degraded. To work around this situation, you might have to configure the NetScaler appliance to temporarily forward to other services the requests that would otherwise be included in the persistence session that is associated with the overloaded service. In other words, you might have to override the persistence setting that is configured for the load balancing virtual server. You can achieve this functionality by setting the skippersistency parameter. With the parameter set, when the virtual server receives new connections for an overloaded service, the virtual server disregards any existing persistence sessions that are associated with that service, until the service returns to a state at which it can accept requests. Persistence sessions associated with other services are not affected. The functionality is available for only virtual servers whose type is ANY or UDP.

In Branch Repeater load balancing configurations, you must also configure a load monitor and bind it to the service. The monitor takes the service out of subsequent load balancing decisions until the load on the service is brought below the configured threshold. For information about configuring a load monitor for your virtual server, see Understanding Load Monitors.

You can configure the virtual server to perform one of the following actions with the requests that would otherwise form a part of the persistence session:

- **Send each request to one of the other services.** The virtual server takes a load balancing decision and sends each request to one of the other services on the basis of the configured load balancing method. If all the services are overloaded, requests are dropped until a service becomes available.
  Both wildcard and IP address–based virtual servers support this option. This action is appropriate for all deployments, including deployments in which the virtual server is load balancing Branch Repeater appliances or firewalls.

- **Bypass the virtual server-service configuration.** The virtual server does not take a load balancing decision. Instead, it simply bridges each request through to a physical server on the basis of the destination IP address in the request.
  Only wildcard virtual servers of type ANY and UDP support the bypass option. Wildcard virtual servers have a *:* IP and port combination. This action is appropriate for deployments in which you are using the virtual server to load balance Branch Repeater appliances or firewalls. In these deployments, the NetScaler appliance first forwards a request to a Branch Repeater appliance or firewall, and then forwards the processed response to a physical server. If you configure the virtual server to bypass the virtual server–service configuration for overloaded services, if a Branch Repeater appliance or firewall gets overloaded, the virtual server bridges requests directly to their destination IP addresses until the Branch Repeater appliance or firewall can accept requests.

To override persistence settings for overloaded services by using the command line interface

At the command prompt, type the following commands to override persistence settings for overloaded services and verify the configuration:

- set lb vserver <name> -skippersistency <skippersistency>
- show lb vserver <name>

Example

```
> set lb vserver mylbvserver -skippersistency ReLb
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 Done
> show lb vserver mylbvserver
 mylbvserver (*:*) - ANY Type: ADDRESS
   . . .
   . . .
 Skip Persistency: ReLb
   . . .
 Done
>
```

To override persistence settings for overloaded services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the virtual server of type UDP or ANY.
2. In the Advanced Settings pane, select Traffic Settings, and specify the type of Skip Persistency.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Nov 23, 2015

The statistics from the NetScaler VPX appliance indicate that the appliance has reached the session persistence limit. As a result, persistence sessions are failing. Is possible to increase the session persistence limit?

Cause: The NetScaler appliance has the system limit of 250,000 persistence session for a core.

Resolution: To resolve this issue, you can perform any of the following tasks:

- Reduce the time out value for persistence
- Increase the number of cores for the appliance

After configuring Cookie Insert persistence on the NetScaler appliance, the users report that the connections work fine for some time, but then start getting disconnected. What best practice should I follow when configuring persistence?

Cause: By default, the time-out value for Cookie Insert persistence is 120 seconds.

Resolution: When you configure persistence for applications for which idle time cannot be determined, set the Cookie Insert persistence time-out value to 0. With this setting, the connection does not time out.

After configuring an HTTP virtual server on the NetScaler appliance, I need to make sure that a user always connects to the same server for the requested content, so I configured SourceIP persistence. Now, increasing the time-out value for persistence introduces latency. How can I increase the timeout value without affecting performance?

Resolution: Consider using Cookie Insert persistence with the time-out value set to 0. This setting enables long-duration persistence settings, because the appliance does not specify a time for expiring the cookie.

After configuring Cookie Insert persistence on the NetScaler appliance, it works as expected when clients from the same time zone access the content. However, when a client from another time zone makes an attempt to connect, the connection is immediately timed out.

Cause: Time based Cookie Insert persistence works as expected when a client from the same time zone makes a connection. However, when the client machine and NetScaler appliance are in different time zones, the cookie is not valid. For example, when a client in EST time zone sends a cookie at 11:00 AM EST to a NetScaler appliance in the PST time zone, the appliance receives the cookie at 2:00 PM PST. As a result of the difference in time, the cookie is not valid, and the connection is immediately timed-out.

Resolution: Set the time-out value for Cookie Insert persistence to 0.

A NetScaler appliance is used to load balance application servers, such as Oracle Weblogic server. To make sure that clients get persistent connections to these servers, SourceIP persistence is configured. It works as expected when a connection is made from a computer. However, when thin clients attempt a connection through a terminal server and, as a result, the appliance receives requests from multiple clients from the same IP address (the terminal server IP address). Therefore, the connections from all thin clients are directed to the same application server. Is it possible to configure persistency for requests from individual thin clients based on the client IP address?

Cause: The NetScaler appliance receives requests from the terminal server and the source IP address of the request remains the same. As a result, the appliance cannot distinguish among the requests received from the thin clients and provide persistence according to the requests from thin clients.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Resolution: To avoid this problem, you can configure Rule persistence based on some unique parameter value for each thin client.

The NetScaler appliance is used to load balance Web Interface servers. When accessing the servers, the user receives the "State Error" error message. Additionally, when one of the Web Interface servers is shut down or not available, some of the users receive an error message.

Cause: Lack of persistence to the Web Interface servers can result in error messages when a user attempts to connect to the server.

Resolution: Citrix recommends that you specify the Cookie Insert persistence method on the NetScaler appliance when load balancing Web Interface servers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Customizing a Load Balancing Configuration

Aug 29, 2017

After you configure a basic load balancing setup, you can make a number of modifications to it so that it distributes load exactly as you need. The load balancing feature is complex. You can modify the basic elements by changing the load balancing algorithm, configuring load balancing groups and using them to create your load balancing configuration, configuring persistent client-server connections, configuring the redirection mode, and assigning different weights to different services that have different capacities.

The default load balancing algorithm on the NetScaler appliance is the least connection method, which configures the appliance to send each incoming connection to the service that is currently handling the fewest connections. You can specify different load balancing algorithms, each of which is suited to different conditions.

To accommodate applications such as shopping carts, which require that all requests from the same user be directed to the same server, you can configure the appliance to maintain persistent connections between clients and servers. You can also specify persistence for a group of virtual servers, causing the appliance to direct individual client requests to the same service regardless of which virtual server in the group receives the client request.

You can enable and configure the redirection mode that the appliance uses when redirecting user requests, choosing between IP-based and MAC-based forwarding. You can assign weights to different services, specifying what percentage of incoming load should be directed to each service, so that you can include servers with different capacities in the same load balancing setup without overloading the lower-capacity servers or allowing the higher-capacity servers to sit idle.

This section includes the following details:

- Customizing the Hash Algorithm for Persistence across Virtual Servers
- Configuring the Redirection Mode
- Configuring per-VLAN Wildcarded Virtual Servers
- Assigning Weights to Services
- Configuring the MySQL and Microsoft SQL Server Version Setting

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Customizing the Hash Algorithm for Persistence across Virtual Servers

Aug 29, 2013

The NetScaler appliance uses hash-based algorithms for maintaining persistence across virtual servers. By default, the hash-based load balancing method uses a hash value of the IP address and port number of the service. If a service is made available at different ports on the same server, the algorithm generates different hash values. Therefore, different load balancing virtual servers might send requests for the same application to different services, breaking the pseudo-persistence.

As an alternative to using the port number to generate the hash value, you can specify a unique hash identifier for each service. For a service, the same hash identifier value must be specified on all the virtual servers. If a physical server serves more than one type of application, each application type should have a unique hash identifier.

The algorithm for computing the hash value for a service works as follows:

- By default, a global setting specifies the use of port number in a hash calculation.
- If you configure a hash identifier for a service, it is used, and the port number is not, regardless of the global setting.
- If you do not configure a hash identifier, but change the default value of the global setting so that it does not specify use of the port number, the hash value is based only on the IP address of the service.
- If you do not configure a hash identifier or change the default value of the global setting to use the port number, the hash value is based on the IP address and the port number of the service.

You can also specify hash identifiers when using the NetScaler command line to bind services to a service group. In the configuration utility, you can open a service group and add hash identifiers on the Members tab.

To change the use-port-number global setting by using the command line interface

At the command prompt, type:

set lb parameter -usePortForHashLb (YES | NO)

Example

```
> set lb parameter -usePortForHashLb NO
 Done
>show lb parameter
Global LB parameters:
      Persistence Cookie HttpOnly Flag: DISABLED
      Use port for hash LB: NO
 Done
```

To change the use-port-number global setting by using the configuration utility

1. Navigate to Traffic Managment > Load Balancing > Configure Load Balancing parameters.
2. Select or clear Use Port for Hash Based LB Methods.

To create a new service and specify a hash identifier for a service by using the command line interface

At the command prompt, type the following commands to set the hash ID and verify the setting:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add service < name > (< ip > |< serverName >) < serviceType > < port > -hashId < positive_integer >

show service <name>

Example

```
> add service flbkng 10.101.10.1 http 80 -hashId 12345
 Done
>show service flbkng
      flbkng (10.101.10.1:80) - HTTP
      State: DOWN
      Last state change was at Thu Nov  4 10:14:52 2010
      Time since last state change: 0 days, 00:00:15.990
      Server Name: 10.101.10.1
      Server ID : 0   Monitor Threshold : 0

      Down state flush: ENABLED
      Hash Id: 12345

1)     Monitor Name: tcp-default
            State: DOWN     Weight: 1

 Done
```

To specify a hash identifier for an existing service by using the command line interface

Type the set service command, the name of the service, and **-hashID** followed by the ID value.

To specify a hash identifier while adding a service group member

To specify a hash identifier for each member to be added to the group and verify the setting, at the command prompt, type the following commands (Be sure to specify a unique hashID for each member.):

bind servicegroup <serviceGroupName> <memberName> <port> -hashId <positive_integer>

show servicegroup <serviceGroupName>

Example

```
 bind servicegroup http_svc_group 10.102.27.153 80 -hashId 2222222

>show servicegroup SRV
      SRV - HTTP
      State: ENABLED  Monitor Threshold : 0
      …

      1)       1.1.1.1:80   State: DOWN     Server Name: 1.1.1.1    Server ID: 123  Weight: 1
      Hash Id: 32211

            Monitor Name: tcp-default       State: DOWN
      …
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    2)      2.2.2.2:80   State: DOWN     Server Name: 2.2.2.2    Server ID: 123  Weight: 1
    Hash Id: 12345

        Monitor Name: tcp-default       State: DOWN
    ...
Done
```

To specify a hash identifier for a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Create a new service, or open an existing service and specify the hash ID.

To specify a hash identifier for an already configured service group member by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Open a member and type a unique hash ID.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Redirection Mode

May 11, 2017

The redirection mode configures the method used by a virtual server to determine where to forward incoming traffic. The NetScaler appliance supports the following redirection modes:

- IP-Based forwarding (the default)
- MAC-Based forwarding

You can configure MAC-Based forwarding on networks that use direct server return (DSR) topology, link load balancing, or firewall load balancing. For more information on MAC-Based forwarding, see Configuring MAC-Based Forwarding.

To configure the redirection mode by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -m <RedirectionMode>
```

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

To configure the redirection mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server and select the redirection mode.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring per-VLAN Wildcarded Virtual Servers

Nov 07, 2017

If you want to configure load balancing for traffic on a specific virtual local area network (VLAN), you can create a wildcarded virtual server with a listen policy that restricts it to processing traffic only on the specified VLAN.

To configure a wildcarded virtual server that listens to a specific VLAN by using the command line interface

At the command prompt, type the following commands to configure a wildcarded virtual server that listens to a specific VLAN and verify the configuration:

- add lb vserver <name> <serviceType> IPAddress * Port * -listenpolicy <expression> [-listenpriority <positive_integer>]
- show vserver

Example

```
add lb vserver Vserver-LB-vlan1 ANY -listenpolicy "CLIENT.VLAN.ID.EQ(2)" -listenpriority 10

show vserver Vserver-LB-vlan1
```

To configure a wildcarded virtual server that listens to a specific VLAN by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Create a new virtual server or open an existing virtual server.
3. Specify a listen policy priority and expression.

After you have created this virtual server, you bind it to one or more services as described in Setting Up Basic Load Balancing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Assigning Weights to Services

Nov 12, 2013

In a load balancing configuration, you assign weights to services to indicate the percentage of traffic that should be sent to each service. Services with higher weights can handle more requests; services with lower weights can handle fewer requests. Assigning weights to services allows the NetScaler appliance to determine how much traffic each load balanced server can handle, and therefore more effectively balance load.

Note: If you use a load balancing method that supports weighting of services (for example, the round robin method), you can assign a weight to the service.

The following table describes the load balancing methods that support weighting, and briefly describes the manner in which weighting affects how a service is selected for each one.

Load Balancing Methods	**Service Selection with Weights**
Round Robin	The virtual server prioritizes the queue of available services such that services with the highest weights come to the front of the queue more frequently than those with the lowest weights and receive proportionately more traffic. For a complete description, see The Round Robin Method.
Least Connection	The virtual server selects the service with the best combination of fewest active transactions and highest weight. For a complete description, see The Least Connection Method.
Least Response Time and Least Response Time Method using Monitors	The virtual server selects the service with the best combination of fewest active transactions and fastest average response time. For a complete description, see The Least Response Time Method.
Least Bandwidth	The virtual server selects the service with the best combination of least traffic and highest bandwidth. For a complete description, see The Least Bandwidth Method.
Least Packets	The virtual server selects the service with the best combination of fewest packets and highest weight. For a complete description, see The Least Packets Method.
Custom Load	The virtual server selects the service with the best combination of lowest load and highest weight. For a complete description, see The Custom Load Method.
Hashing methods and Token method	Weighting is not supported by these load balancing methods.

To configure a virtual server to assign weights to services by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

set lb vserver <name> -weight <Value> <ServiceName>

Example

set lb vserver Vserver-LB-1 -weight 10 Service-HTTP-1

To configure a virtual server to assign weights to services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open the virtual server, and then click in the Services section.
3. In the weight column for the service, assign a weight to the service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the MySQL and Microsoft SQL Server Version Setting

Mar 24, 2015

You can specify the version of Microsoft® SQL Server® and the MySQL server for a load balancing virtual server that is of type MSSQL and MySQL respectively. The version setting is recommended if you expect some clients to not be running the same version as your MySQL or Microsoft SQL Server product. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version.

To set the Microsoft SQL Server version parameter by using the command line interface

At the command prompt, type the following commands to set the Microsoft SQL Server version parameter for a load balancing virtual server and verify the configuration:

- set lb vserver <name> -mssqlServerVersion <mssqlServerVersion>
- show lb vserver <name>

Example

```
> set lb vserver myMSSQLvip -mssqlServerVersion 2008R2
 Done
> show lb vserver myMSSQLvip
 myMSSQLvip (190.0.2.12:1433) - MSSQL Type: ADDRESS
 . . .
 . . .
 MSsql Server Version: 2008R2
 . . .
 . . .
 Done
>
```

To set the MySQL Server version parameter by using the command line interface

At the command prompt, type the following commands to set the MySQL Server version parameter for a load balancing virtual server and verify the configuration:

- set lb vserver <name> -mysqlServerVersion <string>
- show lb vserver <name>

Example

```
> set lb vserver mysqlsvr -mysqlserverversion 5.5.30
 Done
> sh lb vserver mysqlsvr
  mysqlsvr (2.22.2.222:3306) - MYSQL      Type: ADDRESS
 . . .
 . . .
  Mysql Server Version: 5.5.30
 . . .
 . . .
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done
>
```

To set the MySQL or Microsoft SQL Server version parameter by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server of type MySQL or MSSQL, and set the server version.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Diameter Load Balancing

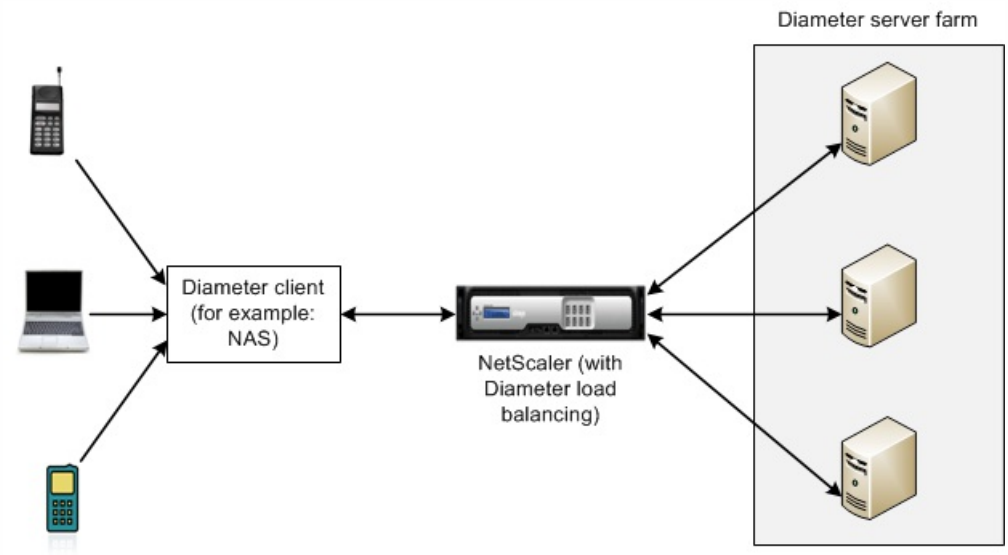
Oct 21, 2015

The Diameter protocol is a next generation Authentication, Authorization, and Accounting (AAA) signaling protocol used mainly on mobile devices such as laptops and mobile phones. It is a peer-to-peer protocol, as opposed to the traditional client-server model used by most other protocols. However, in most Diameter deployments, the clients originates the request and the server responds to the request.

When Diameter messages are exchanged, the Diameter server usually does much more processing than does the Diameter client. With the increase in control plane signaling volume, the Diameter server becomes a bottleneck. Therefore, Diameter messages must be load balanced to multiple servers. A virtual server performing load balancing of Diameter messages provides the following benefits:

- Lighter load on Diameter servers, which translates to faster response time to end users.
- Server health monitoring and better failover capabilities.
- Better scalability in terms of server addition without changing client configuration.
- High availability.
- SSL-Diameter offloading.

The following figure shows a Diameter system in a NetScaler deployment:



A Diameter system has the following components:

- **Diameter client.** Supports Diameter client applications in addition to the base protocol. Diameter clients are often implemented in devices situated at the edge of a network and provide access control services for that network. Typical examples of Diameter clients are a Network Access Server (NAS) and the Mobile IP Foreign Agent (FA).
- **Diameter agent.** Provides relay, proxy, redirect, or translation services. The NetScaler appliance (configured with a Diameter load balancing virtual server) plays the role of a Diameter agent.
- **Diameter server.** Handles the authentication, authorization, and accounting requests for a particular realm. A Diameter server must support Diameter server applications in addition to the base protocol.

In a typical Diameter topology, when an end-user device (such as a mobile phone) needs a service, it sends a request to a

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Diameter client. Each Diameter client establishes a single connection (TCP connection—SCTP is not yet supported) with a Diameter server as specified by the Diameter base-protocol RFC 6733. The connection is long-lived and all messages between the two Diameter nodes (client and server) are exchanged over this connection. The NetScaler uses message based load balancing .

Example

A mobile service provider uses Diameter for its billing system. When a subscriber uses a prepaid number, the Diameter client repeatedly sends requests to the server to check the available balance. The Diameter protocol establishes a connection between the client and the server, and all requests are exchanged over that connection. Connection based load balancing would be pointless, because there is only one connection. However, with the large number of messages on the connection, message based load balancing expedites the process of billing the prepaid mobile subscriber.

How Diameter Load Balancing Works

A Diameter client opens a connection to the NetScaler appliance and sends a Diameter capability exchange (CER) message. Diameter messages are composed of command codes and each command has a set of Attribute-Value Pairs (AVPs), such as Origin-Host and Host-IP-Address.

The NetScaler selects a Diameter server, opens a connection to the server, and forwards the CER message to the server. The server reads the client identity and determines that it is directly connected to the client.

The Diameter server prepares the Diameter handshake reply and sends it to the NetScaler appliance. The appliance modifies the handshake and inserts its own identity. At this point, the Diameter client determines that it is directly connected to the NetScaler (the agent).

Note: Until the Diameter handshake is complete, all Diameter request messages from the client are queued on the selected server. The packets are forwarded to the server when the handshake is complete.

Load Balancing Diameter Traffic

When a client sends a request to the NetScaler appliance, the appliance parses the request and contextually load balances it to a Diameter server on the basis of a persist AVP. The NetScaler has advertised the client identity to the server, so it does not add route entries, because the server is expecting messages directly from client.

Server initiated requests are not as frequent as client requests. Server initiated requests are similar to client initiated requests, except:

- Since messages are received from multiple servers, the NetScaler maintains the transaction state by adding a unique Hop by Hop (HbyH) number to each forwarded request message. When the message response arrives (with same HbyH number), the appliance translates this HbyH number to the HbyH number that was received on the server when the request arrived.
- NetScaler adds a route entry by putting its identity, because the client sees the NetScaler as a relay agent.

Note: If a Diameter message spans more than one packet, the NetScaler accumulates the packets in an incomplete header queue and forwards them to the server when the full message is accumulated. Similarly, if a single packet contains more than one Diameter message, the NetScaler splits the packet and forwards the messages to servers as determined by the load balancing virtual server.

Disconnecting a Session

A Disconnect Peer Request (DPR) indicates the peer's intention of closing the connection, with the reason for closing the connection. The peer replies with a DPA (TCP always provides successful DPA).

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- When the NetScaler receives a DPR from the client, it broadcasts the DPR to all servers and immediately replies with a DPA to the client. The servers reply with DPAs, but the NetScaler ignores them. The client sends a FIN, which the NetScaler broadcasts to all servers.
- When the NetScaler receives a DPR from the server, it replies with a DPA to that server alone, and does not remove the server from the reuse pool. When the server sends a FIN, the NetScaler replies with FIN/ACK and removes connections from the reuse pool.
- If the NetScaler receives a FIN from the client, it sends the client a FIN/ACK, broadcasts the FIN, and immediately removes the server connection from the reuse pool.
- If the NetScaler receives a FIN from the server, it sends a FIN/ACK and removes it from reuse pool. Any new message for this server is sent on a new connection.

Configuring Load Balancing for Diameter Traffic

To configure the NetScaler appliance to load balance Diameter traffic, you must first set the Diameter parameters on the appliance, then add the Diameter monitor, add the Diameter services, bind the services to the monitor, add the Diameter load balancing virtual server, and bind the services to the virtual server.

To configure load balancing for Diameter traffic by using the command line interface

1. Configure the Diameter parameters.
   set ns diameter -identity <string> -realm <string> -serverClosePropagation <YES | NO>

   Example
   set ns diameter -identity mydomain.org -realm org -serverClosePropagation YES
2. Add a Diameter monitor.
   add lb monitor <monitorName> DIAMETER -originHost <string> -originRealm <string>

   Example
   add lb monitor diameter_mon DIAMETER -originHost mydomain.org -originRealm org
3. Create the Diameter services.
   add service <name> <IP> DIAMETER <port>

   Example
   add service diameter_svc0 10.102.82.86 DIAMETER 3868
   add service diameter_svc1 10.102.82.87 DIAMETER 3868
   add service diameter_svc2 10.102.82.88 DIAMETER 3868
   add service diameter_svc3 10.102.82.89 DIAMETER 3868
4. Bind the Diameter services to the Diameter monitor.
   bind service <name>@ monitorName <monitorName>

   -**Example**
   bind service diameter_svc0 -monitorName diameter_mon
   bind service diameter_svc1 -monitorName diameter_mon
   bind service diameter_svc2 -monitorName diameter_mon
   bind service diameter_svc3 -monitorName diameter_mon
5. Add a Diameter load balancing virtual server with Diameter persistence.
   add lb vserver <name> DIAMETER <IPAddress> <port> -persistenceType DIAMETER -persistAVPno <positive_integer>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

add lb vserver diameter_vs DIAMETER 10.102.112.152 3868 -persistenceType DIAMETER -persistAVPno 263

6. Bind the Diameter services to the Diameter load balancing virtual server.
   bind lb vserver <name> <serviceName>

   Example

   bind lb vserver diameter_vs diameter_svc0
   bind lb vserver diameter_vs diameter_svc1
   bind lb vserver diameter_vs diameter_svc2
   bind lb vserver diameter_vs diameter_svc3
7. Save the configuration.
   save ns config

   Note: You can also configure load balancing of Diameter traffic over SSL by using the **SSL_DIAMETER** service type.

To configure load balancing for Diameter traffic by using the configuration utility

1. Navigate to System > Settings > Change Diameter Parameters and set the diameter parameters.
2. Navigate to Traffic Management > Load Balancing > Virtual Servers, and create a load balancing virtual server of type Diameter.
3. Create a service of type Diameter.
4. Create a monitor of type Diameter. In Special parameters, set the origin host and origin realm.
5. Bind the monitor to the service, and bind the service to the Diameter virtual server.
6. In Advanced Settings, click Persistence, specify Diameter and enter a persistence AVP number.
7. Click Save, and click Done.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring FIX Load Balancing

Nov 15, 2016

Financial Information eXchange (FIX) protocol is an open message standard used in financial industry for electronic exchange of information related to securities transaction between trading partners. FIX protocol is used extensively by buy-side and sell-side firms, trading platforms, and regulators for communicating trade information.

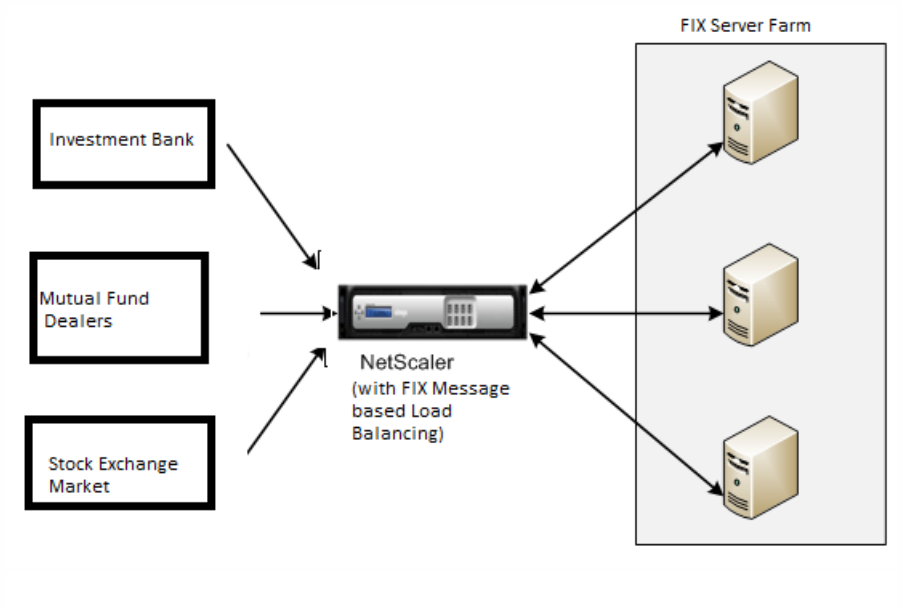
This feature enables you to configure a FIX load balancing virtual server to distribute incoming FIX messages across multiple FIX servers. NetScaler supports FIX message based load balancing (MBLB) for FIX 4.1, FIX 4.2, FIX 4.3 and FIX 4.4 versions.

FIX MBLB on a NetScaler appliance provides the following benefits:

1. Efficient management of FIX servers with superior health monitoring.
2. SYN protection for all FIX servers.
3. FIX session persistence.

How FIX Load Balancing Works

A FIX MBLB setup includes a FIX load-balancing virtual server and multiple load-balanced FIX servers. The FIX virtual server receives incoming client traffic, parses the incoming traffic into FIX messages, selects a FIX server for each FIX message and forwards the message to the selected FIX server. The following conceptual drawing illustrates a typical FIX load balancing set up.



In a basic FIX MBLB setup, the FIX virtual server distributes FIX messages coming from clients to the load-balanced FIX servers using the round robin load-balancing method. With persistence of type FIXSESSION enabled, the FIX virtual server selects the same server for different FIX messages belonging to the same FIX session. The FIX session is determined based on the values of FIX fields SenderCompID (tag 49) and TargetCompID (tag 56).

Configuring and Monitoring Load Balancing for FIX Traffic

Following are the configurations that you must do to load balance FIX message traffic:

1. Configure a FIX load balancing virtual server.
2. Configure FIX load balancing services.
3. Configure FIXSESSION persistence.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Set the persistence timeout.
5. Displaying FIX stats
6. Monitoring FIX persistent sessions

To configure a FIX load balancing server by using the command line interface

At the command prompt, type:

add load balancing virtual server

```
add lb vserver <name> FIX <IP> <PORT>
```

Example

```
add lb vserver vs1 FIX 10.102.82.86 3868
```

To configure a FIX service by using the command line interface

At the command prompt, type:

Add Service

add service <name> <ip-addr> **FIX** <port>

Example

```
add service_svc1 10.102.82.86 FIX 3868
```

To configure FIXSESSION persistence by using the command line interface

At the command prompt, type:

configure FIXSESSION persistence

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set lb vserver <name> -persistenceType FIXSESSION

Example

set lb vserver vs1 -persistenceType FIXSESSION
```

To set persistence timeout by using the command line interface

At the command prompt, type:

```
configure persistence timeout

set lb vserver <name> -timeout <value>

Example

set lb vserver vs1 –timeout 2
```

To display FIX stats by using the command line interface

At the command prompt, type:

```
To display FIX stats

stat lb vserver <name>

Example

stat lb vserver_svc1
```

To bind FIX service to FIX virtual server by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

To bind FIX service to FIX virtual server by using the command line interface

```
bind lb vserver <name> <service name>
```

Example

```
bind lb vserver vs1 svc1
```

To display FIX persistent sessions by using the command line interface

At the command prompt, type:

Display FIX persistent sessions

```
show lb persistentSessions <name>
```

Example

```
show lb persistentSessions vs1
```

To configure FIX load balancing virtual server by using the NetScaler GUI

1. Navigate to the **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page and click **Add** to create a FIX Load Balancing virtual server.
2. On the **Load Balancing Virtual Server** page, set the server parameters:
   1. Virtual Server Name
   2. Protocol type as "FIX"
   3. Server IP Address Type
   4. Server IP Addres
   5. Server Port Number
3. Click **OK** and **Continue** to set additional parameters.
4. In the **Services** section, select or add a new FIX load balancing virtual service, and bind it to the FIX server.
5. In the **Persistence** section, set the following parameters:
   1. Persistence type as 'FIXSESSION'
   2. Time-out interval
6. Click **OK** and then **Done**.

To edit a FIX load balancing virtual server by using the NetScaler GUI

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page, select a FIX server and click **Edit**.

To delete a FIX load balancing virtual server by using the NetScaler GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page, select a FIX server and click **Delete**.

To configure FIX Load Balancing Virtual Service by using the NetScaler GUI

1. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** page and click **Add** to create a FIX Load Balancing virtual service.
2. On the **Services** page, set the following parameters. You can click the 'More' arrow to set additional parameters such as Traffic Domain, Hash ID, Server ID, Cache Type, and Number of Active Connections.
   1. Service Name – FIX Virtual Service Name
   2. Choose Virtual Server type as (New or Existing)
   3. Protocol – Protocol Type as 'FIX'
   4. Server – Virtual Server IP address
   5. Port – Server Port Number
3. Click **OK** and **Continue** to set other parameters such as Monitors, Threshold & Timeout, Profiles, and Policies.
4. Click **OK** and then **Done**.

To edit a FIX load balancing virtual service by using the NetScaler GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** page, select a **FIX service** and click **Edit**.

To delete a FIX load balancing virtual service by using the NetScaler GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** page, select a FIX service and click **Delete**.

To display FIX load balancing server statistics

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page and then click **Statistics** to display the FIX server statistics.

To display Persistent sessions for a FIX server by using the NetScaler GUI

Navigate to **Configuration** > **Traffic Management** page and, under **Monitor Sessions** click **Virtual Server Persistent Sessions**.

To clear Persistent sessions for a FIX server by using the NetScaler GUI

1. Navigate to **Configuration** > **Traffic Management** page and, under **Monitor Sessions** click **Clear Persistent Sessions**.
2. On the **Clear Persistent Sessions** page, set the following parameters:
   1. Virtual Server – Choose a FIX virtual server
   2. Persistence Parameter – Choose a FIX persistence parameter
3. Click **OK**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Protecting a Load Balancing Configuration against Failure

Jun 03, 2015

When a load balancing virtual server fails, or when the virtual server is unable to handle excessive traffic, the load balancing setup can fail. You can protect your load balancing setup against failure by configuring the NetScaler appliance to redirect excess traffic to an alternate URL, configuring a backup load balancing virtual server, and configuring stateful connection failover.

To protect a load balancing configuration against failure, see the following sections:

- Redirecting Client Requests to an Alternate URL
- Configuring a Backup Load Balancing Virtual Server
- Configuring Spillover
- Connection Failover
- Flushing the Surge Queue

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Redirecting Client Requests to an Alternate URL

Sep 02, 2013

In the event that a load balancing virtual server of type HTTP or type HTTPS goes DOWN or is disabled, you can redirect requests to an alternate URL by using an HTTP 302 redirect. The alternate URL can provide information about the status of the server.

You can redirect to a page on the local server or a remote server. You can redirect to a relative URL or an absolute URL. If you configure a redirect to a relative URL consisting of a domain name with no path, the NetScaler appliance appends the path of the incoming URL to the domain. If you use an absolute URL, the HTTP redirect is sent to that URL with no modification.

Note: If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect is used only when both the primary and backup virtual servers are DOWN.

To configure a virtual server to redirect the client request to a URL by using the command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -redirectURL <URLValue>
```

Example

```
set lb vserver Vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
```

To configure a virtual server to redirect the client request to a URL by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Protection, and specify a redirect URL.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Backup Load Balancing Virtual Server

Sep 02, 2013

You can configure the NetScaler appliance to direct requests to a backup virtual server in the event that the primary load balancing virtual server is DOWN or unavailable. The backup virtual server is a proxy and is transparent to the client. The appliance can also send a notification message to the client regarding the site outage.

You can configure a backup load balancing virtual server when you create it, or you can change the optional parameters of an existing virtual server. You can also configure a backup virtual server for an existing backup virtual server, thus creating cascading backup virtual servers. The maximum depth of cascading backup virtual servers is 10.

If you have multiple virtual servers that connect to two servers, you have a choice for what happens if the primary virtual server goes DOWN and then comes back up. The default behavior is for the primary virtual server to resume its role as primary. However, you may want to configure the backup virtual server to remain in control in the event that it takes over. For example, you may want to sync updates on the backup virtual server to the primary virtual server and then manually force the original primary server to resume its role. In this case, you can designate the backup virtual server to remain in control in the event that the primary virtual server goes DOWN and then comes back up.

You can configure a redirect URL on the primary load balancing virtual server as a fallback for when both the primary and the backup virtual servers are DOWN or have reached their threshold for handling requests. When services bound to virtual servers are OUT OF SERVICE, the appliance uses the redirect URL.

Note: If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect is used only when the primary and backup virtual servers are down.

To set a backup virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -backupVserver <BackupVServerName> [-disablePrimaryOnDown]
```

Example

```
set lb vserver Vserver-LB-1 -backupVserver Vserver-LB-2 -disablePrimaryOnDown
```

To set a backup virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Protection, and select a backup virtual server.
3. If you want the backup virtual server to remain in control until you manually enable the primary virtual server even if the primary virtual server comes back up, select Disable Primary When Down.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Spillover

Sep 02, 2013

A spillover configuration on the appliance consists of a primary virtual server that is configured with a spillover method, a spillover threshold, and a backup virtual server. Backup virtual servers can also be configured for spillover, creating a chain of backup virtual servers.

The spillover method specifies the operational condition on which you want to base your spillover configuration (for example, the number of established connections, bandwidth, or combined health of the server farm). When a new connection arrives, the appliance verifies that the primary virtual server is up and compares the operational condition with the configured spillover threshold. If the threshold is reached, the spillover feature diverts new connections to the first available virtual server in the backup chain. The backup virtual server manages the connections it receives until the load on the primary falls below the threshold.

If you configure spillover persistence, the backup virtual server continues to process the connections it received, even after the load on the primary falls below the threshold. If you configure spillover persistence and a spillover persistence timeout, the backup virtual server processes connections for only the specified period of time after the load on the primary falls below the threshold.

Note: In most cases, spillover is triggered if the value associated with the spillover method exceeds the threshold (for example, number of connections). Keep in mind, however, that with the server-health spillover method, spillover is triggered if the health of the server farm falls below the threshold.
You can configure spillover in one of the following ways:

- Specify a predefined spillover method. Four predefined methods are available, and they fulfill common spillover requirements.
- Configure policy based spillover. In policy based spillover, you use a NetScaler rule to specify the conditions that should be met for spillover to occur. NetScaler rules give you the flexibility to configure spillover for various operational conditions.

Use policy based spillover if a predefined method does not satisfy your requirements. If you configure both for a primary virtual server, the policy based spillover configuration takes precedence over the predefined method.

First, you create the primary virtual server and the virtual servers that you need for the backup chain. You set up the backup chain by specifying one virtual server as the backup for the primary (that is, you create a secondary virtual server), a virtual server as the backup for the secondary (that is, you create a tertiary virtual server), and so on. Then, you configure spillover by either specifying a predefined spillover method or creating and binding spillover policies.

For instructions for assigning a virtual server as the backup for another virtual server, see Configuring a Backup Load Balancing Virtual Server.

Configuring a Predefined Spillover Method

Updated: 2013-09-02

Predefined spillover methods fulfill some of the more common spillover requirements. To use one of the predefined spillover methods, you configure spillover parameters on the primary virtual server. To create a chain of backup virtual servers, you also configure spillover parameters on backup virtual servers.

If the backup virtual servers reach their own threshold values, and the service type is TCP, the NetScaler appliance sends clients a TCP reset. For service types HTTP, SSL, and RTSP , it diverts new requests to the redirect URL configured for the primary virtual server. A redirect URL can be specified for only HTTP, SSL, and RTSP virtual servers. If a redirect URL is not configured, the NetScaler appliance sends clients a TCP reset (if the virtual server is of type TCP) or an HTTP 503 response (if the virtual server is of type HTTP or SSL).

Note: With RTSP virtual servers, the NetScaler appliance uses only data connections for spillover. If the backup RTSP virtual server is not available, the requests are redirected to an RTSP URL and an RTSP redirect message is sent to the client.

To configure a predefined spillover method for a virtual server by using the command line interface

At the command prompt, type:

set lb vserver <vServerName> -soMethod <spillOverType> -soThreshold <positiveInteger> -soPersistence ENABLED -soPersistenceTimeout <positiveInteger>

Example

```
set lb vserver Vserver-LB-1 -soMethod Connection -soThreshold 1000 -soPersistence enabled -soPersistenceTimeout 2
```

To configure a predefined spillover method for a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Protection, and set the spillover parameters.

Configuring Policy Based Spillover

Spillover policies, which are based on rules (expressions), enable you to configure the appliance for a wider range of spillover scenarios. For example, you can configure spillover on the basis of the virtual server's response time, or on the basis of number of connections in the virtual server's surge queue.

To configure policy based spillover, first create a spillover action. You then select the expression that you want to use in the spillover policy , configure the policy, and associate the action with it. Finally, you bind the spillover policy to a load balancing, content switching, or global server load balancing virtual server. You can bind multiple spillover policies to a virtual server, with priority numbers. The appliance evaluates the spillover policies in ascending order of priority numbers and performs the action associated with the last policy to evaluate to TRUE.

A virtual server can also have a backup action. The backup action is performed if the virtual server does not have one or more backup virtual servers, or if all of the backup virtual servers are DOWN, disabled, or have reached their own spillover limits.

When a spillover policy results in an UNDEF condition (an exception thrown when the result of policy evaluation is undefined), an UNDEF action is performed. The UNDEF action is always ACCEPT. You cannot specify an UNDEF action of your choice.

Configuring a Spillover Action

A spillover action is performed when the spillover policy with which it is associated evaluates to TRUE. Currently, SPILLOVER is the only supported spillover action.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure policy based spillover by using the command line interface

At the command prompt, type the following commands to configure a spillover policy and verify the configuration:

- add spillover action <name> -action SPILLOVER
- show spillover action <name>

Example

```
> add spillover action mySoAction -action SPILLOVER
 Done
> show spillover action mySoAction
1) Name:  mySoAction       Action:  SPILLOVER
 Done
>
```

Selecting an Expression for the Spillover Policy

In the policy expression, you can use any virtual-server based expression that returns a Boolean value. For example, you could use one of the following expressions:

SYS.VSERVER("vserver").RESPTIME.GT(<int>), SYS.VSERVER("vserver").STATE.EQ("<string>"), and SYS.VSERVER("vserver").THROUGHPUT.LT(<int>).

In addition to the existing functions such as RESPTIME, STATE, and THROUGHPUT, you can use the following virtual server based functions that have been introduced with this feature:

Averagesurgecount

Returns the average number of requests in the surge queues of active services. Returns 0 (zero) if there are no active services. Raises an UNDEF condition if used with a content switching or global server load balancing virtual server.

Activeservices

Returns the number of active services. Raises an UNDEF condition if used with a content switching or global server load balancing virtual server.

Activetransactions

Returns the value of the virtual-server-level counter for current active transactions.

is_dynamic_limit_reached

Returns a Boolean TRUE if the number of connections being managed by the virtual server equals the dynamically calculated threshold. The dynamic threshold is the sum of the maximum client (Max Clients) settings of the bound services that are UP.

You can use a policy expression to implement any of the predefined spillover methods. The following table maps the predefined spillover methods to the expressions you can use to implement them:

Table 1. Converting predefined spillover methods to policy expressions

Predefined spillover method	Corresponding expression
CONNECTION	SYS.VSERVER(\"<vserver-name>\").CONNECTIONS, used with the GT(int) arithmetic function.
BANDWIDTH	SYS.VSERVER(\"<vserver-name>\").THROUGHPUT, used with the GT(int) arithmetic function.
HEALTH	SYS.VSERVER(\"<vserver-name>\").HEALTH, used with the LT(int) arithmetic function.
DYNAMICCONNECTION	SYS.VSERVER(\"<vserver-name>\").IS_DYNAMIC_LIMIT_REACHED Note: If you implement policy based spillover by using the IS_DYNAMIC_LIMIT_REACHED function, you must also configure the predefined DYNAMICCONNECTION method for the virtual server, so that statistics required for spillover to work are collected.

Configuring a Spillover Policy

A spillover policy uses a Boolean expression as a rule to specify the conditions that must be met for spillover to occur.

To configure a spillover policy by using the command line interface

At the command prompt, type the following commands to configure a spillover policy and verify the configuration:

- add spillover policy <name> -rule <expression> -action <string> [-comment <string>]
- show spillover policy <name>

Example

```
> add spillover policy mySoPolicy -rule SYS.VSERVER("v1").RESPTIME.GT(50) -action mySoAction -comment "Triggers spillover when the vserver's response time is greater than 50 ms."
 Done
> show spillover policy mySoPolicy
1) Name:  mySoPolicy       Rule:  "SYS.VSERVER(\"v1\").RESPTIME.GT(50)" Action:  mySoAction      Hits:  0              ActivePolicy:  0
 Comment:  "Triggers spillover when the vserver\'s response time is greater than 50 ms."
 Done
>
```

Binding a Spillover Policy to a Virtual Server

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You can bind a spillover policy to load balancing, content switching, or global server load balancing virtual servers). You can bind multiple policies to a virtual server, with Goto expressions controlling the flow of evaluation.

To bind a spillover policy to a virtual server by using the command line interface

At the command prompt, type the following commands to bind a spillover policy to a load balancing, content switching, or global server load balancing virtual server and verify the configuration:

- bind (lb | cs | gslb) vserver <name> -policyName <string> -priority <positive_integer> [-gotoPriorityExpression <expression>]
- show (lb | cs | gslb) vserver <name>

Example

```
> bind lb vserver vserver1 -policyName mySoPolicy -priority 5
 Done
> show lb vserver vserver1
 vserver1 (2.2.2.12:80) - HTTP Type: ADDRESS
 . . .

1) Spillover Policy Name: mySoPolicy Priority: 5
 GotoPriority Expression: END
 Flowtype: REQUEST
 Done
>
```

Configuring a Backup Action for a Spillover Event

A backup action specifies what to do in the event that the spillover threshold is reached but one or more backup virtual serves are either not configured or are down, disabled, or have reached their own thresholds.

Note: For the predefined spillover methods that are configured directly on the virtual server (as values of the Spillover Method parameter), the backup action is not configurable. By default, the appliance sends clients a TCP reset (if the virtual server is of type TCP) or an HTTP 503 response (if the virtual server is of type HTTP or SSL).
The backup action is configured on the virtual server. You can configure the virtual server to accept requests (after the threshold specified by the policy is reached), redirect clients to a URL, or simply drop requests until the number of requests falls below the threshold.

To configure a backup action for spillover by using the command line interface

At the command prompt, type the following commands to configure a backup action and verify the configuration:

- set lb vserver <name> -soBackupAction <soBackupAction>
- show lb vserver <name>

Example

```
> set lb vserver vs1 -soBackupAction REDIRECT -redirectURL http://www.mysite.com/maintenance
 Done
> show lb vserver vs1
 vs1 (10.102.29.76:80) - HTTP Type: ADDRESS
 State: UP
 . . .
 Redirect URL: http://www.mysite.com/maintenance
          . . .
Done
>
```

To configure a backup action for spillover by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Protection, and then specify a spillover backup action.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Connection Failover

Nov 10, 2016

Connection failover helps prevent disruption of access to applications deployed in a distributed environment. In a NetScaler High Availability (HA) setup, *connection failover* (or *connection mirroring-CM*) refers to keeping active an established TCP or UDP connection when a failover occurs. The new primary NetScaler appliance has information about the connections established before the failover and continues to serve those connections. After failover, the client remains connected to the same physical server. The new primary appliance synchronizes the information with the new secondary appliance by using the SSF framework. If the L2Conn parameter is set, Layer 2 connection parameters are also synchronized with the secondary.

You can set up connection failover in either stateless or stateful mode. In the stateless connection failover mode, the HA nodes do not exchange any information about the connections that are failed over. This method has no runtime overhead.

In the stateful connection failover mode, the primary appliance synchronizes the data of the failed-over connections with the new secondary appliance.

Connection failover is helpful if your deployment has long lasting connections. For example, if you are downloading a large file over FTP and a failover occurs during the download, the connection breaks and the download is aborted. However, if you configure connection failover in stateful mode, the download continues even after the failover.

How Connection Failover Works on NetScaler Appliances

In stateless connection failover, the new primary appliance tries to re-create the packet flow according to the information contained in the packets it receives.

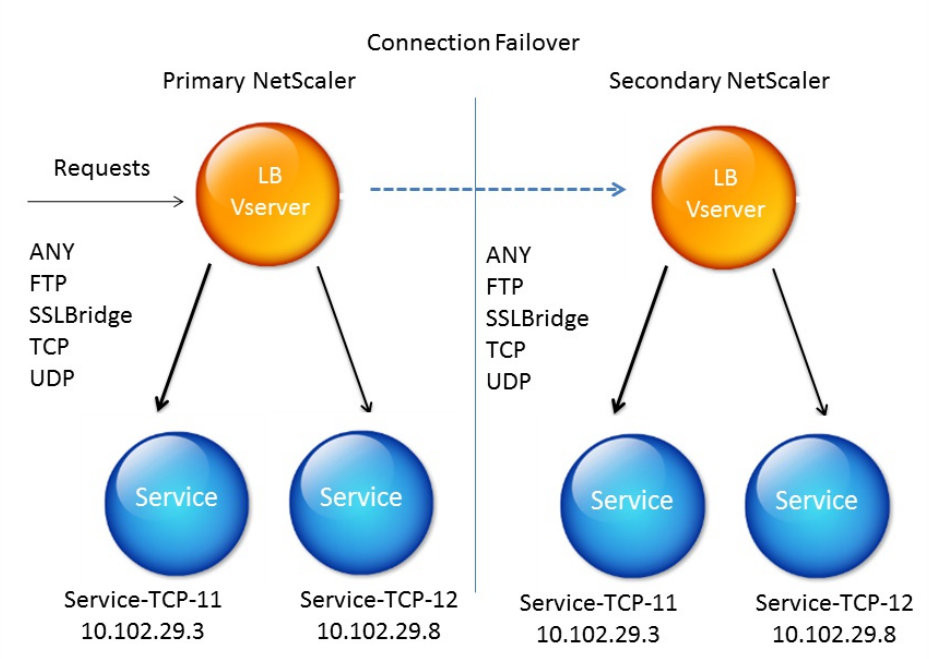
In stateful failover, to maintain current information about the mirrored connections, the primary appliance sends messages to the secondary appliance. The secondary appliance maintains the data related to the packets but uses it only in the event of a failover. If a failover occurs, the new primary (old secondary) appliance starts using the stored data about the mirrored connections and accepting traffic. During the transition period, the client and server may experience a brief disruption and retransmissions.

Note:
Verify that the primary appliance is able to authorize itself on the secondary appliance. To verify correct configuration of the passwords, use the show rpcnode command from command line or use the RPC option of the Network menu from the configuration utility.

A basic HA configuration with connection failover contains the entities shown in the following figure.

Figure 1. Connection Failover Entity Diagram

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Note

Connection failover is not supported after either of the following events:

- An upgrade to a later release.
- An upgrade to a later build within the same release, if the new build uses a different HA version.

Supported Setup

Connection failover can be configured only on load balancing virtual servers. It cannot be configured on content switching virtual servers. If you enable connection failover on load balancing virtual servers that are attached to a content switching virtual server, connection failover will not work because the load balancing virtual servers do not initially accept the traffic.

The following table describes the setup supported for connection failover.

Table 1. Connection Failover - Supported Setup

Setting	Stateless	Stateful
Service type	ANY.	ANY, UDP, TCP, FTP, SSL_BRIDGE.
Load balancing methods	All methods supported for the service type ANY. However, if Source IP persistence is not set, the SRCIPSRCPORTHASH method must be used.	All methods applicable to the supported service types.
Persistence types	SOURCEIP persistence.	All types applicable to the supported service types are supported.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting USIP	Stateless Must be ON.	Stateful No restriction.
		It can be ON or OFF.
Service bindings	Service can be bound to only one virtual server.	Service can be bound to one or more virtual servers.
Internet Protocol (IP) versions	IPv4 and IPv6	IPV4
Redundancy support	Clustering and high availability	High availability

Features Affected by Connection Failover

The following table lists the features affected if connection failover is configured.

Table 2. How Connection Failover Affects NetScaler Features

Feature	Impact of Connection Failover
SYN protection	For any connection, if a failover occurs after the NetScaler issues SYN-ACK but before it receives the final ACK, the connection is not supported by connection failover. The client must reissue the request to establish the connection.
Surge protection	If the failover occurs before a connection with the server is established, the new primary NetScaler tries to establish the connection with the server. It also retransmits all the packets held in the course of surge protection.
Access down	If enabled, the access-down functionality takes precedence over connection failover.
Application Firewall™	The Application Firewall feature is not supported.
INC	Independent network configuration is not supported in the high availability (HA) mode.
TCP buffering	TCP buffering is not compatible with connection mirroring.
Close on	After failover, the NATPCBs may not be closed on response.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature	Impact of Connection Failover
IPv6 virtual servers	Not yet supported.

To configure connection failover by using the configuration utility, navigate to Traffic Management > Load Balancing > Virtual Servers. Open the virtual server, and in Advanced Settings click Protection, and select Connection Failover as Stateful.

To configure connection failover by using the command line interface, enter the following command:

```
set lb vserver <vServerName> -connFailover <Value>
```

Example

```
set lb vserver Vserver-LB-1 -connFailover stateful
```

When connection failover is disabled on a virtual server, the resources allocated to the virtual server are freed.

To disable connection failover by using the configuration utility, navigate to Traffic Management > Load Balancing > Virtual Servers. Open the virtual server, under Protection, select Connection Failover as Disabled.

To disable connection failover by using the command line interface, enter the following command:

```
set lb vserver <vServerName> -connFailover <Value>
```

Example

```
set lb vserver Vserver-LB-1 -connFailover disable
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Flushing the Surge Queue

Dec 04, 2013

When a physical server receives a surge of requests, it becomes slow to respond to the clients that are currently connected to it, which leaves users dissatisfied and disgruntled. Often, the overload also causes clients to receive error pages. To avoid such overloads, the NetScaler appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.

The appliance does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the appliance looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the appliance establishes a new connection with the server, and establishes a virtual link between client and the server. However, if the appliance cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the appliance cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The appliance maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which the appliance cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you may want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed; the other requests in the surge queue of the load balancing virtual server are not flushed.

Note: You cannot flush the surge queues of cache redirection, authentication, VPN or GSLB virtual servers or GSLB services.
Note: Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

To flush a surge queue by using the command line interface

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while executing the command, surge queue of the specified entity will be flushed. If more than one entity has the same name, the appliance flushes surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while executing the command, the appliance flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member (<serverName> and <port>) without specifying the name of the service group (<name>) and you cannot specify <port> without a <serverName>. Specify the <serverName> and <port> if you want to flush the surge queue for a specific service group member.
- If you execute the command without specifying any names, the appliance flushes the surge queues of all the entities present on the appliance.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the command prompt, type:

flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Examples

1.

flush ns surgeQ –name SVC1ANZGB –serverName 10.10.10.1 80

The above command flushes the surge queue of the service or virtual server that is named SVC1ANZGB and has IP address as 10.10.10

2.

flush ns surgeQ

The above command flushes all the surge queues on the appliance.

To flush a surge queue by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, select a virtual server and, in the Action list, select Flush Surge Queue.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing a Load Balancing Setup

Jan 01, 2018

An existing Load Balancing setup does not require a great deal of work to maintain as long as it is unchanged, but most do not remain unchanged for long. Increasing load requires new load-balanced servers and eventually new NetScaler appliances, which must be configured and added to the existing setup. Old servers wear out and need to be replaced, requiring removal of some servers and addition of others. Upgrades to your networking equipment or changes to topology may also require modifications to your load balancing setup. Therefore, you will need to perform operations on server objects, services, and virtual servers. The Visualizer can display your configuration graphically, and you can perform operations on the entities in the display. You can also take advantage of a number of other features that facilitate management of the traffic through your load balancing setup.

This section includes the following details:

- Managing Server Objects
- Managing Services
- Managing a Load Balancing Virtual Server
- The Load Balancing Visualizer

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Server Objects

Nov 12, 2013

During basic load balancing setup, when you create a service, a server object with the IP address of the service is created, if one does not already exist. If you prefer for your service objects named with domain names rather than IP addresses, you might also have created one or more server objects manually. You can enable, disable, or remove any server object.

When you enable or disable a server object, you enable or disable all services associated with the server object. When you refresh the NetScaler appliance after disabling a server object, the state of its service appears as OUT OF SERVICE. If you specify a wait time when disabling a server object, the server object continues to handle established connections for the specified amount of time, but rejects new connections. If you remove a server object, the service to which it is bound is also deleted.

To enable a server by using the command line interface

At the command prompt, type:

enable server <name>

Example

```
enable server 10.102.29.5
```

To enable or disable a server object by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Servers.
2. Select the server and, in the Action list, select Enable or Disable.

To disable a server object by using the command line interface

At the command prompt, type:

disable server <name> <delay>

Example

```
disable server 10.102.29.5 30
```

To remove a server object by using the command line interface

At the command prompt, type:

rm server <name>

Example

```
rm server 10.102.29.5
```

To remove a server object by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Servers.
2. Select a server, and click Remove.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Services

Nov 12, 2013

Services are enabled by default when you create them. You can disable or enable each service individually. When disabling a service, you normally specify a wait time during which the service continues to handle established connections, but rejects new ones, before shutting down. If you do not specify a wait time, the service shuts down immediately. During the wait time, the service's state is OUT OF SERVICE.

You can remove a service when it is no longer used. When you remove a service, it is unbound from its virtual server and deleted from the NetScaler configuration.

To enable or disable a service by using the command line interface

At the command prompt, type:

- enable service <name>
- disable service <name> <DelayInSeconds>

Examples

```
enable service Service-HTTP-1
disable service Service-HTTP-1 30
```

To enable or disable a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open a service and, in the Action list, select Enable or Disable.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing a Load Balancing Virtual Server

Feb 17, 2014

Virtual servers are enabled by default when you create them. You can disable and enable virtual servers manually. If you disable a virtual server, the virtual service's state appears as OUT OF SERVICE. When this happens, the virtual server terminates all connections, either immediately or after allowing existing connections to complete, depending on the setting of the downStateFLush parameter. If downStateFLush is ENABLED (default), all the connections are flushed. If DISABLED, the virtual server continues to serve requests on existing connections.

You remove a virtual server only when you no longer require the virtual server. Before you remove it, you must unbind all services from it.

To enable or disable a virtual server by using the command line interface

At the command prompt, type:

- enable lb vserver <name>
- disable lb vserver
  SYNOPSIS
  disable lb vserver <name>

Examples

```
enable lb vserver Vserver-LB-1
disable lb vserver Vserver-LB-1
```

To enable or disable a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Select a virtual server, and in the Action list, select Enable or Disable.

To unbind a service from a virtual server by using the command line interface

At the command prompt, type:

```
unbind lb vserver <name> <serviceName>
```

Example

```
unbind lb vserver Vserver-LB-1 Service-HTTP-1
```

To unbind a service from a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server, and click in the Services section.
3. Select a service and click Unbind.

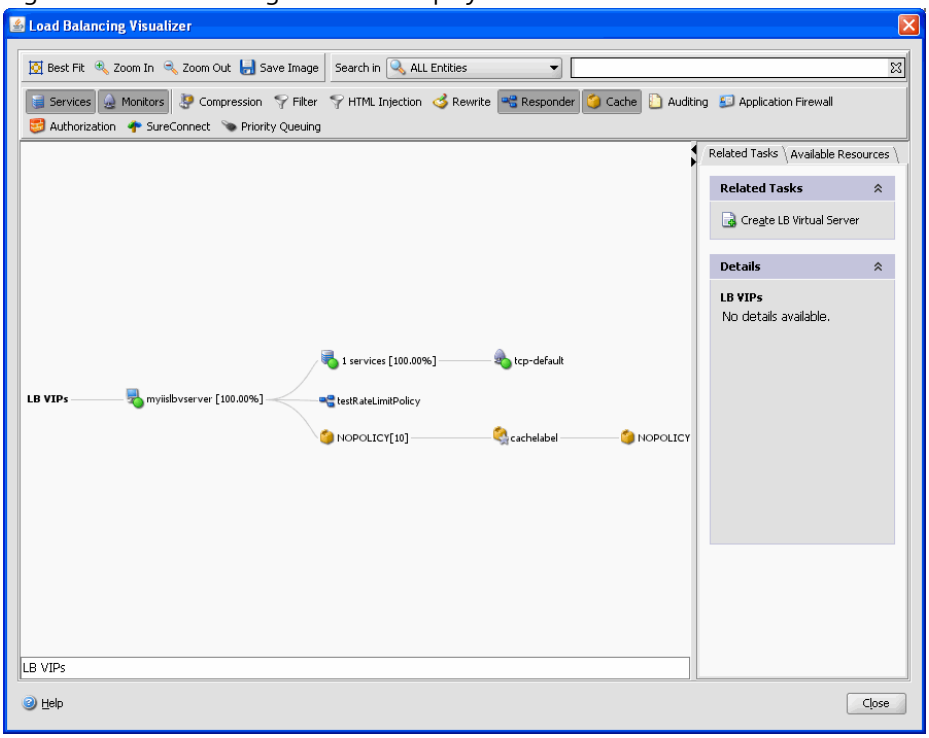
© 1999-2017 Citrix Systems, Inc. All rights reserved.

The Load Balancing Visualizer

Sep 03, 2013

The Load Balancing Visualizer is a tool that you can use to view and modify the load balancing configuration in graphical format. Following is an example of the Visualizer display

Figure 1. Load Balancing Visualizer Display



You can use the visualizer to view the following:

- The services and service groups that are bound to a virtual server.
- The monitors that are bound to each service.
- The policies that are bound to the virtual server.
- The policy labels, if configured.
- Configuration details of any displayed element.
- Load balancing virtual server statistics.
- Statistical information such as the number of requests received per second by the virtual server and the number of hits per second for rewrite, responder, and cache policies.
- A comparative list of all the parameters whose values either differ or are not defined across service containers.

You can also use the Visualizer to add and bind new objects, modify existing ones, and enable or disable objects. Most configuration elements displayed in the Visualizer appear under the same names as in other parts of the configuration utility. However, unlike the rest of the configuration utility, the Visualizer groups services that have the same configuration details and monitor bindings into an entity called a service container.

A service container is set of similar services and service groups that are bound to a single load balancing virtual server. Next to the service container is a number that shows the number of services in the group. The services in the container have the same properties, with the exception of the name, IP address, and port, and their monitor bindings should have the same weight and binding state. When you bind a new service to a virtual server, it is placed into an existing container if its configuration and monitor bindings match those of other services; otherwise, it is placed in its own container.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The service container display can help you troubleshoot your configuration if something is not functioning as you expect. More than one container for a particular virtual server is an indication that something is wrong with the configuration of that virtual server and its services. To correct the problem, you must first identify the container that has the desired configuration. You can do so by using the Service Attributes Diff feature, described below. After you identify the container, you right-click the container and click Apply Configuration.

The following procedures provide only basic steps for using the Visualizer. Because the Visualizer duplicates functionality in other areas of the Load Balancing feature, other methods of viewing or configuring all of the settings that can be configured in the Visualizer are provided throughout the Load Balancing documentation.

Note: The Visualizer requires a graphic interface, so it is available only through the configuration utility.

To view load balancing virtual server properties by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, you can adjust the viewable area as follows:
   - Click the Zoom In and Zoom Out icons to increase or decrease the size of the viewed objects. You can click and drag the viewable area if an item that you want to see disappears from view after zooming in.
   - Click the Best Fit icon to optimize the viewing area.
   - Click the Save Image icon to save the graph as an image file.
   - Click the image, hold down the mouse button, and drag the image to pan the view.
   - In the Search in text field, begin typing the name of the item you are looking for. The item's location is then highlighted. To restrict the search, click the drop-down menu and select the type of element that you want to search for

To view configuration details for services, service groups, and monitors by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, to view configuration details for entities that are bound to this virtual server, you can do the following:
   - To view a summary of bound services, position the cursor over the virtual server icon.
   - To view services in a service container, click the icon for a service group, click the Related Tasks tab, click Show Member Services, and then click the service group name. To view additional details about the services click Open.
   - To view common properties of services in a service group, click the icon for the service group, click the Related Tasks tab, and view the Details section of the tab.
   - To view a comparative list of the parameters whose values either differ or are not defined across service containers, click the icon for a container, click the Related Tasks tab, and then click Service Attributes Diff. To view monitor binding details for the services in a container, in the Service Attributes Diff dialog box, in the Group column for the container, click Details.
   - To view the details for a monitor, position the cursor over the icon or click the icon for the monitor. For additional details, click the icon, click the Related Tasks tab, and then click View Monitor.
   - To view binding details of a monitor, click the connecting line between the monitor and its related service.

To view configuration details for policies and policy labels by using the Visualizer in the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Load Balancing Visualizer dialog box, to view configuration details for entities that are bound to this virtual server, you can do the following:
   - To view policies that are bound to this virtual server, select one or more policy icons in the tool bar at the top of the dialog box. For example, you can select Compression, Filter, Rewrite, and Responder. If policy labels are configured, they appear in the main view area.
   - For bound policies that appear in the view pane of the Visualizer, to view a policy's expression and actions, position the cursor over the policy icon. To view binding details, position the cursor over the line that connects the policy to the virtual server. To view these details, click the policy. The details of the policy appear in the details pane.

To view statistical information by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, to view statistical information, you can do the following:
   - To view detailed statistics for the load balancing virtual server, click the icon for the virtual server, click the Related Tasks tab, and then click Statistics.
   - To view the number of requests received per second at a given point in time by the load balancing virtual server and the number of hits per second at a given point in time for rewrite, responder, and cache policies, click Show Stats. The statistical information is displayed on the respective nodes in the Visualizer. This information is not updated in real time and has to be refreshed manually. To refresh this information, click Refresh Stats.

     Note: The Show Stats option is available only on NetScaler nCore builds.

To save configuration properties for any entity by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.
3. To copy configuration details for an element to a document or spreadsheet, click the icon for that element, click Related Tasks.
4. In the Related Tasks tab, click Copy Properties and then paste the information into a document.

To bind a resource to a load balancing configuration by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure bindings, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, click the Available Resources tab, select a resource type in the drop-down menu, and do one or more of the following:
   - To bind a new monitor to a service, select Monitors, click a particular monitor, and then drag it to the service container icon. Use CONTROL + click to select multiple monitors and drag them to the service.
   - To bind a service or service group, select Services or Service Groups, respectively, click a particular service or service group, and then drag it to the virtual server icon. To bind multiple services or service groups at one time, press CONTROL + click to select multiple services and drag them over the virtual server.
   - To bind a policy, select one of the policy groups, click a particular policy, and then drag it to a virtual server. To bind multiple policies (classic policies only) at one time, press CONTROL + policies and drag them over the virtual server. For details on classic and advanced policies, see Policy Configuration and Reference.

To unbind a resource by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server from which you want to unbind a service, policy, or monitor, and then click

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Visualizer.

3. In the Load Balancing Visualizer dialog box, on the Visualizer image, click the connecting line between the resources that you want to unbind, and then click Unbind. For example, to unbind a monitor, you would click the link between the monitor and its bound service and click Unbind.
4. In the Unbind dialog box, click Yes.

To modify a resource in a load balancing configuration by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to configure, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, on the Visualizer image, double-click the resource that you want to modify.
   Note: Alternatively, on the Available Resources tab, select the resource type from the drop-down menu, select the particular resource that you want to configure and then click Open.
4. In the modify dialog box, enter new settings for the resource.

To add, remove, or disable a resource in a load balancing configuration by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to configure, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, right-click the icon for the resource that you want to add, remove, or disable, and then select the corresponding option from the menu.
   Note: Alternatively, on the Available Resources tab, click the resource type from the drop-down menu, and then click Add to add an entity, or select the particular resource that you want to configure and then click Open.
   Note: These options are not available for service groups or policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Client Traffic

Dec 27, 2017

Managing client connections properly helps to ensure that your applications remain available to users even when your NetScaler appliance is experiencing high loads. A number of load balancing features and other features available on the appliance can be integrated into a load balancing setup to process load more efficiently, divert it when necessary, and prioritize the tasks that the appliance must perform:

- **Sessionless load balancing.** You can configure sessionless load balancing virtual servers and perform load balancing without creating sessions in configurations that use DSR or intrusion detection systems (IDS).
- **Integrated caching.** You can redirect HTTP requests to a cache.
- **Priority queuing.** You can direct requests based on priority, by integrating your configuration with the Priority Queuing feature.
- **SureConnect.** You can use load balancing with the SureConnect feature to redirect important requests to a custom Web page, insulating them from delays due to network congestion.
- **Delayed cleanup.** You can configure delayed cleanup of virtual server connections to prevent the cleanup process from using CPU cycles during periods when the NetScaler appliance is experiencing high loads.
- **Rewrite.** You can use the Rewrite feature to modify port and protocol when performing HTTP redirection, or insert the virtual server IP address and port into a custom Request header.
- **RTSP NAT.**
- **Rate-based monitoring.** You can enable rate-based monitoring to divert excess traffic.
- **Layer 2 Parameters.** You can configure a virtual server to use the L2 parameters to identify a connection.
- **ICMP Response**. You can configure the NetScaler to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP VSERVER RESPONSE.
  The following settings can be made on a virtual server:
  - When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, NetScaler always responds.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds even if one virtual server is UP.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds even if one virtual server set to ACTIVE is UP.

To manage client traffic, see the following sections:

- Configuring Sessionless Load Balancing Virtual Servers
- Redirecting HTTP Requests to a Cache
- Directing Requests According to Priority
- Directing Requests to a Custom Web Page
- Enabling Cleanup of Virtual Server Connections
- Graceful Shut down of Services
- Rewriting Ports and Protocols for HTTP Redirection
- Inserting the IP Address and Port of a Virtual Server in the Request Header
- Using a Specified Source IP for Backend Communication
- Setting a Timeout Value for Idle Client Connections
- Managing RTSP Connections
- Managing Client Traffic on the Basis of Traffic Rate
- Identifying a connection with Layer 2 Parameters

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Configuring the Prefer Direct Route Option

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Sessionless Load Balancing Virtual Servers

May 11, 2017

When the NetScaler appliance performs load balancing, it creates and maintains sessions between clients and servers. The maintenance of session information places a significant load on the NetScaler resources, and sessions might not be needed in scenarios such as a direct server return (DSR) setup and the load balancing of intrusion detection systems (IDS). To avoid creating sessions when they are not necessary, you can configure a virtual server on the appliance for sessionless load balancing. In sessionless load balancing, the appliance carries out load balancing on a per-packet basis.

Sessionless load balancing can operate in MAC-based forwarding mode or IP-based forwarding mode.

For MAC-based forwarding, the IP address of the sessionless virtual server must be specified on all the physical servers to which the traffic is forwarded.

For IP-based forwarding in sessionless load balancing, the IP address and port of the virtual server need not be specified on the physical servers, because this information is included in the forwarded packets. When forwarding a packet from the client to the physical server, the appliance leaves client details such as IP address and port unchanged and adds the IP address and port of the destination.

Supported Setup

NetScaler sessionless load balancing supports the following service types and load balancing methods:

Service Types

- ANY for MAC-based redirection
- ANY, DNS, and UDP for IP-based redirection

Load Balancing Methods

- Round Robin
- Least Bandwidth
- LRTM (Least response time method)
- Source IP Hash
- Destination IP Hash
- Source IP Destination IP Hash
- Source IP Source Port Hash
- Custom Load

Limitations

Sessionless load balancing has the following limitations:

- The NetScaler must be deployed in two-arm mode.
- A service must be bound to only one virtual server.
- Sessionless load balancing is not supported for service groups.
- Sessionless load balancing is not supported for domain based services (DBS services).
- Sessionless load balancing in the IP mode is not supported for a virtual server that is configured as a backup to a primary virtual server.
- You cannot enable spillover mode.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- For all the services bound to a sessionless load balancing virtual server, the Use Source IP (USIP) option must be enabled.
- For a wildcard virtual server or service, the destination IP address will not be changed.

Note: While configuring a virtual server for sessionless load balancing, explicitly specify a supported load balancing method. The default method, Least Connection, cannot be used for sessionless load balancing.
Note: To configure sessionless load balancing in MAC-based redirection mode on a virtual server, the MAC-based forwarding option must be enabled on the NetScaler.

To add a sessionless virtual server by using the command line interface

At the command prompt, type the following commands to add a sessionless virtual server and verify the configuration:

- add lb vserver <name>@ <serviceType> <IPAddress>@ <port> -m <redirectionMode> -sessionless <(ENABLED | DISABLED)> -lbMethod <load_balancing_method>
- show lb vserver <name>

Example

```
add lb vserver sesslessv1 any 11.11.12.123 54 -sessionless ENABLED -lbMethod roundrobin -m ip
    Done
show lb vserver sesslessv1
    sesslessv1 (11.11.12.123:54) - ANY Type: ADDRESS
    State: DOWN
    ...
    Effective State: DOWN
    Client Idle Timeout: 120 sec
    Down state flush: ENABLED
    ...
    Persistence: NONE
    Sessionless LB: ENABLED
    Connection Failover: DISABLED
    L2Conn: OFF
    1) Policy : cmp_text Priority:8680 Inherited
    2) Policy : cmp_nocmp_ie60 Priority:8690 Inherited
```

To configure sessionless load balancing on an existing virtual server

At the command prompt, type:

set lb vserver <name>@ -m <redirectionMode> -sessionless <(ENABLED | DISABLED)> -lbMethod <load_balancing_method>
Example

```
set lb vserver sesslessv1 -m mac -sessionless ENABLED -lbmethod lrtm
 Done
```

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure a sessionless virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open the virtual server, and in Advanced Settings, click Traffic Settings, and then select Sessionless Load Balancing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Redirecting HTTP Requests to a Cache

Nov 11, 2013

The NetScaler cache redirection feature redirects HTTP requests to a cache. You can significantly reduce the impact of responding to HTTP requests and improve your Web site performance through proper implementation of the cache redirection feature.

A cache stores frequently requested HTTP content. When you configure cache redirection on a virtual server, the NetScaler appliance sends cacheable HTTP requests to the cache, and non-cacheable HTTP requests to the origin Web server.

To configure cache redirection on a virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -cacheable <Value>
```

Example

```
set lb vserver Vserver-LB-1 -cacheable yes
```

To configure cache redirection on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select Cacheable.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Directing Requests According to Priority

Nov 11, 2013

The NetScaler appliance supports prioritization of client requests with its priority queuing feature. This feature allows you to designate certain requests, such as those from important clients, as priority requests and sends them to the “front of the line,” so that the appliance responds to them first. This allows you to provide uninterrupted service to those clients through demand surges or DDoS attacks on your web site.

To configure priority queuing on a virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -pq <Value>
```

Example

```
set lb vserver Vserver-LB-1 -pq yes
```

To configure priority queuing on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select Priority Queuing.

Note: You must configure priority queuing globally for it to function correctly.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Directing Requests to a Custom Web Page

Nov 11, 2013

The NetScaler appliance provides the SureConnect option to ensure that web applications respond despite delays caused by limited server capacity or processing speed. SureConnect does this by displaying an alternative web page of your choice when the server that hosts the primary web page is either unavailable or responding slowly.

To configure SureConnect on a virtual server, you must first configure the alternative content. For information about configuring a SureConnect website, see SureConnect. After you configure the website, enable SureConnect on the load balancing virtual server to put your SureConnect custom web page in use.

Note: For SureConnect to function correctly, you must configure it globally.

To enable SureConnect on a virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -sc <Value>
```

Example

```
set lb vserver Vserver-LB-1 -sc yes
```

To enable SureConnect on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select SureConnect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Cleanup of Virtual Server Connections

Feb 08, 2016

Under certain conditions, you can configure the downStateFlush setting to immediately terminate existing connections when a service or a virtual server is marked DOWN. Terminating existing connections frees resources, and in certain cases speeds recovery of overloaded load balancing setups.

The state of a virtual server depends on the states of the services bound to it. The state of each service depends on the responses of the load balanced servers to probes and health checks sent by the monitors that are bound to that service. Sometimes the load balanced servers do not respond. If a server is slow or busy, monitoring probes can time out. If repeated monitoring probes are not answered within the configured timeout period, the service is marked DOWN.

A virtual server is marked DOWN only when all services bound to it are marked DOWN. When a virtual server goes DOWN, it terminates all connections, either immediately or after allowing existing connections to complete.

You must not enable the downStateFlush setting on those application servers that must complete their transactions. You can enable this setting on Web servers whose connections can safely be terminated when they marked DOWN.

The following table summarizes the effect of this setting on an example configuration consisting of a virtual server, Vserver-LB-1, with one service bound to it, Service-TCP-1. In the table, E and D denote the state of the downStateFlush setting: E means Enabled, and D means Disabled.

Vserver-LB-1	Service-TCP-1	State of connections
E	E	Both client and server connections are terminated.
E	D	For some service types, such as TCP, for which the NetScaler appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only client connections are terminated.
D	E	For some service types, such as TCP, for which the NetScaler appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only server connections are terminated.
D	D	Neither client nor server connections are terminated.

If you want to disable a service only when all the established connections are closed by the server or the client, you can use the graceful shutdown option. For information about the graceful shutdown of a service, see Graceful Shutdown of

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Services.

To configure the down state flush setting on a virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -downStateFlush <Value>
```

Example

```
set lb vserver Vserver-LB-1 -downStateFlush enabled
```

To configure the down state flush setting on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select Down State Flush.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rewriting Ports and Protocols for HTTP Redirection

Nov 11, 2013

Virtual servers and the services that are bound to them may use different ports. When a service responds to an HTTP connection with a redirect, you might need to configure the NetScaler appliance to modify the port and the protocol to make sure that the redirection goes through successfully. You do this by enabling and configuring the redirectPortRewrite setting.

This setting affects only HTTP and HTTPS traffic. If this setting is enabled on a virtual server, the virtual server rewrites the port on redirects, replacing the port used by the service with the port used by the virtual server.

If the virtual server or service is of type SSL, you must enable SSL redirect on the virtual server or service. If both the virtual server and service are of type SSL, enable SSL redirect on the virtual server.

The redirectPortRewrite setting can be used in the following scenarios:

- The virtual server is of type HTTP and the services are of type SSL.
- The virtual server is of type SSL and the services are of type HTTP.
- The virtual server is of type HTTP and the services are of type HTTP.
- The virtual server is of type SSL and the services are of type SSL.

Scenario 1: The virtual server is of type HTTP and services are of type SSL. SSL redirect, and optionally port rewrite, is enabled on the service. If port rewrite is enabled, the port of HTTPS URLs is rewritten. HTTP URLs from the server are sent as is to the client.

Redirect URL from the Server	**Redirect URL sent to the Client**
Only SSL redirect is enabled. The virtual server can be configured on any port.	
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/
SSL redirect and port rewrite are enabled. The virtual server is configured on port 80.	
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Redirect URL from the Server	Redirect URL sent to the Client
https://domain.com:444/	https://domain.com/
SSL redirect and port rewrite are enabled. Virtual server is configured on port 8080.	
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	http://domain.com:8080/
https://domain.com:444/	http://domain.com:8080/

Scenario 2: The virtual server is of type SSL and services are of type HTTP. If port rewrite is enabled, only the port of HTTP URLs is rewritten. HTTPS URLs from the server are sent as is to the client.

Redirect URL from the Server	Redirect URL sent to the Client
SSL redirect is enabled on the virtual server. The virtual server can be configured on any port.	
http://domain.com/	https://domain.com/
http://domain.com:8080/	https://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/
SSL redirect and port rewrite are enabled on the virtual server. The virtual server is configured on port 443.	
http://domain.com/	https://domain.com/
http://domain.com:8080/	https://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/
SSL redirect and port rewrite are enabled. The virtual server is configured on port 444.	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Redirect URL from the Server	Redirect URL sent to the Client
http://domain.com/	https://domain.com:444/
http://domain.com:8080/	https://domain.com:444/
https://domain.com/	https://domain.com/
https://domain.com:445/	https://domain.com:445/

Scenario 3: The virtual server and service are of type HTTP. Port rewrite must be enabled on the virtual server. Only the port of HTTP URLs is rewritten. HTTPS URLs from the server are sent as is to the client.

Redirect URL from the Server	Redirect URL sent to the Client
The virtual server is configured on port 80.	
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/
The virtual server is configured on port 8080.	
http://domain.com/	http://domain.com:8080/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:445/	https://domain.com:445/

Scenario 4: The virtual server and service are of type SSL. If port rewrite is enabled, only the port of HTTPS URLs is rewritten. HTTP URLs from the server are sent as is to the client.

Redirect URL from the Server	Redirect URL sent to the Client
SSL redirect is enabled on the virtual server. The virtual server can be configured on any port.	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Redirect URL from the Server http://domain.com/	Redirect URL sent to the Client http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/
SSL redirect and port rewrite are enabled on the virtual server. The virtual server is configured on port 443.	
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com/
SSL redirect and port rewrite are enabled on the virtual server. The virtual server is configured on port 444.	
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com:444/
https://domain.com:445/	https://domain.com:444/

To configure HTTP redirection on a virtual server by using the command line interface

At the command prompt, type:

set lb vserver <name> -redirectPortRewrite (ENABLED | DISABLED)

Example

```
set lb vserver Vserver-LB-1 -redirectPortRewrite enabled
```

To configure HTTP redirection on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open the virtual server, and in the Advanced Settings pane, click Traffic Settings, and then select Rewrite.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure SSL Redirect on an SSL virtual server or service by using the command line interface

At the command prompt, type:

- set ssl vserver <vServerName> - sslRedirect (ENABLED | DISABLED)
- set ssl service <serviceName> - sslRedirect (ENABLED | DISABLED)

Example

```
set ssl vserver Vserver-SSL-1 -sslRedirect enabled
set ssl service service-SSL-1 -sslRedirect enabled
```

To configure SSL redirection and SSL port rewrite on an SSL virtual server or service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click SSL Parameters, and select SSL Redirect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Inserting the IP Address and Port of a Virtual Server in the Request Header

Nov 11, 2013

If you have multiple virtual servers that communicate with different applications on the same service, you must configure the NetScaler appliance to add the IP address and port number of the appropriate virtual server to the HTTP requests that are sent to that service. This setting allows applications running on the service to identify the virtual server that sent the request.

If the primary virtual server is down and the backup virtual server is up, the configuration settings of the backup virtual server are added to the client requests. If you want the same header tag to be added, regardless of whether the requests are from the primary virtual server or backup virtual server, then you must configure the required header tag on both virtual servers.

Note: This option is not supported for wild card virtual servers or dummy virtual servers.

To insert the IP address and port of the virtual server in the client requests by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -insertVserverIPPort <insertVserverIPPort> [<vipHeader>]
```

Example

```
set lb vserver Vserver-LB-1 -insertVserverIPPort VipAddr
```

To insert the IP address and port of the virtual server in the client requests by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open the virtual server, and in the Advanced Settings pane, click Traffic Settings, and then select Virtual Server IP Port Insertion and specify a virtual server IP port header.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using a Specified Source IP for Backend Communication

Aug 02, 2017

For communication with the physical servers or other peer devices, the NetScaler appliance uses an IP address owned by it as the source IP address. NetScaler maintains a pool of its IP addresses, and dynamically selects an IP address while connecting with a server. Depending on the subnet in which the physical server is placed, NetScaler decides which IP address to use. This address pool is used for sending traffic as well as monitor probes.

In many situations, you may want the NetScaler to use a specific IP address or any IP address from a specific set of IP addresses for backend communications. The following are a few examples:

- A server can distinguish monitor probes from traffic if the source IP address used for monitor probes belongs to a specific set.
- To improve server security, a server may be configured to respond to requests from a specific set of IP addresses or, sometimes, from a single specific IP address. In such a case, the NetScaler can use only the IP addresses accepted by the server as the source IP address.
- The NetScaler can manage its internal connections efficiently if it can distribute its IP addresses into IP sets and use an address from a set only for connecting to a specific service.

To configure the NetScaler to use a specified source IP address, create net profiles (network profiles) and configure the NetScaler entities to use the profile. A net profile can be bound to load balancing or content switching virtual servers, services, service groups, or monitors. A net profile has NetScaler owned IP addresses (SNIPs and VIPs) that can be used as the source IP address. It can be a single IP address or a set of IP addresses, referred to as an IP set. If a net profile has an IP set, NetScaler dynamically selects an IP address from the IP set at the time of connection. If a profile has a single IP address, the same IP address is used as the source IP.

If a net profile is bound to a load balancing or content switching virtual server, the profile will be used for sending traffic to all the services bound to it. If a net profile is bound to a service group, NetScaler uses the profile for all the members of the service group. If a net profile is bound to a monitor, NetScaler uses the profile for all the probes sent from the monitor.

Note: When a NetScaler appliance uses a VIP address to communicate with a server, it uses session entries to identify whether the traffic destined to the VIP address is a response from a server or a request from a client.

Usage of a net profile for sending traffic:

If the Use Source IP Address (USIP) option is enabled, NetScaler uses the IP address of the client and ignores all the net profiles. If the USIP option is not enabled, NetScaler selects the source IP in the following manner:

- If there is no net profile on the virtual server or the service/service group, NetScaler uses the default method.
- If there is a net profile only on the service/service group, NetScaler uses that net profile.
- If there is a net profile only on the virtual server, NetScaler uses the net profile.
- If there is a net profile both on the virtual server and service/service group, NetScaler uses the net profile bound to the service/service group.

Usage of a net profile for sending monitor probes:

For monitor probes, NetScaler selects the source IP in the following manner:

- If there is a net profile bound to the monitor, NetScaler uses the net profile of the monitor. It ignores the net profiles

© 1999-2017 Citrix Systems, Inc. All rights reserved.

bound to the virtual server or service/service group.

- If there is no net profile bound to the monitor,
  - If there is a net profile on the service/service group, NetScaler uses the net profile of the service/service group.
  - If there is no net profile even on the service/service group, NetScaler uses the default method of selecting a source IP.

Note: If there is no net profile bound to a service, NetScaler looks for a net profile on the service group if the service is bound to a service group.
To use a specified source IP address for communication, go through the following steps:

1. Create IP sets from the pool of SNIPs and VIPs owned by the NetScaler. An IP set can consist of both SNIP and VIP addresses. For instructions, see Creating IP Sets.
2. Create net profiles. For instructions, see Creating a Net Profile.
3. Bind the net profiles to NetScaler entities. For instructions, see Binding a Net Profile to a NetScaler Entity.

Note: A net profile can have only the IP addresses specified as SNIP and VIP on the NetScaler.

Managing Net Profiles

A net profile (or network profile) contains an IP address or an IP set. During communication with physical servers or peers, the NetScaler appliance uses the addresses specified in the profile as the source IP address.

- For instructions on creating a network profile, see Creating a Network Profile.
- For instructions on binding a network profile to a NetScaler entity, see Binding a Network Profile.

Creating an IP Set

An IP set is a set of IP addresses, which are configured on the NetScaler appliance as Subnet IP addresses (SNIPs) or Virtual IP addresses (VIPs). An IP set is identified with a meaningful name that helps in identifying the usage of the IP addresses contained in it. To create an IP set, add an IP set and bind NetScaler owned IP addresses to it. SNIP addresses and VIP addresses can be present in the same IP set.

To create an IP set by using the command line interface

At the command prompt, type the following commands:

- add ipset <name>
- bind ipset <name> <IPAddress>
  or
- bind ipset <name> <IPAddress>
- show ipset [<name>]
  The above command shows the names of all the IP sets on the NetScaler if you do not pass any name. It shows the IP addresses bound to the specified IP set if you pass a name.

Examples

1.

```
> add ipset skpnwipset
 Done
> bind ipset skpnwipset 21.21.20.1
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

2.

```
 > add ipset testnwipset
 Done
> bind ipset testnwipset 21.21.21.[21-25]
 IPAddress "21.21.21.21" bound
 IPAddress "21.21.21.22" bound
 IPAddress "21.21.21.23" bound
 IPAddress "21.21.21.24" bound
 IPAddress "21.21.21.25" bound
 Done
```

3.

```
 > bind ipset skpipset 11.11.11.101
 ERROR: Invalid IP address
[This IP address could not be added because this is not an IP address owned by the NetScaler]
 > add ns ip 11.11.11.101 255.255.255.0 -type SNIP
 ip "11.11.11.101" added
 Done
 > bind ipset skpipset 11.11.11.101
 IPAddress "11.11.11.101" bound
 Done
```

4.

```
> sh ipset
1) Name: ipset-1
2) Name: ipset-2
3) Name: ipset-3
4) Name: skpnewipset
 Done
```

5.

```
> sh ipset skpnewipset
IP:21.21.21.21
IP:21.21.21.22
IP:21.21.21.23
IP:21.21.21.24
IP:21.21.21.25
 Done
```

To create an IP set by using the configuration utility

Navigate to System > Network > IP Sets, and create an IP set.

Creating a Net Profile

A net profile (network profile) consists of one or more SNIP or VIP addresses of the NetScaler.

To create a net profile by using the command line interface

At the command prompt, type:

add netprofile <name> [-srcIp <srcIpVal>] If the srcIpVal is not provided in this command, it can be provided later by using

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

the set netprofile command.
Examples

```
> add netprofile skpnetprofile1 -srclp 21.21.20.1
Done

> add netprofile baksnp -srclp bakipset
Done

> set netprofile yahnp -srclp 12.12.23.1
Done

> set netprofile citkbnp -srclp citkbipset
Done
```

Binding a Net Profile to a NetScaler Entity

A net profile can be bound to a load balancing virtual server, service, service group, or a monitor.
Note: You can bind a net profile at the time of creating a NetScaler entity or bind it to an already existing entity.

To bind a net profile to a server by using the command line interface

You can bind a net profile to load balancing virtual servers and content switching virtual servers. Specify the appropriate virtual server.

At the command prompt, type:

- set lb vserver <name> -netProfile <net_profile_name>
  or
- set cs vserver <name> -netProfile <net_profile_name>

Examples

```
set lb vserver skpnwvs1 -netProfile gntnp
Done
set cs vserver mmdcsv -netProfile mmdnp
Done
```

To bind a net profile to a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Profiles, and set a net profile.

To bind a net profile to a service by using the command line interface

At the command prompt, type:

set service <name> -netProfile <net_profile_name>
Example

```
set service brnssvc1 -netProfile brnsnp
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To bind a net profile to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, click Profiles, and set a net profile.

To bind a net profile to a service group by using the command line interface

At the command prompt, type:

set servicegroup <serviceGroupName> -netProfile <net_profile_name>
Example

```
set servicegroup ndhsvcgrp -netProfile ndhnp
 Done
```

To bind a net profile to a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups, and opena service group.
2. In Advanced Settings, click Profiles, and set a net profile.

To bind a net profile to a monitor by using the command line interface

At the command prompt, type:

set monitor <monitor_name> -netProfile <net_profile_name>
Example

```
set monitor brnsecvmon1 -netProfile brnsmonnp
 Done
```

To bind a net profile to a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Open a monitor, and set the net profile.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting a Time-out Value for Idle Client Connections

May 31, 2017

You can configure a virtual server to terminate any idle client connections after a configured time-out period (in seconds) elapses. When you configure this setting, the NetScaler appliance waits for the time you specify and, if the client is idle after that time, it closes the client connection.

To set a time-out value for idle client connections by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -cltTimeout <Value>
```

Example

```
set lb vserver Vserver-LB-1 -cltTimeout 100
```

To set a time-out value for idle client connections by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In **Advanced Settings,** click **Traffic Settings**, and set the client idle time-out value in seconds.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing RTSP Connections

Nov 11, 2013

The NetScaler appliance can use either of two topologies—NAT-on mode or NAT-off mode—to load balance RTSP servers. In NAT-on mode, Network Address Translation (NAT) is enabled and configured on the appliance. RTSP requests and responses both pass through the appliance. You must therefore configure the appliance to perform network address translation (NAT) to identify the data connection.

For more information about enabling and configuring NAT, see "IP Addressing."

In NAT-off mode, NAT is not enabled and configured. The appliance receives RTSP requests from the client and routes them to the service that it selects using the configured load balancing method. The load balanced RTSP servers send their responses directly to the client, bypassing the appliance. You must therefore configure the appliance to use Direct Server Return (DSR) mode, and assign publicly accessible FQDNs in DNS to your load balanced RTSP servers.

For more information about enabling and configuring DSR mode, see "Configuring Load Balancing in Direct Server Return Mode." For more information about configuring DNS, see "Domain Name System."

In either case, when you configure RTSP load balancing, you must also configure rtspNat to match the topology of your load balancing setup.

To configure RTSP NAT by using the command line interface

At the command prompt, type:

set lb vserver <name> –RTSPNAT <ValueOfRTSPNAT>

Example

set lb vserver vserver-LB-1 –RTSPNAT ON

To configure RTSP NAT by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server of type RTSP.
2. In Advanced Settings, click **Traffic Settings**, and select **RTSP Natting**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Client Traffic on the Basis of Traffic Rate

Sep 30, 2015

You can monitor the rate of traffic that flows through load balancing virtual servers and control the behavior of the NetScaler appliance based on the traffic rate. You can throttle the traffic flow if it is too high, cache information based on the traffic rate, and if the traffic rate is too high redirect excess traffic to a different load balancing virtual server. You can apply rate-based monitoring to HTTP and Domain Name System (DNS) requests.

For more information on rate-based policies, see Rate Limiting.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Identifying a connection with Layer 2 Parameters

Nov 12, 2013

Generally, to identify a connection, the NetScaler uses the 4-tuple of client IP address, client port, destination IP address, and destination port. When you enable the L2 Connection option, the Layer 2 parameters of the connection (channel number, MAC address, and VLAN ID) are used in addition to the normal 4-tuple.

Enabling the L2Conn parameter for a load balancing virtual server allows multiple TCP and non-TCP connections with the same 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) to co-exist on the NetScaler appliance. The appliance uses both the 4-tuple and the Layer 2 parameters to identify TCP and non-TCP connections.

You can enable the L2Conn option in the following scenarios:

- Multiple VLANs are configured on the NetScaler appliance, and a firewall is set up for each VLAN.
- You want the traffic originating from the servers in one VLAN and bound for a virtual server in another VLAN to pass through the firewalls configured for both VLANs.

Therefore, when an nCore NetScaler appliance on which the l2Conn parameter is set for one or more load balancing virtual servers is downgraded to a Classic build or to an nCore build that does not support the l2Conn parameter, the load balancing configurations that use the l2Conn parameter become ineffective.

To configure the L2 connection option by using the command line interface

At the command prompt, type:

add lb vserver <name> <serviceType> <IPAddress>@ <port> -l2Conn ON

Example

```
add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -l2Conn ON
```

To configure the L2 connection option by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Layer 2 Parameters.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Prefer Direct Route Option

Aug 29, 2013

On a wildcard load balancing virtual server if you explicitly configure a route to a destination, by default, the NetScaler appliance forwards traffic according to the configured route. If you want the NetScaler to not look up for the configured route, you can set the Prefer Direct Route option to NO.

If a device is directly connected to a NetScaler appliance, the NetScaler directly forwards traffic to the device. For example, if the destination of a packet is a firewall, the packet need not be routed through another firewall. However, in some cases, you may want the traffic to go through the firewall even if the device is directly connected to it. In such cases, you can set the Prefer Direct Route Option to NO.

Note: The preferDirectRoute setting is applicable to all the wildcard virtual servers on the NetScaler appliance.

To set the prefer direct route option by using the command line interface

At the command prompt, type:

```
set lb parameter -preferDirectRoute (YES | NO)
```

Example

```
set lb parameter -preferDirectRoute YES
```

To set the prefer direct route option by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing parameters.
2. Select Prefer Direct Route.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using a Source Port from a Specified Port Range for Backend Communication

Jan 04, 2016

By default, for configurations with USIP option disabled or with USIP and use proxy port options enabled, the NetScaler appliance communicates to the servers from a random source port (greater than 1024).

The NetScaler supports using a source port from a specified port range for communicating to the servers. One of the use case of this feature is for servers that are configured to identify received traffic belonging to a specific set on the basis of source port for logging and monitoring purposes. For example, identifying internal and external traffic for logging purpose.

Configuring the NetScaler appliance to use a source port from a port range for communicating to the servers consists of the following tasks:

- **Create a net profile and set the source port range parameter**. A source port range parameter specifies one or more port ranges. The NetScaler randomly selects one of the free ports from the specified port ranges and used it as the source port for each connection to servers.
- **Bind the net profile to load balancing virtual servers, services, or service groups**: A net profile with source port range setting can be bound to a virtual server, service, or a service group of a load balancing configuration. For a connection to a virtual server, the NetScaler randomly selects one of the free ports from the specified port ranges of a net profile and use this port as the source port for connecting to one of the bound server.

To specify a source port range or ranges by using the NetScaler command line

At the command prompt, type:

- **bind netProfile** <name> (-**srcPortRange** <int[-int]> ...)
- **show netprofile** <name>

To specify a source port range or ranges by using the configuration utility

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Set the **Source Port Range** parameter while adding or modifying NetProfiles.

Sample Configuration COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

In the following sample configuration, net profile PARTIAL-NAT-1 has partial NAT settings and is bound to load balancing virtual server LB

```
> add netprofile CUSTOM-SRCPORT-NP-1

Done

> bind netprofile CUSTOM-SRCPRT-NP-1 –srcportrange 2000-3000

Done

> bind netprofile CUSTOM-SRCPRT-NP-1 –srcportrange 5000-6000

Done

> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1

Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Advanced Load Balancing Settings

Dec 15, 2015

In addition to configuring virtual servers, you can configure advanced settings for services.

To configure advanced load balancing settings, see the following sections:

- Gradually Stepping Up the Load on a New Service with Virtual Server–Level Slow Start
- The No-Monitor Option for Services
- Protecting Applications on Protected Servers Against Traffic Surges
- Enabling Cleanup of Service Connections
- Directing Requests to a Custom Web Page
- Enabling Access to Services When Down
- Enabling TCP Buffering of Responses
- Enabling Compression
- Maintaining Client Connection for Multiple Client Requests
- Inserting the IP Address of the Client in the Request Header
- Using the Source IP Address of the Client When Connecting to the Server
- Configuring the Source Port for Server-Side Connections
- Setting a Limit on the Number of Client Connections
- Setting a Limit on Number of Requests Per Connection to the Server
- Setting a Threshold Value for the Monitors Bound to a Service
- Setting a Timeout Value for Idle Client Connections
- Setting a Timeout Value for Idle Server Connections
- Setting a Limit on the Bandwidth Usage by Clients
- Redirecting Client Requests to a Cache
- Retaining the VLAN Identifier for VLAN Transparency
- Configuring Automatic State Transition Based on Percentage Health of Bound Services

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Gradually Stepping Up the Load on a New Service with Virtual Server–Level Slow Start

Mar 16, 2012

You can configure the NetScaler appliance to gradually increase the load on a service (the number of requests that the service receives per second) immediately after the service is either added to a load balancing configuration or has a state change from DOWN to UP (hereafter, the term “new service” is used for both situations). You can either increase the load manually with load values and intervals of your choice (manual slow start) or configure the appliance to increase the load at a specified interval (automated slow start) until the service is receiving as many requests as the other services in the configuration. During the ramp-up period for the new service, the appliance uses the configured load balancing method.

This functionality is not available globally. It has to be configured for each virtual server. The functionality is available only for virtual servers that use one of the following load balancing methods:

- Round robin
- Least connection
- Least response time
- Least bandwidth
- Least packets
- LRTM (Least Response Time Method)
- Custom load

For this functionality, you need to set the following parameters:

- The new service request rate, which is the amount by which to increase the number or percentage of requests sent to a new service each time the rate is incremented. That is, you specify the size of the increment in terms of either the number of requests per second or the percentage of the load being borne, at the time, by the existing services. If this value is set to 0 (zero), slow start is not performed on new services.
  Note: In automated slow start mode, the final increment is smaller than the specified value if the specified value would place a heavier load on the new service than on the other services.
- The increment interval, in seconds. If this value is set to 0 (zero), the load is not incremented automatically. You have to increment it manually.

With automated slow start, a service is taken out of the slow start phase when one of the following conditions applies:

- The actual request rate is less than the new service request rate.
- The service does not receive traffic for three successive increment intervals.
- The request rate has been incremented 200 times.
- The percentage of traffic that the new service must receive is greater than or equal to 100.

With manual slow start, the service remains in the slow start phase until you take it out of that phase.

Manual Slow Start

If you want to manually increase the load on a new service, do not specify an increment interval for the load balancing virtual server. Specify only the new service request rate and the units. With no interval specified, the appliance does not increment the load periodically. It maintains the load on the new service at the value specified by the combination of the new service request rate and units until you manually modify either parameter. For example, if you set the new service request rate and unit parameters to 25 and “per second,” respectively, the appliance maintains the load on the new service at 25 requests per second until you change either parameter. When you want the new service to exit the slow start mode and receive as many requests as the existing services, set the new service request rate parameter to 0.

As an example, assume that you are using a virtual server to load balance 2 services, Service1 and Service2, in round robin mode. Further assume that the virtual server is receiving 240 requests per second, and that it is distributing the load evenly across the services. When a new service, Service3, is added to the configuration, you might want to increase the load on it manually through values of 10, 20, and 40 requests per second before sending it its full share of the load. The following table shows the values to which you set the three parameters.

Table 1. Parameter Values

Parameter	Value
Interval in seconds	0

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter	Value
New service request rate	10, 20, 40, and 0, at intervals that you choose
Units for the new service request rate	Requests per second

When you set the new service request rate parameter to 0, Service3 is no longer considered a new service, and receives its full share of the load.

Assume that you add another service, Service4, during the ramp-up period for Service3. In this example, Service4 is added when the new service request rate parameter is set to 40. Therefore, Service4 begins receiving 40 requests per second.

The following table shows the load distribution on the services during the period described in this example.

Table 2. Load Distribution on Services when Manually Stepping Up the Load

	new service request rate = 10 req/sec (Service3added)	new service request rate = 20 req/sec	new service request rate = 40 req/sec (Service4added)	new service request rate = 0 req/sec (new services exit slow start mode)
Service1	115	110	80	60
Service2	115	110	80	60
Service3	10	20	40	60
Service4	-	-	40	60
Total req/sec (load on the virtual server)	240	240	240	240

Automated Slow Start

If you want the appliance to increase the load on a new service automatically at specified intervals until the service can be considered capable of handling its full share of the load, set the new service request rate parameter, the units parameter, and the increment interval. When all the parameters are set to values other than 0, the appliance increments the load on a new service by the value of the new service request rate, at the specified interval, until the service is receiving it's full share of the load.

As an example, assume that four services, Service1, Service2, Service3, and Service4, are bound to a load balancing virtual server, vserver1. Further assume that vserver1 receives 100 requests per second, and that it distributes the load evenly across the services (25 requests per second per service). When you add a fifth service, Service5, to the configuration, you might want the appliance to send the new service 4 requests per second for the first 10 seconds, 8 requests per second for the next 10 seconds, and so on, until it is receiving 20 requests per second. For this requirement, the following table shows the values to which you set the three parameters:

Table 3. Parameter Values

Parameter	Value
Interval in seconds	10
Increment value	4

© 1999-2017 Citrix Systems, Inc. All rights reserved.

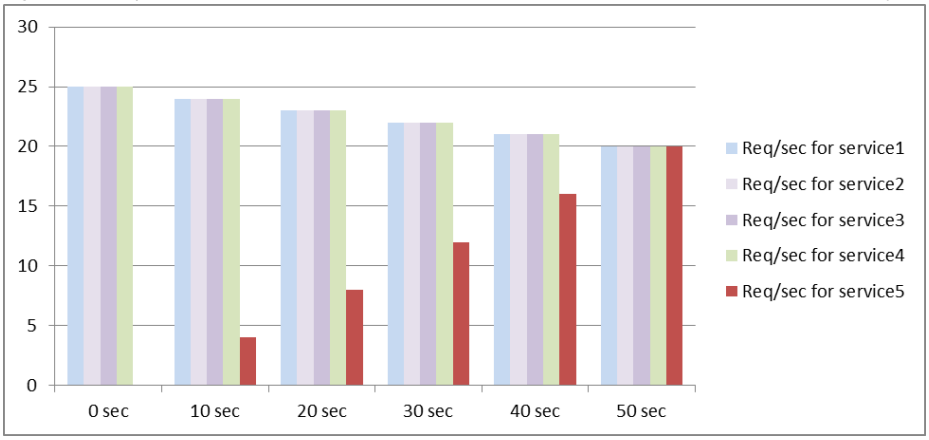
Parameter	Value
Cutoff for the new service request rate	Requests per second

With this configuration, the new service begins receiving as many requests as the existing services 50 seconds after it is added or its state has changed from DOWN to UP. During each interval in this period, the appliance distributes to the existing servers the excess of requests that would have been sent to the new service in the absence of stepwise increments. For example, in the absence of stepwise increments, each service, including Service5, would have received 20 requests each per second. With stepwise increments, during the first 10 seconds, when Service5 receives only 4 requests per second, the appliance distributes the excess of 16 requests per second to the existing services, resulting in the distribution pattern shown in the following table and figure over the 50-second period. After the 50-second period, Service5 is no longer considered a new service, and it receives its normal share of traffic.

Table 4. Load Distribution Pattern on All Services for the 50-second Period Immediately after Service5 is Added

	0 sec	10 sec	20 sec	30 sec	40 sec	50 sec
Req/sec forService1	25	24	23	22	21	20
Req/sec forService2	25	24	23	22	21	20
Req/sec forService3	25	24	23	22	21	20
Req/sec forService4	25	24	23	22	21	20
Req/sec forService5	0	4	8	12	16	20
Total req/sec (load on the virtual server)	100	100	100	100	100	100

Figure 1. A Graph of the Load Distribution Pattern on All Services for the 50-second Period Immediately after Service5 is Added



An alternative requirement might be for the appliance to send Service5 25% of the load on the existing services in the first 5 seconds, 50% in the next 5 seconds, and so on, until it is receiving 20 requests per second. For this requirement, the following table shows the values to which you set the three parameters.

Table 5. Parameter Values

Parameter	Value
Interval in seconds	5
Increment value	25

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter	Value
Units for the new service request rate	Percent

With this configuration, the service begins receiving as many requests as the existing services 20 seconds after it is added or its state has changed from DOWN to UP. The traffic distribution during the ramp-up period for the new service is identical to the one described earlier, where the unit for the step increments was "requests per second."

Setting the Slow Start Parameters

You set the slow start parameters by using either the **set lb vserver** or the **add lb vserver** command. The following command is for setting slow start parameters when adding a virtual server.

To configure stepwise load increments for a new service by using the command line interface

At the command prompt, type the following commands to configure stepwise increments in the load for a service and verify the configuration:

- add lb vserver <name> <serviceType> <IPAddress> <port> [-newServiceRequest <positive_integer>] [<newServiceRequestUnit>] [-newServiceRequestIncrementInterval <positive_integer>]
- show lb vserver <name>

Example

```
> set lb vserver BR_LB -newServiceRequest 5 PER_SECOND -newServiceRequestIncrementInterval 10
 Done
> show lb vserver BR_LB
 BR_LB (192.0.2.33:80) - HTTP Type: ADDRESS
 State: UP
            ...
            ...
 New Service Startup Request Rate: 5 PER_SECOND, Increment Interval: 10
            ...
            ...
Done
>
```

To configure stepwise load increments for a new service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Method, and set the following slow start parameters:
   - New Service Startup Request Rate.
   - New Service Request Unit.
   - Increment Interval.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The No-Monitor Option for Services

Nov 11, 2013

If you use an external system to perform health checks on the services and do not want the NetScaler appliance to monitor the health of a service, you can set the no-monitor option for the service. If you do so, the appliance does not send probes to check the health of the service but shows the service as UP. Even if the service goes DOWN, the appliance continues to send traffic from the client to the service as specified by the load balancing method.

The monitor can be in the ENABLED or DISABLED state when you set the no-monitor option, and when you remove the no-monitor option, the earlier state of the monitor is resumed.

You can set the no-monitor option for a service when creating the service. You can also set the no-monitor option on an existing service.

The following are the consequences of setting the no-monitor option:

- If a service for which you enabled the no-monitor option goes down, the NetScaler continues to show the service as UP and continues to forward traffic to the service. A persistent connection to the service can worsen the situation. In that case, or if many services shown as UP are actually DOWN, the system may fail. To avoid such a situation, when the external mechanism that monitors the services reports that a service that is DOWN, remove the service from the NetScaler configuration.
- If you configure the no-monitor option on a service, you cannot configure load balancing in the Direct Server Return (DSR) mode. For an existing service, if you set the no-monitor option, you cannot configure the DSR mode for the service.

To set the no-monitor option for a new service by using the command line interface

At the command prompt, type the following commands to create a service with the health monitor option, and verify the configuration:

add service <serviceName> <IP | serverName> <serviceType> <port> -healthMonitor (YES|NO)

Example

```
 >add service nomonsrvc 10.102.21.21 http 80
 -healthMonitor no
 Done
 > show service nomonsrvc
nomonsrvc (10.102.21.21:80) - HTTP
State: UP
Last state change was at Mon Nov 15 22:41:29 2010
Time since last state change: 0 days, 00:00:00.970
Server Name: 10.102.21.21
 Server ID : 0 Monitor Threshold : 0
...
Access Down Service: NO
...
Down state flush: ENABLED
Health monitoring: OFF
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

1 bound monitor:
1) Monitor Name: tcp-default
State: UNKNOWN Weight: 1
Probes: 3 Failed [Total: 3 Current: 3]
Last response: Probe skipped - Health monitoring is turned off.
Response Time: N/A
Done

To set the no-monitor option for an existing service by using the command line interface

At the command prompt, type the following command to set the health monitor option:

```
set service <name> -healthMonitor (YES | NO)
```

Example

By default, the state of a service and the state of the corresponding monitor are UP.
>show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP

1) Monitor Name: http-ecv
State: UP Weight: 1
Probes: 99992 Failed [Total: 0 Current: 0]
Last response: Success - Pattern found in response.
Response Time: 3.76 millisec
Done

When the no-monitor option is set on a service, the state of the monitor changes to UNKNOWN.
> set service LB-SVC1 -healthMonitor NO
Done
> show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP
Last state change was at Fri Dec 10 10:17:37 2010.
Time since last state change: 5 days, 18:55:48.710
Health monitoring: OFF

1) Monitor Name: http-ecv
State: UNKNOWN Weight: 1
Probes: 100028 Failed [Total: 0 Current: 0]
Last response: Probe skipped - Health monitoring is turned off.
Response Time: 0.0 millisec
Done
When the no-monitor option is removed, the earlier state of the monitor is resumed.
> set service LB-SVC1 -healthMonitor YES
Done
>show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
State: UP
Last state change was at Fri Dec 10 10:17:37 2010
Time since last state change: 5 days, 18:57:47.880
1) Monitor Name: http-ecv
  State: UP Weight: 1
  Probes: 100029 Failed [Total: 0 Current: 0]
  Last response: Success - Pattern found in response.
  Response Time: 5.690 millisec
  Done
```

To set the no-monitor option for a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and clear Health Monitoring.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Protecting Applications on Protected Servers Against Traffic Surges

Jun 08, 2015

The NetScaler provides the surge protection option to maintain the capacity of a server or cache. The NetScaler regulates the flow of client requests to servers and controls the number of clients that can simultaneously access the servers. The NetScaler blocks any surges passed to the server, thereby preventing overloading of the server.

For surge protection to function correctly, you must enable it globally. For more information about surge protection, see "Surge Protection."

To set surge protection on the service by using the command line interface

At the command prompt, type:

```
set service <name> -sp <Value>
```

Example

```
set service Service-HTTP-1 -sp ON
```

To set surge protection on the service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a source.
2. In Advanced Settings, select Traffic Settings, and select Surge Protection.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Cleanup of Virtual Server and Service Connections

Feb 08, 2016

The state of a virtual server depends on the states of the services bound to it. The state of each service depends on the responses of the load balanced servers to probes or health checks sent by the monitors that are bound to that service. Sometimes the load balanced servers do not respond. If a server is slow or busy, monitoring probes can time out. If repeated monitoring probes are not answered within the configured timeout period, the service is marked DOWN. If a service or virtual server are marked DOWN, the server and client side connections must be flushed. Terminating existing connections frees resources, and in certain cases speeds recovery of overloaded load balancing setups.

Under certain conditions, you can configure the **downStateFlush** setting to immediately terminate existing connections when a service or a virtual server is marked DOWN. You must not enable the downStateFlush setting on those application servers that must complete their transactions. You can enable this setting on Web servers whose connections can safely be terminated when they marked DOWN.

The following table summarizes the effect of this setting on an example configuration consisting of a virtual server, Vserver-LB-1, with one service bound to it, Service-1. In the table, E and D denote the state of the downStateFlush setting: E means Enabled, and D means Disabled.

Vserver-LB-1	Service-1	State of Connections
E	E	Both client and server connections are terminated.
E	D	For some service types, such as TCP, for which the NetScaler appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only client connections are terminated.
D	E	For some service types, such as TCP, for which the NetScaler appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only server connections are terminated.
D	D	Neither client nor server connections are terminated.

If you want to disable a service only when all the established connections are closed by the server or the client, you can use the graceful shutdown option. For information about the graceful shutdown of a service, see Graceful Shutdown of

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Services.

To set down state flush on the service by using the command line interface

At the command prompt, type:

```
set service <name>  -downStateFlush (ENABLED | DISABLED )
```

Example

```
set service Service-HTTP-1 -downStateFlush enabled
```

To set down state flush on the service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Down State Flush.

To set down state flush on the virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <name>  -downStateFlush (ENABLED | DISABLED )
```

Example

```
set lb vserver vsvr1 -downStateFlush enabled
```

To set down state flush on the virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Down State Flush.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Graceful Shut down of Services

Jun 29, 2017

During scheduled network outages such as system upgrades or hardware maintenance, you may have to close or disable some services. You can later enable the service by using the "enable service <name>" command.

To avoid disrupting established sessions, you can place a service in the Transition Out of Service (TROFS) state by doing one of the following:

- Adding a TROFS code or string to the monitor—Configure the server to send a specific code or string in response to a monitor probe.
- Explicitly disable the service and:
  - Set a delay (in seconds).
  - Enable graceful shut down.

Adding a TROFS Code or String

If you bind only one monitor to a service, and the monitor is TROFS-enabled, it can place the service in the TROFS state on the basis of the server's response to a monitor probe. This response is compared with the value in the trofsCode parameter for an HTTP monitor or the trofsString parameter for an HTTP-ECV or TCP-ECV monitor. If the code matches, the service is placed in the TROFS state. In this state, it continues to honor the persistent connections.

If multiple monitors are bound to a service, the effective state of the service is calculated on the basis of the state of all the monitors that are bound to the service. Upon receiving a TROFS response, the state of the TROFS-enabled monitor is considered as UP for the purpose of this calculation. For more information about how a NetScaler appliance designates a service as UP, see Setting a Threshold Value for the Monitors Bound to a Service.

Important:

- You can bind multiple monitors to a service, but must not TROFS-enable more than one of them.
- You can convert a TROFS-enabled monitor to a monitor that is not TROFS-enabled, but not vice versa.

To configure a TROFS code or string in a monitor by using the command line interface

At the command prompt, type one of the following commands:

add lb monitor <monitor-name> **HTTP** –trofsCode <respcode>

add lb monitor <monitor-name> **HTTP-ECV** –trofsString <resp string>

add lb monitor <monitor-name> **TCP-ECV** –trofsString <resp string>

To modify the TROFS code or string by using the command line interface

At the command prompt, type one of the following commands:

set lb monitor <trofs monitorname> **HTTP** -trofscode <newcode>

set lb monitor <trofs monitorname> **HTTP-ECV** -trofsstring <new string>

set lb monitor <trofs monitorname> **TCP-ECV** -trofsstring <new string>

Note: You can use the set command only if a TROFS-enabled monitor was added earlier. You cannot use this command to set the TROFS code or string for a monitor that is not TROFS-enabled.

To configure a TROFS code or string in a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. On the Monitors pane, click Add, and do one of the following:
   - Select Type as HTTP, and specify a TROFS Code.
   - Select Type as HTTP-ECV or TCP-ECV, and specify a TROFS String.

Disabling a Service

Often, however, you cannot estimate the amount of time needed for all the connections to a service to complete the existing transactions. If a transaction is unfinished when the wait time expires, shutting down the service may result in data loss. In this case, you can specify graceful shutdown for the service, so that the service is disabled only when all the current active client connections are closed by either the server or the client. See the following table for behavior if you specify a wait time in addition to graceful shutdown.

Persistence is maintained according to the specified method even if you enable graceful shutdown. The system continues to serve all the persistent clients, including

© 1999-2017 Citrix Systems, Inc. All rights reserved.

new connections from the clients, unless the service is marked DOWN during the graceful shutdown state as a result of the checks made by a monitor.

The following table describes graceful shut down options.

Table 1. Graceful Shut down Options

State	Results
Graceful shutdown is enabled and a wait time is specified.	Service is shut down after the last of the current active client connections is served, even if the wait time has not expired. The appliance checks the status of the connections once every second. If the wait time expires, any open sessions are closed.
Graceful shutdown is disabled and a wait time is specified.	Service is shut down only after the wait time expires, even if all established connections are served before expiration.
Graceful shutdown is enabled and no wait time is specified.	Service is shut down only after the last of the previously established connections is served, regardless of the time taken to serve the last connection.
Graceful shutdown is disabled and no wait time is specified.	No graceful shutdown. Service is shut down immediately after the disable option is chosen or the disable command is issued. (The default wait time is zero seconds.)

To terminate existing connections when a service or a virtual server is marked DOWN, you can use the Down State Flush option. For more information, see Enabling Cleanup of Virtual Server Connections.

To configure graceful shutdown for a service by using the command line interface

At the command prompt, type the following commands to shut down a service gracefully and verify the configuration:

- disable service <name> [<delay>] [-graceFul (YES | NO)]
- show service <name>

Example

```
> disable service svc1 6000 -graceFul YES
Done
>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: GOING OUT OF SERVICE (Graceful, Out Of Service in 5998 seconds)
Last state change was at Mon Nov 15 22:44:15 2010
Time since last state change: 0 days, 00:00:01.160
...
Down state flush: ENABLED

1 bound monitor:
1) Monitor Name: tcp-default
State: UP          Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Last response: Probe skipped - live traffic to service.
Response Time: N/A
Done

>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: OUT OF SERVICE
Last state change was at Mon Nov 15 22:44:19 2010
Time since last state change: 0 days, 00:00:03.250
Down state flush: ENABLED

1 bound monitor:
1)  Monitor Name: tcp-default
State: UNKNOWN          Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - service state OFS.
Response Time: N/A
Done
```

To configure graceful shutdown for a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and from the Action list, click Disable. Enter a wait time, and select Graceful.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Directing Requests to a Custom Web Page

Jun 08, 2015

For SureConnect to function correctly, you must set it globally. The NetScaler provides the SureConnect option to ensure the response from an application. For more information about the SureConnect option, see "Sure Connect."

To set SureConnect on the service by using the command line interface

At the command prompt, type:

```
set service <name> -sc <Value>
```

Example

```
set service Service-HTTP-1 -sc ON
```

To set SureConnect on the service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Sure Connect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Access to Services When Down

Jun 08, 2015

You can enable access to a service when it is disabled or in a DOWN state by configuring the NetScaler appliance to use Layer 2 mode to bridge the packets sent to the service. Normally, when requests are forwarded to services that are DOWN, the request packets are dropped. When you enable the Access Down setting, however, these request packets are sent directly to the load balanced servers.

For more information about Layer 2 and Layer 3 modes, see IP Addressing.

For the appliance to bridge packets sent to the DOWN services, enable Layer 2 mode with the accessDown parameter.

To enable access down on a service by using the command line interface

At the command prompt, type:

set service <name> -accessDown <Value>

Example

```
set service Service-HTTP-1 -accessDown YES
```

To enable access down on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Access Down.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling TCP Buffering of Responses

Jun 08, 2015

The NetScaler appliance provides a TCP buffering option that buffers only responses from the load balanced server. This enables the appliance to deliver server responses to the client at the maximum speed that the client can accept them. The appliance allocates from 0 through 4095 megabytes (MB) of memory for TCP buffering, and from 4 through 20480 kilobytes (KB) of memory per connection.

Note: TCP buffering set at the service level takes precedence over the global setting. For more information about configuring TCP buffering globally, see "TCP Buffering."

To enable TCP Buffering on a service by using the command line interface

At the command prompt, type:

```
set service <name> -TCPB <Value>
```

Example

```
set service Service-HTTP-1 -TCPB YES
```

To enable TCP Buffering on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select TCP Buffering.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Compression

Jun 08, 2015

The NetScaler appliance provides a compression option to transparently compress HTML and text files by using a set of built-in compression policies. Compression reduces bandwidth requirements and can significantly improve server responsiveness in bandwidth-constrained setups. The compression policies are associated with services bound to the virtual server. The policies determine whether a response can be compressed and send compressible content to the appliance, which compresses it and sends it to the client.

Note: For compression to function correctly, you must enable it globally. For more information about configuring compression globally, see Compression.

To enable compression on a service by using the command line interface

At the command prompt, type:

```
set service <name> -CMP <YES | NO>
```

Example

```
set service Service-HTTP-1 -CMP YES
```

To enable compression on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Compression.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Maintaining Client Connection for Multiple Client Requests

Jun 08, 2015

You can set the client keep-alive parameter to configure an HTTP or SSL service to keep a client connection to a Web site open across multiple client requests. If client keep-alive is enabled, even when the load balanced Web server closes a connection, the NetScaler appliance keeps the connection between the client and itself open. This setting allows services to serve multiple client requests on a single client connection.

If you do not enable this setting, the client will open a new connection for every request that it sends to the Web site. The client keep-alive setting saves the packet round trip time required to establish and close connections. This setting also reduces the time to complete each transaction. Client keep-alive can be enabled only on HTTP or SSL service types.

Client keep-alive set at the service level takes precedence over the global client keep-alive setting. For more information about client keep-alive, see Client Keep-Alive.

To enable client keep-alive on a service by using the command line interface

At the command prompt, type:

set service <name> -CKA <Value>

Example

```
set service Service-HTTP-1 -CKA YES
```

To enable client keep-alive on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Client Keep-Alive.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Inserting the IP Address of the Client in the Request Header

Oct 06, 2016

A NetScaler uses the subnet IP (SNIP) address to connect to the server. The server need not be aware of the client.

However, in some situations, the server needs to be aware of the client it has to serve. When you enable the client IP setting, the NetScaler inserts the client's IPv4 or IPv6 address while forwarding the requests to the server. The server inserts this client IP in the header of the responses. The server is thus aware of the client.

To insert client IP address in the client request by using the command line interface

At the command prompt, type:

set service <name> -CIP <Value> <cipHeader>

Example

```
set service Service-HTTP-1 -CIP enabled X-Forwarded-For
```

To insert client IP address in the client request by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**, and open a service.
2. In Advanced Settings, select **Traffic Settings**, and select **Client IP Address**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using the Source IP Address of the Client When Connecting to the Server

Jun 08, 2015

You can configure the NetScaler appliance to forward packets from the client to the server without changing the source IP address. This is useful when you cannot insert the client IP address into a header, such as when working with non-HTTP services.

For more information about configuring USIP globally, see "Enabling Use Source IP Mode."

To enable USIP mode for a service by using the command line interface

At the command prompt, type:

set service <name> -usip (**YES** | **NO**)

Example

```
set service Service-HTTP-1 -usip YES
```

To enable USIP mode for a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Use Source IP Address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Source Port for Server-Side Connections

Jun 08, 2015

When the NetScaler appliance connects to a physical server, it can use the source port from client's request, or it can use a proxy port as the source port for the connection. You can set the Use Proxy Port parameter to YES to handle situations such as the following scenario:

- The NetScaler appliance is configured with two load balancing virtual servers, LBVS1 and LBVS2.
- Both the virtual servers are bound to the same service, S-ANY.
- Use (the client's) source IP address (USIP) is enabled on the service.
- Client C1 sends two requests, Req1 and Req2, for the same service.
- Req1 is received by LBVS1 and Req2 is received by LBVS2.
- LBVS1 and LBVS2 forward the request to S-ANY, and when S-ANY sends the response, they forward the response to the client.
- Consider two cases:
  - Use the client port. When the NetScaler uses the client port, both the virtual servers use the client's IP address (because USIP is ON) and the client's port when connecting to the server. Therefore, when the service sends the response, the NetScaler cannot determine which virtual server should receive the response.
  - Use proxy port. When the NetScaler uses a proxy port, the virtual servers use the client's IP address (because USIP is ON), but different ports when connecting to the server. Therefore, when the service sends the response, the port number identifies the virtual server that should receive the response.

However, if you require a fully transparent configuration, such as a fully transparent cache redirection configuration, you must disable the Use Proxy port Setting so that the NetScaler appliance can use the source port from the client’s request.

The Use Proxy Port option becomes relevant if the use source IP (USIP) option is enabled. For TCP-based service types, such as TCP, HTTP, and SSL, the option is enabled by default. For UDP-based service types, such as UDP and DNS, including ANY, the option is disabled by default. For more information about the USIP option, see "Enabling Use Source IP Mode."

You can configure the Use Proxy Port setting either globally or on a given service.

Configuring the Use Proxy Port Setting on a Service

You configure the Use ProxyPort settingon the service if you want to override the global setting.

To configure the Use Proxy Port setting on a service by using the command line interface

At the command prompt, type:

set service <name> -useProxyPort (YES | NO)

Example

```
> set service svc1 -useproxyport YES
 Done > show service svc1
 svc1 (10.102.29.30:80) - HTTP
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
State: UP
. . .
Use Source IP: YES  Use Proxy Port: YES
. . .
Done
>
```

To configure the Use Proxy Port setting on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Use Proxy Port.

Configuring the Use Proxy Port Setting Globally

You configure the Use Proxy Port setting globally if you want to apply the setting to all the services on the NetScaler appliance. The global setting is overridden by service-specific Use Proxy Port settings.

To configure the Use Proxy Port setting globally by using the command line interface

At the command prompt, type the following commands to configure the Use Proxy Port setting globally and verify the configuration:

- set ns param -useproxyport (ENABLED | DISABLED)
- show ns param

Example

```
> set ns param -useproxyport ENABLED
 Done
> show ns param
 Global configuration settings:
            . . .
            Use Proxy Port: ENABLED
 Done
>
```

To configure the Use Proxy Port setting globally by using the configuration utility

Navigate to System > Settings > Change global system settings, and select or clear Use Proxy Port.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting a Limit on the Number of Client Connections

Nov 11, 2013

You can specify a maximum number of client connections that each load balanced server can handle. The NetScaler appliance then opens client connections to a server only until this limit is reached. When the load balanced server reaches its limit, monitor probes are skipped, and the server is not used for load balancing until it has finished processing existing connections and frees up capacity.

For more information on the Maximum Client setting, see Load Balancing Domain-Name Based Services.

Note: Connections that are in the process of closing are not considered for this limit.

To set a limit to the number of client connections by using the command line interface

At the command prompt, type:

```
set service <name> -maxclient <Value>
```

Example

```
set service Service-HTTP-1 -maxClient 1000
```

To set a limit to the number of client connections by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Maximum Clients.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting a Limit on Number of Requests Per Connection to the Server

Aug 29, 2013

The NetScaler appliance can be configured to reuse connections to improve performance. In some scenarios, however, load balanced Web servers may have issues when connections are reused for too many requests. For HTTP or SSL services, use the max request option to limit the number of requests sent through a single connection to a load balanced Web server.

Note: You can configure the max request option for HTTP or SSL services only.

To limit the number of client requests per connection by using the command line interface

At the command prompt, type:

set service <ServiceName> -maxReq <Value>

Example

```
set service Service-HTTP-1 -maxReq 100
```

To limit the number of client requests per connection by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Maximum Requests.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting a Threshold Value for the Monitors Bound to a Service

Nov 11, 2013

The NetScaler appliance designates a service as UP only when the sum of the weights of all monitors bound to it and that are UP is equal to or greater than the threshold value configured on the service. The weight for a monitor specifies how much that monitor contributes to designating the service to which it is bound as UP.

For example, assume that three monitors, named Monitor-HTTP-1, Monitor-HTTP-2, and Monitor-HTTP-3 respectively, are bound to Service-HTTP-1, and that the threshold configured on the service is three. Suppose the following weights are assigned to each monitor:

- The weight of Monitor-HTTP-1 is 1.
- The weight of Monitor-HTTP-2 is 3.
- The weight of Monitor-HTTP-3 is 1.

The service is marked UP only when one of the following is true:

- Monitor-HTTP-2 is UP.
- Monitor-HTTP-2 and Monitor-HTTP-1 or Monitor-HTTP-3 are UP
- All three monitors are UP.

To set the monitor threshold value on a service by using the command line interface

At the command prompt, type:

set service <name> -monThreshold <Value>

Example

```
set service Service-HTTP-1 -monThreshold 100
```

To set the monitor threshold value on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Monitor Threshold.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting a Timeout Value for Idle Client Connections

Nov 11, 2013

You can configure the service with a time-out value to terminate any idle client connections when the configured time elapses. If the client is idle during the configured time, the NetScaler closes the client connection.

To set a timeout value for idle client connections by using the command line interface

At the command prompt, type:

set service <name> -cltTimeout <Value>

Example

```
set service Service-HTTP-1 -cltTimeout 100
```

To set a timeout value for idle client connections by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Client Idle Time-out.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting a Timeout Value for Idle Server Connections

Nov 04, 2016

You can configure a service with a timeout value to terminate any idle server connections when the configured time (in seconds) elapses. If the server is idle for the configured amount of time, the NetScaler appliance closes the server connection.

To set a timeout value for idle server connections by using the command line interface

At the command prompt, type:

set service <name> -svrTimeout <Value>

Example

```
set service Service-HTTP-1 -svrTimeout 100
```

To set a timeout value for idle server connections by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Server Idle Time-out.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting a Limit on the Bandwidth Usage by Clients

Jun 29, 2017

In some cases, servers may have limited bandwidth to handle client requests and may become overloaded. To prevent overloading a server, you can specify a maximum limit on the bandwidth, in Kbps, processed by the server. The NetScaler appliance forwards requests to a load balanced server only until this limit is reached.

To set a maximum bandwidth limit on a service by using the command line interface

At the command prompt, type:

set service <name> -maxBandwidth <Value>

Example

```
set service Service-HTTP-1 -maxBandwidth 100
```

To set a maximum bandwidth limit on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Maximum Bandwidth.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Redirecting Client Requests to a Cache

Nov 11, 2013

You can configure a service to redirect client requests to a cache, and forward only those requests that are cache misses to a service chosen by the configured load balancing method.

To set cache redirection on a service by using the command line interface

At the command prompt, type:

set service <name> -cacheable <Value>

Example

```
set service Service-HTTP-1 -cacheable YES
```

To set cache redirection on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open a service, and set the Cache Type.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Retaining the VLAN Identifier for VLAN Transparency

May 11, 2017

You can configure a load balancing virtual server to retain the client's VLAN identifier in packets that are to be forwarded to servers. The virtual server must be a wildcard virtual server of type ANY, and must be functioning in MAC mode.

To configure a load balancing virtual server to retain the client VLAN ID by using the command line interface

At the command prompt, type the following command to configure a load balancing virtual server to retain the client VLAN ID and verify the configuration:

- set lb vserver <name> -m MAC -macmodeRetainvlan ENABLED
- show lb vserver <name>

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

To configure a load balancing virtual server to retain the client VLAN ID by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Retain VLAN ID.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Automatic State Transition Based on Percentage Health of Bound Services

Nov 11, 2013

You can configure a load balancing virtual server to automatically transition from the UP state to the DOWN state if the percentage of active services falls below a configured threshold. For example, if you bind 10 services to a load balancing virtual server and configure a threshold of 50% for that virtual server, it transitions from UP to DOWN if six or more services are DOWN. When the percentage health rises above the threshold value, the virtual server returns to the UP state.

You can also enable an SNMP alarm called ENTITY-STATE if you want the NetScaler appliance to notify you when the percentage health of bound services causes a virtual server to change state.

To configure percentage based automatic state transition by using the command line interface

At the command prompt, type the following commands to configure automatic state transition for a virtual server and verify the configuration:

- set lb vserver <name> -healthThreshold <positive_integer>
- show lb vserver <name>

To configure percentage based automatic state transition by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and set a Health Threshold.

To enable the ENTITY-STATE alarm by using the command line interface

At the command prompt, type the following commands to enable the ENTITY-STATE SNMP alarm and verify the configuration:

- enable snmp alarm ENTITY-STATE
- show snmp alarm

To enable the ENTITY-STATE alarm by using the configuration utility

1. Navigate to System > SNMP > Alarms.
2. Select ENTITY-STATE and, in the Action list, select Enable.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The Built-in Monitors

Oct 21, 2015

The NetScaler appliance contains a number of built-in monitors that you can use to monitor your services. These built-in monitors handle most of the common protocols. You cannot modify or remove the built-in monitors; you can only bind a built-in monitor to a service and unbind it from the service.

Note: You can create a custom monitor based on a built-in monitor. To learn how to create custom monitors, see Configuring Monitors in a Load Balancing Setup.

This section includes the following details:

Monitoring TCP-based Applications Monitoring SSL Services Monitoring FTP Services Monitoring SIP Services Monitoring RADIUS Services Monitoring Accounting Information Delivery from a RADIUS Server Monitoring DNS and DNS-TCP Services Monitoring LDAP Services Monitoring MySQL Services Monitoring SNMP Services Monitoring NNTP Services Monitoring POP3 Services Monitoring SMTP Services Monitoring RTSP Services Monitoring the XML Broker Services Monitoring ARP Requests Monitoring the XenDesktop Delivery Controller Services Monitoring Web Interface Services Monitoring Citrix StoreFront Stores

This section includes the following details:

- Monitoring TCP-based Applications
- Monitoring SSL Services
- Monitoring FTP Services
- Monitoring SIP Services
- Monitoring RADIUS Services
- Monitoring Accounting Information Delivery from a RADIUS Server
- Monitoring DNS and DNS-TCP Services
- Monitoring LDAP Services
- Monitoring MySQL Services
- Monitoring SNMP Services
- Monitoring NNTP Services
- Monitoring POP3 Services
- Monitoring SMTP Services
- Monitoring RTSP Services
- Monitoring the XML Broker Services
- Monitoring ARP Requests

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Monitoring the XenDesktop Delivery Controller Services
- Monitoring Web Interface Services
- Monitoring Citrix StoreFront Stores

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring TCP-based Applications

May 26, 2017

The NetScaler appliance has two built-in monitors that monitor TCP-based applications: tcp-default and ping-default. When you create a service, the appropriate default monitor is bound to it automatically, so that the service can be used immediately if it is UP. The tcp-default monitor is bound to all TCP services; the ping-default monitor is bound to all non-TCP services.

You cannot delete or modify default monitors. When you bind any other monitor to a TCP service, the default monitor is unbound from the service. The following table lists the monitor types, and the parameters and monitoring processes associated with each type.

Monitor type	Specific parameters	Process
tcp	Not applicable	The NetScaler appliance establishes a 3-way handshake with the monitor destination, and then closes the connection. If the appliance observes TCP traffic to the destination, it does not send TCP monitoring requests. This occurs if LRTM is disabled. By default, LRTM is disabled on this monitor.
http	httprequest [“HEAD /”] - HTTP request that is sent to the service. respcode [200] - A set of HTTP response codes are expected from the service.	The NetScaler appliance establishes a 3-way handshake with the monitor destination. After the connection is established, the appliance sends HTTP requests, and then compares the response code with the configured set of response codes.
tcp-ecv	send [""] - is the data that is sent to the service. The maximum permissible length of the string is 512 K bytes. recv [""] - expected response from the service. The maximum permissible length of the string is 128 K bytes.	The NetScaler appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send specific data to the service and expects a specific response through the receive parameter. Different servers send different sizes of segments. However, the pattern must be within 16 TCP segments.
http-ecv	send [""] - HTTP data that is sent to the service recv [""] - the expected HTTP response data from the service	The NetScaler appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send the HTTP data to the service and expects the HTTP response that the receive parameter specifies. (HTTP body part without including HTTP headers). Empty response data matches any response. Expected data may be anywhere in the first 24K bytes of the HTTP body of the response.
ping	Not Applicable	The NetScaler appliance sends an ICMP echo request to the destination of the monitor and expects an ICMP echo response.

To configure built-in monitors for TCP-based applications, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring SSL Services

Jun 16, 2014

The NetScaler appliance has built-in secure monitors, TCPS and HTTPS. You can use the secure monitors to monitor HTTP as well as non-HTTP traffic. To configure a secure HTTP monitor, select the monitor type as HTTP, and then set the secure flag. To configure a secure TCP monitor, select the monitor type as TCP, and then set the secure flag.The secure monitors work as described below:

- **Secure TCP monitoring.** The NetScaler appliance establishes a TCP connection. After the connection is established, the appliance performs an SSL handshake with the server. After the handshake is over, the appliance closes the connection.
- **Secure HTTP monitoring.** The NetScaler appliance establishes a TCP connection. After the connection is established, the appliance performs an SSL handshake with the server. When the SSL connection is established, the appliance sends HTTP requests over the encrypted channel and checks the response codes.

The following table describes the available built-in monitors for monitoring SSL services.

Monitor type	Probe	Success criteria (Direct condition)
TCP	TCP connection SSL handshake	Successful TCP connection established and successful SSL handshake.
HTTP	TCP connection SSL handshake Encrypted HTTP request	Successful TCP connection is established, successful SSL handshake is performed, and expected HTTP response code in server HTTP response is encrypted.
TCP-ECV	TCP connection SSL handshake (Data sent to a server is encrypted.)	Successful TCP connection is established, successful SSL handshake is performed, and expected TCP data is received from the server.
HTTP-ECV	TCP connection SSL handshake (Encrypted HTTP request)	Successful TCP connection is established, successful SSL handshake is performed, and expected HTTP data is received from the server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring FTP Services

Oct 20, 2015

To monitor FTP services, the NetScaler appliance opens two connections to the FTP server. It first connects to the control port, which is used to transfer commands between a client and an FTP server. After it receives the expected response, it connects to the data port, which is used to transfer files between a client and an FTP server. Only when the FTP server responds as expected on both connections is it marked UP.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.
The NetScaler appliance has two built-in monitors for FTP services: the FTP monitor and the FTP-EXTENDED monitor. The FTP monitor checks basic functionality; the FTP-EXTENDED monitor also verifies that the FTP server is able to transmit a file correctly.

Parameter	Specifies
userName	User name used in the probe. Applies to both the FTP and FTP-EXTENDED monitor.
password	Password used in monitoring. Applies to both the FTP and FTP-EXTENDED monitor
fileName	File name to be used for FTP-EXTENDED monitor only.

To configure built-in monitors to check the state of FTP services, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring SIP Services

Oct 14, 2014

A NetScaler ADC has two built-in monitors that you can use to monitor SIP services: the **SIP-UDP** and **SIP-TCP** monitors. A SIP monitor periodically checks the SIP service to which the SIP monitor is bound, by sending SIP request methods to the SIP service. If the SIP service replies with a response code, the monitor marks the service as UP. If the SIP service does not respond, or responds incorrectly, it is marked as DOWN.

Parameter	Specifies
sipURI	SIP addressing schema of the SIP server.
sipmethod	Type of SIP request used to probe the SIP service. Specify one of the following methods: • INVITE • OPTION (the default) • REGISTER
respcode	SIP response code with which the SIP service responds the probe request. Default: 200.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring RADIUS Services

Oct 20, 2015

The NetScaler appliance RADIUS monitor periodically checks the state of the RADIUS service to which it is bound by sending an authentication request to the service. The RADIUS server authenticates the RADIUS monitor and sends a response. By default, the monitor expects to receive a response code of 2, the default Access-Accept response, from the RADIUS server. As long as the monitor receives the appropriate response, it marks the service UP.

Note: RADIUS monitor supports only PAP type authentication.

- If the client authenticated successfully, the RADIUS server sends an Access-Accept response. The default access-accept response code is 2, and this is the code that the appliance uses.
- If the client fails to authenticate successfully (such as when there is a mismatch in the user name, password, or secret key), the RADIUS server sends an Access-Reject response. The default access-reject response code is 3, and this is the code that the appliance uses.

Parameter	Specifies
userName	User name on the RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 server. This user name is used in the probe.
password	Password used in monitoring RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP servers.
radKey	Shared secret key value that the RADIUS server uses during client authentication.
radNASid	NAS-ID that is encapsulated in the payload when an access request is made.
radNASip	The IP address that is encapsulated in the payload when an access-request is made. When radNASip is not configured, the NetScaler sends the mapped IP address (MIP) to the RADIUS server as the NAS IP address.

To monitor a RADIUS service, you must configure the RADIUS server to which it is bound as follows:

1. Add the user name and password of the client that the monitor will use for authentication to the RADIUS authentication database.
2. Add the IP address and secret key of the client to the appropriate RADIUS database.
3. Add the IP addresses that the appliance uses to send RADIUS packets to the RADIUS database. If the NetScaler appliance has more than one mapped IP address, or if a subnet IP address (SNIP) is used, you must add the same secret key for all of the IP addresses.

   Caution: If the IP address used by the appliance are not added to the RADIUS database, the RADIUS server will discard all packets.

To configure built-in monitors to check the state of RADIUS server, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring Accounting Information Delivery from a RADIUS Server

Sep 13, 2013

You can configure a monitor called a *RADIUS accounting* monitor to determine whether the Radius server used for Authentication, Authorization, and Accounting (AAA) is delivering accounting information as expected. The monitor is of type RADIUS_ACCOUNTING. The probe is generated by a Perl script called nsbmradius.pl, which is located in the /nsconfig/monitors/ directory. The script sends successive accounting request probes to the RADIUS server. The probe is considered successful only if the RADIUS accounting server responds with a packet whose Code field is set to 5, which, according to RFC 2866, indicates an Accounting-Response packet.

When configuring a RADIUS accounting monitor, you must specify a secret key. You can specify optional parameters, each of which represents a RADIUS attribute, such as Acct-Status-Type and Framed-IP-Address. For information about these attributes, see RFC 2865, "Remote Authentication Dial In User Service (RADIUS)," and RFC 2866, "RADIUS Accounting."

To configure a RADIUS accounting monitor by using the command line interface

At the command prompt, type the following commands to configure a RADIUS accounting monitor and verify the configuration:

- add lb monitor <monitorName> RADIUS_ACCOUNTING [-userName <string>] {-password } {-radKey } [-radNASip <ip_addr>] [-radAccountType <positive_integer>] [-radFramedIP <ip_addr>] [-radAPN <string>] [-radMSISDN <string>] [-radAccountSession <string>]
- show lb monitor <monitorName>

Example

```
add lb monitor radAccntMon RADIUS_ACCOUNTING -radKey "8d#>9jr4rV)L7%a2-zW13sM"
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring DNS and DNS-TCP Services

Oct 20, 2015

The NetScaler appliance has two built-in monitors that can be used to monitor DNS services: DNS and DNS-TCP. When bound to a service, either monitor periodically checks the state of that DNS service by sending a DNS query to it. The query resolves to an IPv4 or IPv6 address. That IP address is then checked against the list of test IP addresses that you configure. The list can contain up to five IP addresses. If the resolved IP address matches at least one IP address on the list, the DNS service is marked as up. If the resolved IP does not match any IP addresses on the list, the DNS service is marked as down.

Parameter	Parameter
query	The DNS query (domain name) sent to the DNS service that is being monitored. Default value: "\007" If the DNS query succeeds, the service is marked as UP; otherwise, it is marked as DOWN. For a reverse monitor, if the DNS query succeeds, the service is marked as DOWN; otherwise, it is marked as UP. If no response is received, the service is marked as DOWN.
queryType	The type of DNS query that is sent. Possible values: Address, Zone.
IPAddress	List of IP addresses that are checked against the response to the DNS monitoring probe.
IPv6	Select this check box if the IP address uses IPv6 format.

To configure the built-in DNS or DNS-TCP monitors, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring LDAP Services

Mar 24, 2015

The NetScaler appliance has one built-in monitor that can be used to monitor LDAP services: the LDAP monitor. It periodically checks the LDAP service to which it is bound by authenticating and sending a search query to it. If the search is successful, the service is marked UP. If the LDAP server does not locate the entry, a failure message is sent to the LDAP monitor, and the service is marked DOWN.

You configure the LDAP monitor to define the search that it should perform when sending a query. You can use the Base DN parameter to specify a location in the directory hierarchy where the LDAP server should start the test query. You can use the Attribute parameter to specify an attribute of the target entity.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

Parameter	Specifies
baseDN	Base name for the LDAP monitor from where the LDAP search must start. If the LDAP server is running locally, the default value of base is dc=netscaler, dc=com.
bindDN	BDN name for the LDAP monitor.
filter	Filter for the LDAP monitor.
password	Password used in monitoring LDAP servers.
attribute	Attribute for the LDAP monitor.

To configure the built-in LDAP monitor, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring MySQL Services

Mar 24, 2015

The NetScaler appliance has one built-in monitor that can be used to monitor MySQL services: the MySQL monitor. It periodically checks the MySQL service to which it is bound by sending a search query to it. If the search is successful, the service is marked UP. If the MySQL server does not respond or the search fails, a failure message is sent to the MySQL monitor, and the service is marked DOWN.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

Parameter	Specifies
database	Database that is used for the MySQL monitor.
sqlQuery	SQL query that is used for the MySQL monitor.

To configure built-in MySQL monitor, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring SNMP Services

Mar 24, 2015

The NetScaler appliance has one built-in monitor that can be used to monitor SMNP services: the SNMP monitor. It periodically checks the SNMP agent on the service to which it is bound by sending a query for the enterprise identification ID (OID) that you configure for monitoring. If the query is successful, the service is marked UP. If the SNMP service finds the OID that you specified, the query succeeds and the SNMP monitor marks the service UP. If it does not find the OID, the query fails and the SNMP monitor marks service DOWN.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

Parameter	Specifies
SNMPOID	OID that is used for the SNMP monitor.
snmpCommunity	Community that is used for the SNMP monitor.
snmpThreshold	Threshold that is used for the SNMP monitor.
snmpVersion	SNMP version that is used for load monitoring. Possible Values: V1, V2.

To configure the built-in SNMP monitor, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring NNTP Services

Mar 24, 2015

The NetScaler appliance has one built-in monitor that can be used to monitor NNTP services: the NNTP monitor. It periodically checks the NNTP service to which it is bound by connecting to the service and checking for the existence of the newsgroup that you specify. If the newsgroup exists, the search is successful and the service is marked UP. If the NNTP service does not respond or the search fails, the service is marked DOWN.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.
The NNTP monitor can optionally be configured to post a test message to the newsgroup as well.

Parameter	Specifies
userName	User name on the RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 server. This user name is used in the probe.
password	Password used in monitoring RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP servers.
group	Group name to be queried for NNTP monitor.

To configure the built-in NNTP monitor, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring POP3 Services

Mar 24, 2015

The NetScaler appliance has one built-in monitor that can be used to monitor POP3 services: the POP3 monitor. It periodically checks the POP3 service to which it is bound by opening a connection with a POP3 server. If the POP3 server responds with the correct response codes within the configured time period, it marks the service UP. If the POP3 service does not respond, or responds incorrectly, it marks the service DOWN.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

Parameter	Specifies
userName	User name POP3 server. This user name is used in the probe.
password	Password used in monitoring POP3 servers.
scriptName	The path and name of the script to execute.
dispatcherIP	The IP address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.

To configure the built-in POP3 monitor, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring SMTP Services

Mar 24, 2015

The NetScaler appliance has one built-in monitor that can be used to monitor SMTP services: the SMTP monitor. It periodically checks the SMTP service to which it is bound by opening a connection with it and conducting a series of handshakes to ensure that the server is operating correctly. If the SMTP service completes the handshakes properly, the monitor marks the service UP. If the SMTP service does not respond, or responds incorrectly, it marks the service DOWN.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

Parameter	Specifies
userName	User name SMTP server. This user name is used in the probe.
password	Password used in monitoring SMTP servers.
scriptName	The path and name of the script to execute.
dispatcherIP	The IP Address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.

To configure the built-in SMTP monitor, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring RTSP Services

Apr 08, 2013

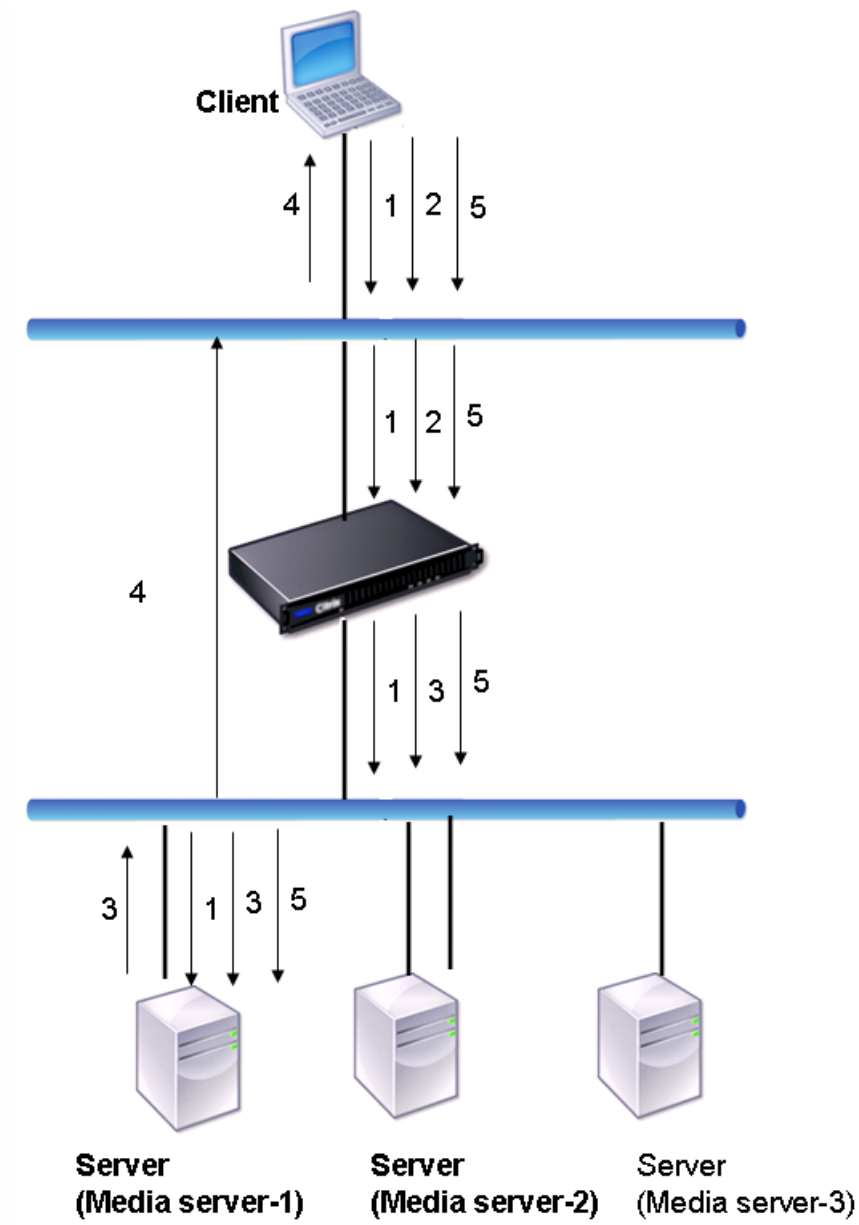
The NetScaler appliance has one built-in monitor that can be used to monitor RTSP services: the RTSP monitor. It periodically checks the RTSP service to which it is bound by opening a connection with the load balanced RTSP server. The type of connection that it opens, and the response that it expects, differs depending upon the network configuration. If the RTSP service responds as expected within the configured time period, it marks the service UP. If the service does not respond, or responds incorrectly, it marks the service DOWN.

The NetScaler appliance can be configured to load balance RTSP servers using two topologies: NAT-off and NAT-on. RTSP servers send their responses directly to the client, bypassing the appliance. The appliance must be configured to monitor RTSP services differently depending upon which topology your network uses. The appliance can be deployed either in inline or non-inline mode in both NAT-off and NAT-on mode.

In NAT-off mode, the appliance operates as a router: it receives RTSP requests from the client and routes them to the service that it selects using the configured load balancing method. If your load balanced RTSP servers are assigned publicly accessible FQDNs in DNS, the load balanced servers send their responses directly to the client, bypassing the appliance. The following figure demonstrates this configuration.

Figure 1. RTSP in NAT-off Mode

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The flow of requests and responses in this scenario is as follows:

1. The client sends a DESCRIBE request to the appliance. The appliance uses the configured load balancing method to choose a service, and routes the request to Media Server-1.
2. The client sends a SETUP request to the appliance. If the RTSP session ID is exchanged in the DESCRIBE request, the appliance, using RTSPSID persistence, routes the request to Media Server-1. If the RTSP session ID is exchanged in the SETUP request, the appliance does one of the following:

   - If the RTSP request comes on the same TCP connection, it routes the request to Media Server-1, maintaining persistence.
   - If the request arrives on a different TCP connection, it uses the configured load balancing method to choose a service, and sends the request to that service, not maintaining persistence. This means that the request may be sent to a different service.
3. Media Server-1 receives the SETUP request from the appliance, allocates resources to process the RTSP request, and sends the appropriate session ID to the client.

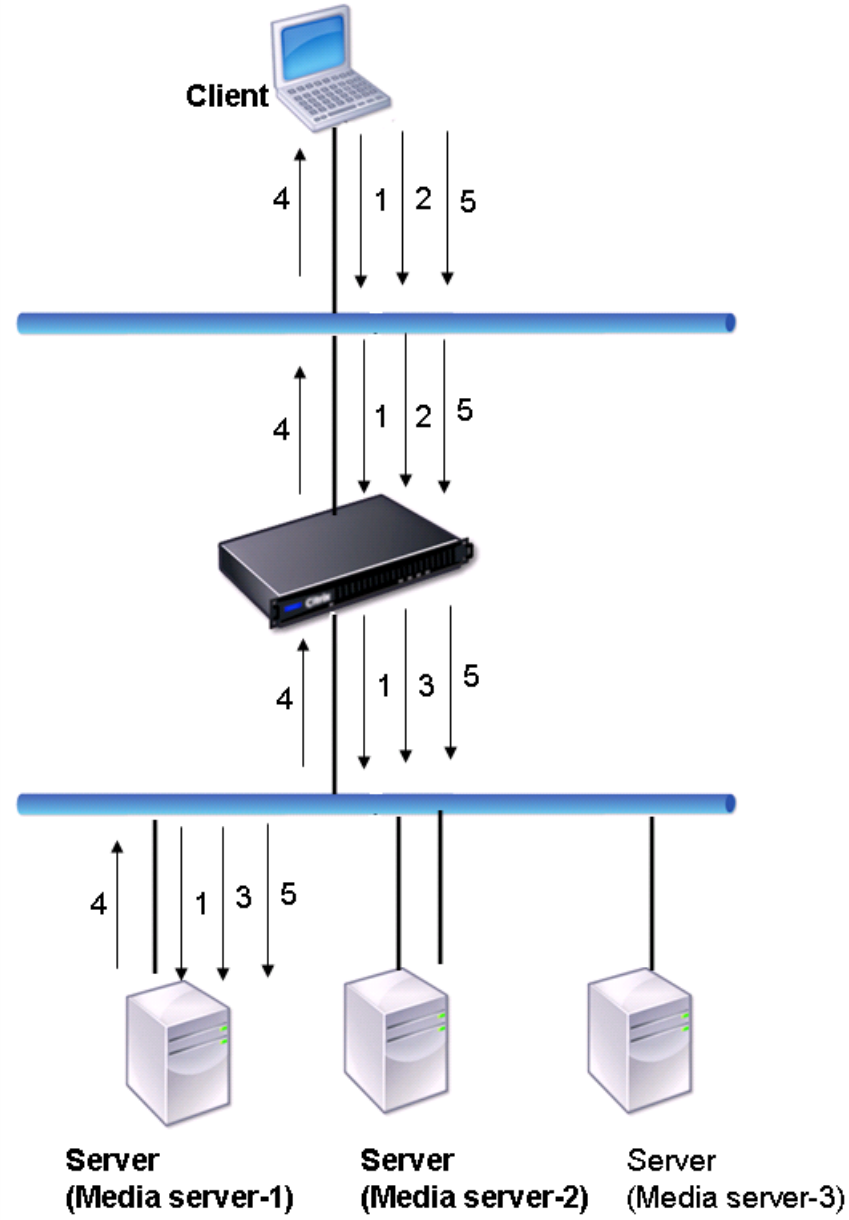
 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: The appliance does not perform NAT to identify the RTSP connection, because the RTSP connections bypass it.

4. For subsequent requests, the client then uses the session ID to identify the session and send control messages to the media server. Media Server-1 performs the requested actions, such as play, forward, or rewind.

In NAT-on mode, the appliance receives RTSP requests from the client and routes those requests to the appropriate media server using the configured load balancing method. The media server then sends its responses to the client through the appliance, as illustrated in the following diagram.

Figure 2. RTSP in NAT-on Mode



The flow of requests and responses in this scenario is as follows:

1. The client sends a DESCRIBE request to the appliance. The appliance uses the configured load balancing method to choose a service, and routes the request to Media Server-1.
2. The client sends a SETUP request to the appliance. If the RTSP session ID is exchanged in the DESCRIBE request, the appliance, using the RTSPSID persistence, routes the request to Media Server-1. If the RTSP session ID is exchanged in the SETUP request, the appliance does one of the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- If the RTSP request comes on the same TCP connection, it routes the request to Media Server-1, maintaining persistence.
- If the request arrives on a different TCP connection, it uses the configured load balancing method to choose a service, and sends the request to that service, not maintaining persistence. This means that the request may be sent to a different service.

3. Media Server-1 receives the SETUP request from the appliance, allocates resources to process the RTSP request, and sends the appropriate session ID to the client.
4. The appliance performs NAT to identify the client for RTSP data connections, and the RTSP connections pass through the appliance and are routed to the correct client.
5. For subsequent requests, the client then uses the session ID to identify the session and send control messages to the appliance. The appliance uses RTSPSID persistence to identify the appropriate service, and routes the request to Media Server-1. Media Server-1 performs the requested action, such as play, forward, or rewind.

The RTSP monitor uses the RTSP protocol to evaluate the state of the RTSP services. The RTSP monitor connects to the RTSP server and conducts a sequence of handshakes to ensure that the server is operating correctly.

Parameter	Specifies
rtspRequest	The RTSP request string that is sent to the RTSP server (for example, OPTIONS *). The default value is 07. The length of the request must not exceed 163 characters.
respCode	Set of response codes that are expected from the service.

For instructions on configuring an RTSP monitor, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring the XML Broker Services

Sep 13, 2013

The NetScaler appliance has a built-in monitor type, CITRIX-XML-SERVICE, with which you can create monitors to monitor the XML Broker services. The XML Broker services are used by Citrix XenApp. The monitor opens a connection to the service and periodically probes the XML services to which it is bound. If the server responds as expected within the configured time period, the monitor marks the service UP. If the service does not respond, or responds incorrectly, the monitor marks the service DOWN.

To configure a CITRIX-XML-SERVICE monitor, you need to specify the application name in addition to setting the standard parameters. The application name is the name of the application that has to be run to monitor the state of the XML Broker service. The default application is Notepad.

To configure monitors for XML Broker services, see "Configuring Monitors in a Load Balancing Setup."

Note

The parameter “Application Name” for Citrix-XML-Service monitor is invalid for XA and XD versions 7 and later. The probing criteria is different from XA/XD 7. However, you can still use application name for versions below XA/XD 7.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring ARP Requests

Mar 19, 2012

The NetScaler appliance has one built-in monitor that can be used to monitor ARP requests: the ARP monitor. This monitor periodically sends an ARP request to the service to which it is bound, and listens for the expected response. If it receives the expected response, it marks the service UP. If it receives no response or the wrong response, it marks the service DOWN.

ARP locates a hardware address for a load balanced server when only the network layer address is known. ARP works with IPv4 to translate IP addresses to Ethernet MAC addresses. ARP monitoring is not relevant to IPv6 networks, and is therefore not supported on those networks.

There are no special parameters for the ARP monitor.

For instructions on configuring an ARP monitor, see Configuring Monitors in a Load Balancing Setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring the XenDesktop Delivery Controller Services

Nov 26, 2013

In desktop virtualization, the NetScaler appliance can be used to load balance the Web Interface (WI) servers and the XenDesktop Delivery Controller servers deployed by Citrix XenDesktop environment. The NetScaler provides a built-in monitor, CITRIX-XD-DDC monitor, which monitors the XenDesktop Delivery Controller servers. In addition to the health check, you can also verify whether the probe is sent by a valid user of the XenDesktop Delivery Controller server.

The monitor sends a probe to the XenDesktop Delivery Controller server in the form of an XML message. If the server responds to the probe with the identity of the server farm, the probe is considered to be successful and the server's status is marked as UP. If the HTTP response does not have a success code or the identity of the server farm is not present in the response, the probe is considered to be a failure and the server's status is marked as DOWN.

The Validate Credentials option determines the probe to be sent by the monitor to the XenDesktop Delivery Controller server, that is, whether to request only the server name or to also validate the login credentials.

Note: Regardless of whether or not the user credentials (user name, password and domain) are specified on the CITRIX-XD-DDC monitor, the XenDesktop Delivery Controller server validates the user credentials only if the option to validate credentials is enabled on the monitor.
If you use the wizard for configuring the load balancing of the XenDesktop servers, the CITRIX-XD-DDC monitor is automatically created and bound to the XenDesktop Delivery Controller services.

To add an XD-DDC monitor with the validate credentials option by using the command line interface

At the command prompt, type the following commands to add an XD-DDC monitor and verify the configuration:

- add lb monitor <monitorName> <monitorType> -userName <userName> -password <password> -ddcDomain <ddc_domain_name> -validateCred YES
- show lb monitor <monitorName>

Example

```
 > add lb monitor xdddcmon Citrix-xd-ddc -userName Administrator -password E12Dc35450a1 -ddcDomain dhop -validateCred YES
 Done
> show lb monitor xdddcmon
1) Name......:xdddcmon Type......:CITRIX-XD-DDC State......: ENABLED

Standard parameters:
Interval......:..5 sec...Retries......:..3
Response timeout......:..2 sec...Down time......:..30 sec
Reverse......:..NO...Transparent......:..NO
Secure......:..NO...LRTM......:..ENABLED
Action......:..Not applicable...Deviation......:..0 sec
Destination IP......:..Bound service
Destination port......:..Bound service
Iptunnel......:..NO
TOS......:..NO...TOS ID......:..0
SNMP Alert Retries......:..0...Success Retries......:..1
Failure Retries......:..0

Special parameters:
User Name.........."Administrator"
Password..........:*****
DDC Domain.........: "dhop"
Done
```

To specify the validate credentials option on an XD-DDC monitor by using the command line interface

At the command prompt, type:

set lb monitor <monitorName> <monitorType> -userName -password -ddcDomain <ddc_domain_name> -validateCred YES

Example

```
 > set lb monitor XD_DDC_21.21.21.22_443_mn CITRIX-xd-ddc -userName Administrator -password D123S1R2A123 -ddcDomain dhop -validateCred YES
Done
```

To configure an XD-DDC monitor with the validate credentials option by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor of type Citrix-XD-DDC.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring Web Interface Services

May 07, 2015

In desktop virtualization, the NetScaler appliance can be used to load balance the Web Interface (WI) servers and Dynamic Desktop Controller (DDC) servers deployed in the Citrix XenApp and Citrix XenDesktop and environments. The NetScaler appliance has two built-in monitor types for monitoring the WI servers used in these environments.

A CITRIX-WEB-INTERFACE monitor can monitor the Web Interface services efficiently because it monitors a dynamic page at the location specified by the site path. The monitor checks for critical failures in resource availability.

To mark a service as UP, the appliance expects the following response from the server:

1. For the first GET request, 200 OK .
2. For the POST request with credentials, 302 Found with the required WIAuthID.
3. For the last GET request with session cookie, 200 OK.

Note: If a redirect URL is configured, 302 Found is expected in the first request before 200 OK.
Note: Monitor probes originate from the NetScaler IP (NSIP) address.
When you configure a CITRIX-WEB-INTERFACE monitor, specify the site path to the location of the http page that displays the data collected by the monitor. To monitor the status of the service, in the specified site path, you can view the data updated dynamically by the monitoring script auth/nocookies.aspx.

Note: End the site path with a slash (/) to indicate that the monitored resource is dynamic.
Note: When you configure the WI-EXTENDED monitor, when specifying the site path, do not enter a slash (/) at the end of the path as the software internally adds a slash at the end of the path. For example, note the following command:

```
add monitor wi CITRIX-WI-EXTENDED -sitepath "/Citrix/DesktopWeb" -username aaa -password bbb -domain ccc
```

A CITRIX-WI-EXTENDED monitor verifies the logging process with the Web Interface service. This monitor accesses the login page and passes the user name, password, domain, and site path that were specified while configuring the monitor. It verifies the validity of the login credentials, correct configuration of the monitor (for example, the site path), and the connection with the IIS server.

Note: The CITRIX-WI-EXTENDED monitor is supported only for the .NET version of the WI servers. This monitor will not work for the JSP version of the WI servers.
If you use the wizard for configuring load balancing of the XenDesktop servers, a CITRIX-WEB-INTERFACE monitor is automatically created and bound to the WI services. The wizard adds and binds a CITRIX-WEB-INTERFACE monitor by default. If you want to add and bind a CITRIX-WI-EXTENDED monitor, select the Validate Credentials check box and type the necessary data. If you do not use the wizard, add a monitor corresponding to the WI services and bind it to each WI service that you create.

- For instructions on using the wizard, see [Configuring XenDesktop for Load Balancing] or [Configuring XenApp for Load Balancing].
- For instructions on adding a CITRIX-WEB-INTERFACE monitor, see [Creating Monitors].
- For instructions on binding a monitor to a service, see [Binding Monitors to Services].
-
-
- For instructions on using the wizard, see Configuring XenDesktop for Load Balancing or Configuring XenApp for Load Balancing.
- For instructions on adding a CITRIX-WEB-INTERFACE monitor, see Creating Monitors.
- For instructions on binding a monitor to a service, see Binding Monitors to Services.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To add a WI monitor by using the command line interface

At the command prompt, type:

```
add lb monitor <monitorName> <monitorType> -sitePath <site_path> -dispatcherIP 127.0.0.1 -dispatcherPort 3013
-userName <username> -password <password> -domain <domain_name>
```

Examples

```
add lb monitor mwie CITRIX-WEB-INTERFACE -sitePath "/Citrix/XDWI/"
add lb monitor mwie CITRIX-WI-EXTENDED -sitePath "/Citrix/XDWI/"
-dispatcherIP 127.0.0.1 -dispatcherPort 3013 -userName administrator
-password d83d154575d426 -encrypted -domain wi
```

To add a WI monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and create a WI monitor of type CITRIX-WEB-INTERFACE or CITRIX-WI-EXTENDED.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring Citrix StoreFront Stores

Jul 02, 2017

You can configure a user monitor for a Citrix Storefront store. The monitor determines the state of the StoreFront store by successively probing the account service, authentication service, and discovery document (in that order). If any of those services do not respond to the probe, the monitor probe fails, and the StoreFront store is marked as DOWN. The monitor sends probes to the IP address and port of the bound service.

Note: Monitor probes originate from the NetScaler IP (NSIP) address. However, if the subnet of a StoreFront server is different from that of the appliance, then the subnet IP (SNIP) address is used.
Beginning with release 10.1 build 120.13, you can also bind a StoreFront monitor to a service group. A monitor is bound to each member of the service group and probes are sent to the IP address and port of the bound member (service). Also, because each member of a service group is now monitored by using the member's IP address, you can now use the StoreFront monitor to monitor StoreFront cluster nodes that are added as members of the service group.

In earlier releases, the StoreFront monitor tried to authenticate anonymous stores. As a result, a service could be marked as DOWN and you could not launch XenApp or XenDesktop by using the URL of the load balancing virtual server.

From build 64.x, the probe order has changed. The monitor now determines the state of the StoreFront store by successively probing the account service, the discovery document, and then the authentication service, and skips authentication for anonymous stores.

The hostname parameter for StoreFront monitors is deprecated. The secure parameter is now used to determine whether to use HTTP (the default) or HTTPS to send monitor probes.

To use HTTPS, set the secure option to Yes.

To create a StoreFront monitor by using the command line interface

At the command prompt, type the following commands to configure a StoreFront monitor and verify the configuration:

- add lb monitor <monitorName> STOREFRONT <string> -storeName <string> [-storefrontacctservice (YES | NO)] -secure (YES | NO)
- show lb monitor <monitorName>

Example

```
add lb monitor storefront_ssl STOREFRONT -storename myStore -storefrontacctservice YES -secure YES
```

To create a StoreFront monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and create a WI monitor of type STOREFRONT.

> **Note**
>
> For more information about the StoreFront monitors, see Load balancing with NetScaler.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Custom Monitors

Aug 29, 2017

In addition to built-in monitors, you can use custom monitors to check the state of your services. The NetScaler appliance provides several types of custom monitors based on scripts that are included with NetScaler operating system that can be used to determine the state of services based on the load on the service or network traffic sent to the service. These are the inline monitors, user monitors, and load monitors.

With any of these types of monitors, you can use the supplied functionality, or you can create your own scripts and use those scripts to determine the state of the service to which the monitor is bound.

This section includes the following details:

- Configuring HTTP-Inline Monitors
- Understanding User Monitors
- How to Use a User Monitor to Check Web Sites
- Understanding the Internal Dispatcher
- Configuring a Custom User Monitor
- Understanding Load Monitors
- Configuring Load Monitors
- Unbinding Metrics from a Metrics Table

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring HTTP-Inline Monitors

Oct 20, 2015

Inline monitors analyze and probe the responses from the services to which they are bound only when those services receive client requests. The inline monitor is of type HTTP-INLINE and can only be configured to work with HTTP and HTTPS services. An inline monitor determines that the service to which it is bound is UP by checking its responses to the requests that are sent to it. When no client requests are sent to the service, the inline monitor probes the service by using the configured URL.

Note: Inline monitors cannot be bound to HTTP or HTTPS Global Server Load Balancing (GSLB) remote or local services because these services represent virtual servers rather than actual load balanced Web servers.
Inline monitors have a time-out value and a retry count when probes fail. You can select any of the following action types for the NetScaler appliance to take when a failure occurs:

- **NONE.** No explicit action is taken. You can view the service and monitor, and the monitor indicates the number of current contiguous error responses and cumulative responses checked.
- **LOG.** Logs the event in ns/syslog and displays the counters.
- **DOWN.** Marks the service down and does not direct any traffic to the service. This setting breaks any persistent connections to the service. This action also logs the event and displays counters.

After the service is down, the service remains DOWN for the configured down time. After the DOWN time elapses, the inline monitor uses the configured URL to probe the service to see if it is available again. If the probe succeeds, the state of the service is changed to UP. Traffic is directed to the service, and monitoring resumes as before.

To configure inline monitors, see Configuring Monitors in a Load Balancing Setup.

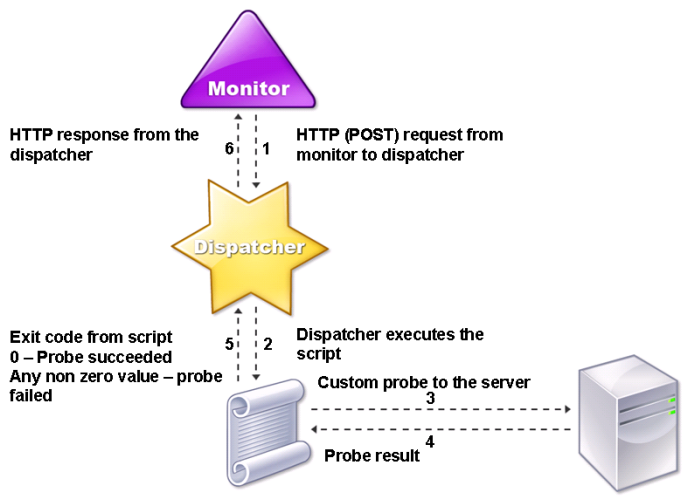
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding User Monitors

Jul 19, 2016

User monitors extend the scope of custom monitors. You can create user monitors to track the health of customized applications and protocols that the NetScaler appliance does not support. The following diagram illustrates how a user monitor works.

Figure 1. User Monitors



A user monitor requires the following components.

- **Dispatcher.** A process, on the appliance, that listens to monitoring requests. A dispatcher can be on the loopback IP address (127.0.0.1) and port 3013. Dispatchers are also known as internal dispatchers. A dispatcher can also be a web server that supports Common Gateway Interface (CGI). Such dispatchers are also known as external dispatchers. They are used for custom scripts that do not run on the FreeBSD environment, such as .NET scripts.

  Note: You can configure the monitor and the dispatcher to use HTTPS instead of HTTP by enabling the “secure” option on the monitor and configure it as an external dispatcher. However, an internal dispatcher understands only HTTP, and cannot use HTTPS.
  In a HA setup, the dispatcher runs on both the primary and secondary NetScaler appliances. The dispatcher remains inactive on the secondary appliance.

-

  At the NetScaler command line, type:

  nsumon-debug.pl <scriptname> <IP> <port> <timeout> <partitionID> [scriptarguments][is_secure]

  Important: Starting with release 10.5 build 57.x, and 11.0 script files for user monitors support IPv6 addresses and include the following changes:
  - For the following protocols, new pm files have been included for IPv6 support.
    - Radius
    - NNTP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- POP3
- SMTP

- The following sample scripts in /netscaler/monitors/ has been updated for IPv6 support:
  - nsbmradius.pl
  - nsldap.pl
  - nsnntp.pl
  - nspop3 nssf.pl
  - nssnmp.pl
  - nswi.pl
  - nstftp.pl
  - nssmtp.pl
  - nsrdp.pl
  - nsntlm-lwp.pl
  - nsftp.pl
  - nsappc.pl

  After upgrading to release 10.5 build 57.x, or 11.0, if you want to use your existing custom scripts with IPv6 services, make sure that you update the existing custom scripts with the changes provided in the updated sample scripts in /netscaler/monitors/.
  Note: The sample script nsmysql.pl does not support IPv6 address. If an IPv6 service is bound to a user monitor that uses nsmysql.pl, the probe will fail.
- The following LB monitor types have been updated to support IPv6 addresses:
  - USER
  - SMTP
  - NNTP
  - LDAP
  - SNMP
  - POP3
  - FTP_EXTENDED
  - STOREFRONT
  - APPC
  - CITRIX_WI_EXTENDED

  If you are creating a new custom script that uses one of these LB monitors types, make sure that you include IPv6 support in the custom script. Refer to the associated sample script in /netscaler/monitors/ for the changes that you

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

have to make in the custom script for IPv6 support.

Script. The script is a program that sends custom probes to the load balanced server and returns the response code to the dispatcher. The script can return any value to the dispatcher, but if a probe succeeds, the script must return a value of zero (0). The dispatcher considers any other value as probe failure.

The NetScaler appliance is bundled with sample scripts for commonly used protocols. The scripts exist in the /nsconfig/monitors directory. If you want to add a new script, add it there. If you want to customize an existing script, create a copy with a new name and modify it.
Important: Starting with release 10.1 build 122.17, the script files for user monitors are in a new location.
If you upgrade an MPX or VPX virtual appliance to release 10.1 build 122.17 or later, the changes are as follows:

- A new directory named conflicts is created in /nsconfig/monitors/ and all the built-in scripts of the previous builds are moved to this directory.
- All new built-in scripts are available in the /netscaler/monitors/ directory. All custom scripts are available in the /nsconfig/monitors/ directory.
- You must save a new custom script in the /nsconfig/monitors/ directory.
- After the upgrade is completed, if a custom script is created and saved in the /nsconfig/monitors/ directory, with the same name as that of a built-in script, the script in the /netscaler/monitors/ directory takes priority. That is, the custom script does not run.

If you provision a virtual appliance with release 10.1 build 122.17 or later, the changes are as follows:

- All built-in scripts are available in the /netscaler/monitors/ directory.
- The /nsconfig/monitors/ directory is empty.
- If you create a new custom script, you must save it in the /nsconfig/monitors/ directory.

For the scripts to function correctly:

- The maximum number of characters in the name of the script must not exceed 63.
- The maximum number of script arguments that can be provided to a script must not exceed 512.
- The maximum number of characters that can be provided in the parameter script arguments must not exceed 639.

To debug the script, you must run it by using the nsumon-debug.pl script from the NetScaler command line. You use the script name (with its arguments), IP address, and the port as the arguments of the nsumon-debug.pl script. Users must use the script name, IP address, port, time-out, and the script arguments for the nsumon-debug.pl script.

To track the status of the server, the monitor sends an HTTP POST request to the configured dispatcher. This POST request contains the IP address and port of the server, and the script that must be executed. The dispatcher executes the script as a child process, with user-defined parameters (if any). Then, the script sends a probe to the server. The script sends the status of the probe (response code) to the dispatcher. The dispatcher converts the response code to an HTTP response and sends it to the monitor. Based on the HTTP response, the monitor marks the service as up or down.

The appliance logs the error messages to the /var/nslog/nsumond.log file when user monitor probes fail. The following table lists the user monitors and the possible reasons for failure.

User monitor type	Probe failure reasons
SMTP	Monitor fails to establish a connection to the server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

User monitor type	Probe failure reasons
NNTP	Monitor fails to establish a connection to the server.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Monitor fails to find the NNTP group.
LDAP	Monitor fails to establish a connection to the server.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Monitor fails to bind to the LDAP server.
	Monitor fails to locate an entry for the target entity in the LDAP server.
FTP	The connection to the server times out.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Logon fails.
	Monitor fails to find the file on the server.
POP3	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Logon fails.
POP3	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Logon fails.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

User monitor type	Probe failure reasons
	Preparation of SQL query fails.
	Execution of SQL query fails.
SNMP	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Logon fails.
	Monitor fails to create the SNMP session.
	Monitor fails to find the object identifier.
	The monitor threshold value setting is greater than or equal to the actual threshold of the monitor.
RDP (Windows Terminal Server)	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Monitor fails to create a socket.
	Mismatch in versions.
	Monitor fails to confirm connection.

You can view the log file from the NetScaler command line by using the following commands, which open a BSD shell, display the log file on the screen, and then close the BSD shell and return you to the NetScaler command prompt:

```
> shell
root@ns# cat /var/nslog/nsumond.log
root@ns# exit
>
```

User monitors also have a time-out value and a retry count for probe failures. You can use user monitors with non-user monitors. During high CPU utilization, a non-user monitor enables faster detection of a server failure.

If the user monitor probe times out during high CPU usage, the state of the service remains unchanged.

Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For scriptable monitors, the response timeout must be configured to a value equal to expected timeout + 1 second.

For example, if you expect the timeout to be 4 seconds, configure the response timeout as 5 seconds.

Example command:

add lb monitor <name> USER –scriptname <script-name> -resptimeout 5 seconds

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How to Use a User Monitor to Check Web Sites

Nov 07, 2017

You can configure a user monitor to check for specific Web site problems that are reported by HTTP servers using specific HTTP codes. The following table lists the HTTP response codes that this user monitor expects.

HTTP response code	Meaning
200 - success	Probe success.
503 - service unavailable	Probe failure.
404 - not found	Script not found or cannot execute.
500 - Internal server error	Internal error/resource constraints in dispatcher (out of memory, too many connections, unexpected system error, or too many processes). The service is not marked DOWN.
400 - bad request	Error parsing HTTP request.
502 - bad gateway	Error decoding script's response.

You configure the user monitor for HTTP by using the following parameters.

Parameter	Specifies
scriptName	The path and name of the script to execute.
scriptArgs	The strings that are added in the POST data. They are copied to the request verbatim.
dispatcherIP	The IP address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.
localfileName	The name of a monitor script file on the local system.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter	Specifies
destPath	A particular location on the NetScaler appliance where the uploaded local file is stored.

To create a user monitor to monitor HTTP, see Configuring Monitors in a Load Balancing Setup.

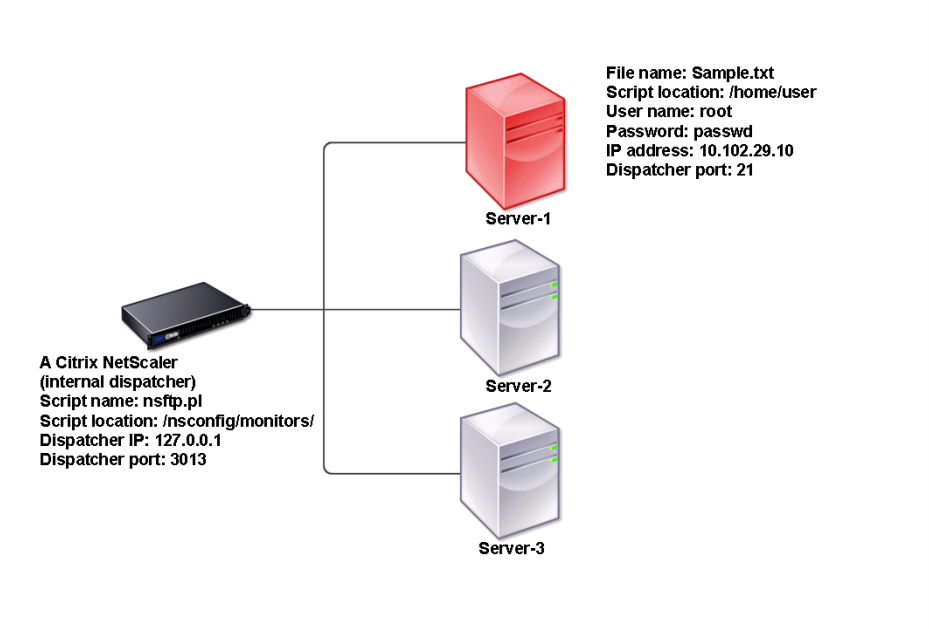
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Understanding the Internal Dispatcher

Mar 19, 2012

You can use a custom user monitor with the internal dispatcher. Consider a case where you need to track the health of a server based on the presence of a file on the server. The following diagram illustrates this scenario.

Figure 1. Using a User Monitor with the Internal Dispatcher

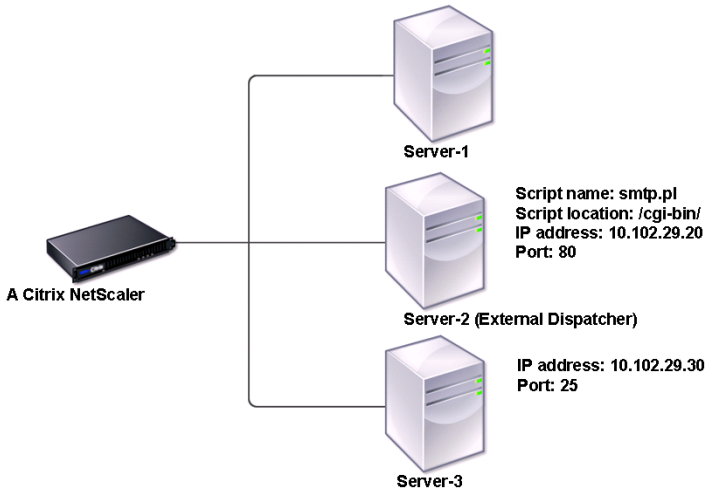


A possible solution is to use a Perl script that initiates an FTP session with the server and checks for the presence of the file. You can then create a user monitor that uses the Perl script. The NetScaler includes such a Perl script (nsftp.pl), in the /nsconfig/monitors/ directory.

You can use a user monitor with an external dispatcher. Consider a case where you must track the health of a server based on the state of an SMTP service on another server. This scenario is illustrated in the following diagram.

Figure 2. Using a User Monitor with an External Dispatcher

© 1999-2017 Citrix Systems, Inc. All rights reserved.



A possible solution would be to create a Perl script that checks the state of the SMTP service on the server. You can then create a user monitor that uses the Perl script.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Custom User Monitor

Aug 24, 2016

To configure a custom user monitor, you must first write the script that performs the action that the monitor will use to check the service that is bound to it, and upload the script to the /nsconfig/monitors directory on the NetScaler appliance. Then you create the monitor on the appliance, as described below.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

To configure a user monitor by using the command line interface

At the command prompt, type:

add lb monitor <monitorName> USER -scriptname <NameOfScript> –scriptargs <Arguments>

Example

add monitor Monitor-User-1 USER -scriptname nsftp.pl –scriptargs "file=/home/user/sample.txt;user=root;password=passwd"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

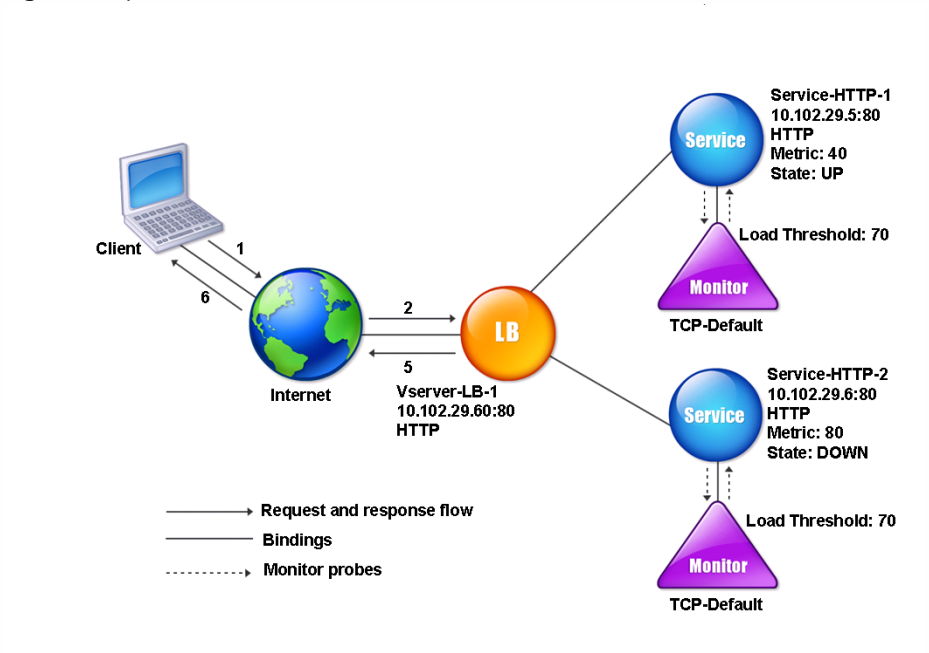
Understanding Load Monitors

Nov 07, 2017

Load monitors use SNMP polled OIDs to calculate load. The load monitor uses the IP address of the service to which it is bound (the destination IP address) for polling. It sends an SNMP query to the service, specifying the OID for a metric. The metrics can be CPU, memory, or number of server connections. The server responds to the query with a metric value. The metric value in the response is compared with the threshold value. The NetScaler appliance considers the service for load balancing only if the metric is less than the threshold value. The service with the lowest load value is considered first.

The following diagram illustrates a load monitor configured for the services described in the basic load balancing setup discussed in Setting Up Basic Load Balancing.

Figure 1. Operation of Load Monitors



Note: The load monitor does not determine the state of the service. It only enables the appliance to consider the service for load balancing.

After you configure the load monitor, you must then configure the metrics that the monitor will use. For load assessment, the load monitor considers server parameters known as metrics, which are defined within the metric tables in the appliance configuration. Metric tables can be of two types:

- **Local.** By default, this table exists in the appliance. It consists of four metrics: connections, packets, response time, and bandwidth. The appliance specifies these metrics for a service, and SNMP queries are not originated for these services. These metrics cannot be changed.
- **Custom.** A user-defined table. Each metric is associated with an OID.

By default, the appliance generates the following tables:

- NetScaler
- RADWARE
- CISCO-CSS
- LOCAL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- FOUNDRY
- ALTEON

You can either add the appliance-generated metric tables, or you can add tables of your own choosing, as shown in the following table. The values in the metric table are provided only as examples. In an actual scenario, consider the real values for the metrics.

Metric name	OIDs	Weight	Threshold
CPU	1.2.3.4	2	70
Memory	4.5.6.7	3	80
Connections	5.6.7.8	4	90

To calculate the load for one or more metrics, you assign a weight to each metric. The default weight is 1. The weight represents the priority given to each metric. If the weight is high, the priority is high. The appliance chooses a service based on the SOURCEIPDESTIP hash algorithm.

You can also set the threshold value for each metric. The threshold value enables the appliance to select a service for load balancing if the metric value for the service is less than the threshold value. The threshold value also determines the load on each service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Load Monitors

Nov 07, 2017

To configure a load monitor, first create the load monitor. For instructions on creating a monitor, see Creating Monitors. Next, select or create the metric table to define a set of metrics that determine the state of the server, and (if you create a metric table) bind each metric to the metric table.

To create a metric table by using the command line interface

At the command prompt, type the following commands:

- add lb metricTable <metricTableName>
- bind lb metricTable <metricTableName> <metric> <SNMPOID>

Example

```
add metricTable Table-Custom-1

bind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5 11
```

To create a metric table and bind metrics to it by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Metric Tables and create a metric table.
2. To bind metrics, click Bind and specify a metric and an SNMP OID.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Unbinding Metrics from a Metrics Table

Nov 12, 2013

You can unbind metrics from a metrics table if the metrics need to be changed, or if you want to remove the metrics table entirely.

To unbind metrics from a metric table by using the command line interface

At the command prompt, type:

unbind lb metricTable <metricTable> <metric>

Example

```
unbind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5
```

To unbind metrics from a metric table by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Metric Tables.
2. Open a metric table, select a metric, and click Delete.

You can view the detail of all configured metric tables, such as name and type, to determine whether the metric table is internal or created and configured.

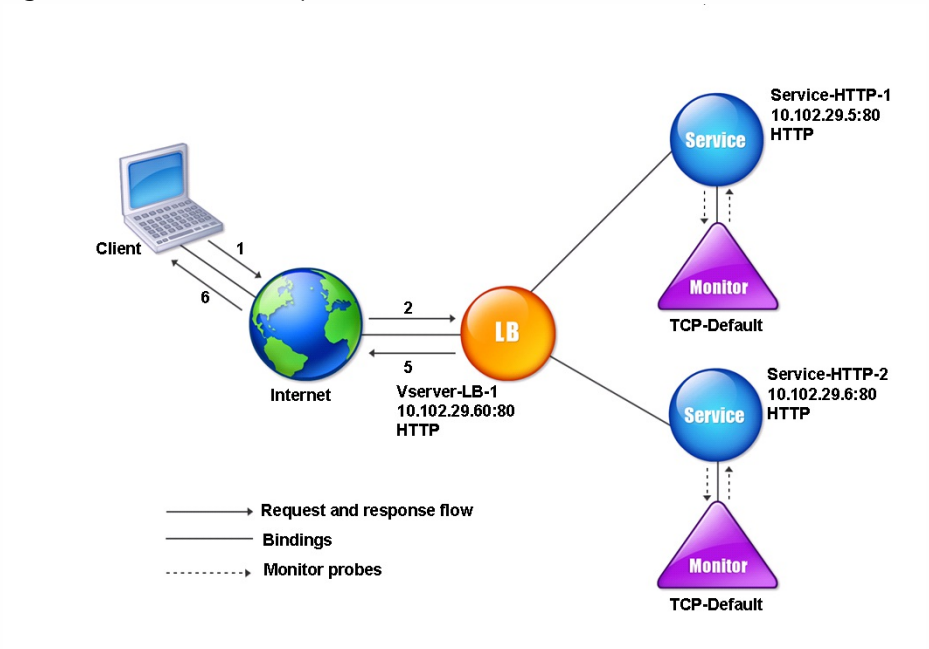
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Monitors in a Load Balancing Setup

Jul 05, 2017

To configure monitors on a Web site, you first decide whether to use a built-in monitor or create your own monitor. If you create a monitor, you can choose between creating a monitor based on a built-in monitor, or creating a custom monitor that uses a script that you write to monitor the service. For more information about creating custom monitors, see Custom Monitors. Once you have chosen or created a monitor, you then bind it to the appropriate service. The following conceptual diagram illustrates a basic load balancing setup with monitors.

Figure 1. How Monitors Operate



As shown above, each service has a monitor bound to it. The monitor probes the load balanced server via its service. As long as the load balanced server responds to the probes, the monitor marks it UP. If the load balanced server should fail to respond to the designated number of probes within the designated time period, the monitor marks it DOWN.

This section includes the following details:

- Creating Monitors
- Binding Monitors to Services
- Modifying Monitors
- Enabling and Disabling Monitors
- Unbinding Monitors
- Removing Monitors
- Viewing Monitors
- Closing Monitor Connections
- Ignoring the Upper Limit on Client Connections for Monitor Probes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating Monitors

Nov 12, 2013

The NetScaler appliance provides a set of built-in monitors. It also allows you to create custom monitors, either based on the built-in monitors or from scratch.

To create a monitor by using the command line interface

At the command prompt, type:

add lb monitor <monitorName> <monitorType> [<interval>]

Example

```
add lb mon monitor-HTTP-1 HTTP

add lb mon monitor-HTTP-2 TCP 2
```

To create a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding Monitors to Services

Nov 12, 2013

After creating a monitor, you bind it to a service. You can bind one or multiple monitors to a service. If you bind one monitor to a service, that monitor determines whether the service is marked UP or DOWN. If you bind multiple monitors to a service, the NetScaler appliance checks all monitors bound to that service using a calculation that you control, and marks the service UP or DOWN depending on the results.

Note: The destination IP address of a monitor probe can be different than the server IP address and port.

To bind a monitor to a service by using the command line interface

At the command prompt, type:

bind lb monitor <monitorName> <ServiceName>

Example

bind mon monitor-HTTP-1 Service-HTTP-1

To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and add a monitor.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Modifying Monitors

Nov 12, 2013

You can modify the settings for any monitor that you created.

Note: Two sets of parameters apply to monitors: those that apply to all monitors, regardless of type, and those that are specific to a monitor type. For information on parameters for a specific monitor type, see the description for that type of monitor.

To modify an existing monitor by using the command line interface

At the command prompt, type:

set lb monitor <monitorName> <type> -interval <interval> -resptimeout <resptimeout>

Example

```
set mon monitor-HTTP-1 HTTP -interval 50 milli
-resptimeout 20 milli
```

To modify an existing monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and open a monitor to modify.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling and Disabling Monitors

Nov 12, 2013

By default, monitors bound to services and service groups are enabled. When you enable a monitor, the monitor begins probing the services to which it is bound. If you disable a monitor bound to a service, the state the service is determined using the other monitors bound to the service. If the service is bound to only one monitor, and if you disable the monitor, the state of the service is determined using the default monitor.

To enable a monitor by using the command line interface

At the command prompt, type:

enable lb monitor <monitorName>

Example

enable lb mon monitor-HTTP-1

To enable a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Select a monitor, and from the Action list, select Enable or Disable.

To disable a monitor by using the command line interface

At the command prompt, type:

disable lb monitor <monitorName>

Example

disable lb mon monitor-HTTP-1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Unbinding Monitors

Nov 12, 2013

You can unbind monitors from a service and service group. When you unbind a monitor from the service group, the monitors are unbound from the individual services that constitute the service group. When you unbind a monitor from a service or a service group, the monitor does not probe the service or the service group.

Note: When you unbind all user-configured monitors from a service or a service group, the default monitor is bound to the service and the service group. The default monitors then probes the service or the service groups.

To unbind a monitor from a service by using the command line interface

At the command prompt, type:

unbind lb monitor <monitorName>

Example

unbind mon monitor-HTTP-1 Service-HTTP-1

To unbind a monitor from a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service to modify.
2. Click in the Monitors section, select a monitor, and click Unbind.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Removing Monitors

Nov 12, 2013

After you unbind a monitor that you created from its service, you can remove that monitor from the NetScaler configuration. (If a monitor is bound to a service, it cannot be removed.)

Note: When you remove monitors bound to a service, the default monitor is bound to the service. You cannot remove default monitors.

To remove a monitor by using the command line interface

At the command prompt, type:

rm lb monitor <monitorName> <type>

Example

rm lb monitor monitor-HTTP-1 HTTP

To remove a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Select a monitor, and click Delete.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing Monitors

Nov 12, 2013

You can view the services and service groups that are bound to a monitor. You can verify the settings of a monitor to troubleshoot your NetScaler configuration. The following procedure describes the steps to view the bindings of a monitor to the services and service groups.

To view monitor bindings by using the command line interface

At the command prompt, type:

show lb monbindings <MonitorName>

Example

show lb monbindings monitor-HTTP-1

To view monitor bindings by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Select a monitor, and in the Action list, click Show Bindings.

To view monitors by using the command line interface

At the command prompt, type:

show lb monitor <monitorName>

Example

show lb mon monitor-HTTP-1

To view monitors by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors. The details of the available monitors appear in the Monitors pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Closing Monitor Connections

Aug 29, 2013

The NetScaler appliance sends probes to the services through the monitors bound to the services. By default, the monitor on the NetScaler and the physical server follow the complete handshake procedure even for monitor probes. However, this procedure adds overhead to the monitoring process and may not be always necessary.

For the TCP monitors, you can configure the NetScaler to close a monitor-probe connection after receiving SYN-ACK from the service. To do so, set the value of the monitorConnectionClose parameter to RESET. If you want the monitor-probe connection to go through the complete procedure, set the value to FIN.

Note: The monitorConnectionClose setting is applicable to all the monitors bound to all the services configured on the NetScaler appliance.

To configure monitor-connection closure by using the command line interface

At the command prompt, type:

```
set lb parameter -monitorConnectionClose <monitor_conn_close_option>
```

Example

```
set lb parameter -monitorConnectionClose RESET
```

To configure monitor-connection closure by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing Parameters.
2. Select FIN or Reset.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Ignoring the Upper Limit on Client Connections for Monitor Probes

Aug 29, 2013

Depending on considerations such as the capacity of a physical server, you can specify a limit on the maximum number of client connections made to any service. If you have set such a limit on a service, the NetScaler appliance stops sending requests to the service when the threshold is reached and resumes sending connections to the service after the number of existing connections falls to within the limits. You can configure the NetScaler to skip this check when it sends monitor-probe connections to a service.

Note: You cannot skip the maximum-client-connections check for an individual service. If you specify this option, it applies to all the monitors bound to all the services configured on the NetScaler appliance.

To set the Skip MaxClients for Monitor Connections option by using the command line interface

At the command prompt, type:

set lb parameter -monitorSkipMaxClient (ENABLED | DISABLED)

Example

```
set lb parameter -monitorSkipMaxClient enabled
```

To set the Skip MaxClients for Monitor Connections option by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing Parameters.
2. Select Skip MaxClients for Monitoring Connections.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing a Large Scale Deployment

Mar 19, 2012

The NetScaler appliance contains several features that are helpful when you are configuring a large load balancing deployment. Instead of configuring virtual servers and services individually, you can create groups of virtual servers and services. You can also create a range of virtual servers and services, and you can translate or mask virtual server and service IP addresses.

You can set persistence for a group of virtual servers. You can bind monitors to a group of services. Creating a range of virtual servers and services of identical type allows you to set up and configure those servers in a single procedure, which significantly shortens the time required to configure those virtual servers and services.

By translating or masking IP addresses, you can take down virtual servers and services, and make changes to your infrastructure, without extensive reconfiguration of your service and virtual server definitions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Ranges of Virtual Servers and Services

Mar 19, 2012

When you configure load balancing, you can create ranges of virtual servers and services, eliminating the need to configure virtual servers and services individually. For example, you can use a single procedure to create three virtual servers with three corresponding IP addresses. When more than one argument uses a range, all of the ranges must be of the same size.

The following are the types of ranges you can specify when adding services and virtual servers to your configuration:

- **Numeric ranges.** Instead of typing a single number, you can specify a range of consecutive numbers.

  For example, you can create a range of virtual servers by specifying a starting IP address, such as 10.102.29.30, and then typing a value for the last byte that indicates the range, such as 34. In this example, five virtual servers will be created with IP addresses that range between 10.102.29.30 and 10.102.29.34.

  Note: The IP addresses of the virtual servers and services must be consecutive.
- **Alphabetic ranges.** Instead of typing a literal letter, you can substitute a range for any single letter, for example, [C-G]. This results in all letters in the range being included, in this case C, D, E, F, and G.

  For example, if you have three virtual servers named Vserver-x, Vserver-y, and Vserver-z, instead of configuring them separately, you can type vserver [x-z] to configure them all.

Creating a Range of Virtual Servers

Updated: 2013-11-12

You create a range of virtual servers as described below.

To create range of virtual servers by using the command line interface

At the command prompt, type one of the following commands:

- add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<port>]
- add lb vserver <name>@**[**<rangeValue>**]**> <protocol> <IPAddress[<rangeValue>]> [<port>]

Example

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
```

OR

```
> add lb vserver vserver[P-R] http 10.102.29.[26-28] 80
vserver "vserverP" added
vserver "vserverQ" added
vserver "vserverR" added
 Done
```

To create range of virtual servers by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Add a virtual server, and specify a range.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating a Range of Services

Updated: 2013-11-12

You create a range of services as described below. If you specify a range for the service name, specify a range for the IP address too.

To create range of services by using the command line interface

At the command prompt, type the command:

add service <name>@ <IP>@ <protocol> <port>

Example

```
> add service serv[1-3] 10.102.29.[102-104] http 80
service "serv1" added
service "serv2" added
service "serv3" added
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Service Groups

Jan 04, 2016

Configuring a service group enables you to manage a group of services as easily as a single service. For example, if you enable or disable any option, such as compression, health monitoring or graceful shutdown, for a service group, the option gets enabled for all the members of the service group.

After creating a service group, you can bind it to a virtual server, and you can add services to the group. You can also bind monitors to service groups.

The members of a service group can be identified by IP address or server name.

Using domain-name based service (DBS) group members is advantageous because you need not reconfigure the member on the NetScaler appliance if the IP address of the member changes. The appliance automatically senses such changes through the configured name server. This feature is particularly useful in cloud scenarios, where the service provider can change a physical server or change the IP address for a service. If you specify a DBS group member, the NetScaler learns the IP address dynamically.

You can bind both IP-based and DBS members to the same service group.

Note: If you use DBS service group members, make sure that either a name server is specified or a DNS server is configured on the NetScaler. A domain name will be resolved into an IP address only if the corresponding address record is present on the NetScaler or the name server.

Creating Service Groups

You can configure up to 8192 service groups on the NetScaler appliance.

To create a service group by using the command line

At the command prompt, type:

add servicegroup <ServiceGroupName> <Protocol>

Example

```
add servicegroup Service-Group-1 HTTP
```

To create a service group by using the configuration utility

Navigate to Traffic Management > Load Balancing > Service Groups, and add a service group.

Binding a Service Group to a Virtual Server

When you bind a service group to a virtual server, the member services are bound to the virtual server.

To bind a service group to a virtual server by using the command line interface

At the command prompt, type:

bind lb vserver <name>@ <serviceGroupName>

Example

```
bind lb vserver Vserver-LB-1 Service-Group-1
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To bind a service group to a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Service Groups.

Binding a Member to a Service Group

Adding services to a service group enables the service group to manage the servers. You can add the servers to a service group by specifying the IP addresses or the names of the servers.

In the configuration utility, if you want to add a domain-name based service group member, select Server Based.

With this option, you can add any server that has been assigned a name, regardless of whether the name is an IP address or a user-assigned name.

To add members to a service group by using the command line interface

To configure a service group, at the command prompt, type:

bind servicegroup <serviceGroupName> (<IP>@ | <serverName>) <port>

Examples

```
bind servicegroup Service-Group-1 10.102.29.30 80

bind servicegroup Service-Group-2 1000:0000:0000:0000:0005:0600:700a:888b 80

bind servicegroup CitrixEdu s1.citrite.net
```

To add members to a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groupsand open a service group.
2. Click in the Service Group section, and do one of the following:
   - To add a new IP based service group member, select IP Based.
   - To add a server-name based service group member, select Server Based.

   If you want to add a domain-name based service group member, select **Server Based**. With this option, you can add any server that has been assigned a name, regardless of whether the name is an IP address or a user-assigned name.
3. If adding a new IP based member, in the IP Address text box, type the IP address. If the IP address uses IPv6 format, select the IPv6 check box and then enter the address in the IP Address text box.

   Note: You can add a range of IP addresses. The IP addresses in the range must be consecutive. Specify the range by entering the starting IP address in the IP Address text box (for example, 10.102.29.30). Specify the end byte of the IP address range in the text box under Range (for example, 35). In the Port text box type the port (for example, 80), and then click Add.
4. Click Create.

Binding a Monitor to a Service Group

When you create a service group, the default monitor of the type appropriate for the group is automatically bound to it. Monitors periodically probe the servers in the service group to which they are bound and update the state of the service groups.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

You can bind a different monitor of your own choice to the service group.

To bind a monitor to a service group by using the command line interface

At the command prompt, type:

bind serviceGroup <serviceGroupName> -monitorName <string> -monState (ENABLED | DISABLED)

Example

```
bind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
```

To a bind monitor to a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Open a service group and, in Advanced Settings, click Monitors.

Retaining the Original State of a Service Group Member after Disabling and Enabling a Virtual Server

From build 64.x, a new global option, –retainDisableServer, enables you to retain a service-group member's state when a server is disabled and reenabled.

Previously, a member's state would change from DISABLED to ENABLED under the following set of conditions:

- Two applications are deployed on the same port on a virtual server.
- Two service groups with a common member are bound to this virtual server, and the common member is enabled in one group and disabled in the other.
- The server is disabled and then reenabled.

Under these conditions, disabling the server disables all the service group members, and reenabling the server enables all the members, by default, regardless of their earlier states. To bring the members back to the original states, you must manually disable those member(s) in the service group. This is a cumbersome task and prone to errors.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Service Groups

Jun 04, 2015

You can change the settings of the services in a service group, and you can perform tasks such as enabling, disabling, and removing service groups. You can also unbind members from a service group.

To manage service groups, see the following sections:

- [Modifying a Service Group]
- [Removing a Service Group]
- [Unbinding a Member from a Service Group]
- [Unbinding a Service Group from a Virtual Server]
- [Unbinding Monitors from Service Groups]
- [Enabling or Disabling a Service Group]
- [Viewing the Properties of a Service Group]
- [Viewing Service Group Statistics]
- [Load Balancing Virtual Servers Bound to a Service Group]

Modifying a Service Group

Updated: 2013-11-12

You can modify attributes of service group members. You can set several attributes of the service group, such as maximum client, SureConnect, and compression. The attributes are set on the individual servers in the service group. You cannot set parameters on the service group such as transport information (IP address and port), weight, and server ID.

Note: A parameter you set for a service group is applied to the member servers in the group, not to individual services.

To modify a service group by using the command line interface

At the command prompt, type the following command with one or more of the optional parameters:

set servicegroup <serviceGroupName> [-type <type>] [-maxClient <maxClient>] [-maxReq <maxReq>] [-cacheable (YES | NO)] [-cip (ENABLED | DISABLED)] [-cipHeader <cipHeader>] [-usip (YES | NO)] [-sc (ON | OFF)] [-sp (ON | OFF)] [-cltTimeout <cltTimeout>] [-svrTimeout <svrTimeout>] [-cka (YES | NO)] [-TCPB (YES | NO)] [-CMP (**YES** | **NO**)] [-maxBandwidth <maxBandwidth>] [-maxThreshold <maxThreshold>] [-state (ENABLED | DISABLED)] [-downStateFlush (ENABLED | DISABLED)

Example

```
set servicegroup Service-Group-1 -type TRANSPARENT

set servicegroup Service-Group-1 -maxClient 4096

set servicegroup Service-Group-1 -maxReq 16384

set servicegroup Service-Group-1 -cacheable YES
```

To modify a service group by using the configuration utility

Navigate to Traffic Management > Load Balancing > Service Groups, and open the service group to modify.

Removing a Service Group

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2013-09-04

When you remove a service group, the servers bound to the group retain their individual settings and continue to exist on the NetScaler.

To remove a service group by using the command line interface

At the command prompt, type:

rm servicegroup <ServiceGroupName>
Example

rm servicegroup Service-Group-1

To remove a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Select a service group, and click Delete.

Unbinding a Member from a Service Group

Updated: 2013-11-12

When you unbind a member from the service group, the attributes set on the service group will no longer apply to the member that you unbound. The member services retains its individual settings, however, and continues to exist on the NetScaler.

To unbind members from a service group by using the command line interface

At the command prompt, type:

unbind servicegroup <serviceGroupName> <IP>@ [<port>]
Example

unbind servicegroup Service-Group-1 10.102.29.30 80

To unbind members from a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Open a service group, and click in the Service Group Members section.
3. Select a service group member, and click Unbind.

Unbinding a Service Group from a Virtual Server

Updated: 2013-11-12

When you unbind a service group from a virtual server, the member services are unbound from the virtual server and continue to exist on the NetScaler appliance.

To unbind a service group from a virtual server by using the command line interface

At the command prompt, type:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

unbind lb vserver <name>@ <ServiceGroupName>

Example

```
unbind lb vserver Vserver-LB-1 Service-Group-1
```

To unbind a service group from a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open the virtual server, and click in the Service Group section.
3. Select the service group, and click Unbind.

Unbinding Monitors from Service Groups

Updated: 2013-12-10

When you unbind a monitor from a service group, the monitor that you unbound no longer monitors the individual services that constitute the group.

To unbind a monitor from a service group using the command line interface

At the command prompt, type:

unbind serviceGroup <serviceGroupName> -monitorName <string>

Example

```
unbind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
```

To unbind a monitor from a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Open a service group, and click in the Monitors section.
3. Select a monitor, and click Unbind.

Enabling or Disabling a Service Group

Updated: 2013-09-04

When you enable a service group and the servers, the services belonging to the service group are enabled. Similarly, when a service belonging to a service group is enabled, the service group and the service are enabled. By default, service groups are enabled.

After disabling an enabled service, you can view the service using the configuration utility or the command line to see the amount of time that remains before the service goes DOWN.

To disable a service group by using the command line interface

At the command prompt, type:

disable servicegroup <ServiceGroupName>

Example

```
disable servicegroup Service-Group-1
```

To disable a service group by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Select a service group, and in the Action list, click Disable.

To enable a service group by using the command line interface

At the command prompt, type:

enable servicegroup <ServiceGroupName>

Example

```
enable servicegroup Service-Group-1
```

To enable a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Select a service group, and in the Action list, click Enable.

Viewing the Properties of a Service Group

Updated: 2013-09-04

You can view the following settings of the configured service groups: name, IP address, state, protocol, maximum client connections, maximum requests per connection, maximum bandwidth, and monitor threshold. Viewing the details of the configuration can be helpful for troubleshooting your configuration.

To view the properties of a service group by using the command line interface

At the command prompt, type one of the following commands to display the group properties or the properties and the group members:

- show servicegroup <ServiceGroupName>
- show servicegroup <ServiceGroupName> -includemembers

Example

```
show servicegroup Service-Group-1
```

To view the properties of a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Click the arrow next to the service group.

Viewing Service Group Statistics

Updated: 2013-09-04

You can view service-group statistical data, such as rate of requests, responses, request bytes, and response bytes. The NetScaler appliance uses the statistics of a service group, such as these, to balance the load on the services.

To view the statistics of a service group by using the command line interface

At the command prompt, type:

stat servicegroup <ServiceGroupName>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

stat servicegroup Service-Group-1

To view the statistics of a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Select a service group, and click Statistics.

Load Balancing Virtual Servers Bound to a Service Group

Updated: 2013-09-04

In large-scale deployments, the same service group can be bound to multiple load balancing virtual servers. In such a case, instead of viewing each virtual server to see the service group it is bound to, you can view a list of all the load balancing virtual servers bound to a service group. You can view the following details of each virtual server:

- Name
- State
- IP address
- Port

To display the virtual servers bound to a service group by using the command line interface

At the command prompt, type the following command to display the virtual servers bound to a service group:

show servicegroupbindings <serviceGroupName>
Example

```
> show servicegroupbindings SVCGRPDTLS
SVCGRPDTLS - State :ENABLED
1) Test-pers (10.10.10.3:80) - State : DOWN
2) BRVSERV (10.10.1.1:80) - State : DOWN
3) OneMore (10.102.29.136:80) - State : DOWN
4) LBVIP1 (10.102.29.66:80) - State : UP
Done
>
```

To display the virtual servers bound to a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Select a service group, and in the Action list, click Show Bindings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Automatic Domain Based Service Group Scaling

Aug 06, 2017

A domain based service group consists of members whose IP addresses are obtained by resolving the domain names of servers that are bound to the service group. The domain names are resolved by a name server whose details you configure on the appliance. A domain based service group can also include IP-address based members.

The process of name resolution for a domain based server might return more than one IP address. The number of IP addresses in the DNS response is determined by the number of address (A) records configured for the domain name, on the name server. Even if the name resolution process returns multiple IP addresses, only one IP address is bound to the service group. To scale up or scale down a service group, you need to manually bind and unbind additional domain based servers to and from the service group, respectively.

However, you can configure a domain based service group to scale automatically on the basis of the complete set of IP addresses returned by a DNS name server for a domain based server. To configure automatic scaling, when binding a domain based server to a service group, enable the automatic scaling option. Following are the steps for configuring a domain based service group that scales automatically:

- Add a name server for resolving domain names. For more information about configuring a name server on the appliance, see Adding a Name Server.
- Add a domain based server. For information about adding a domain based server, see Adding a Server.
- Add a service group and associate the domain based server to the service group, with the autoscale option set to DNS. For information about adding a service group, see Configuring Service Groups.

When a domain based server is bound to a service group and the automatic scaling option is set on the binding, a UDP monitor and a TCP monitor are automatically created and bound to the domain based server. The two monitors function as resolvers. The TCP monitor is disabled by default, and the appliance uses the UDP monitor to send DNS queries to the name server to resolve the domain name. If the DNS response is truncated (has the TC flag set to 1), the appliance falls back to TCP and uses the TCP monitor to send the DNS queries over TCP. Thereafter, the appliance continues to use only the TCP monitor.

The DNS response from the name server might contain multiple IP addresses for the domain name. With the automatic scaling option set, the appliance polls each of the IP addresses by using the default monitor, and then includes in the service group only those IP addresses that are up and available. After the IP address records expire, as defined by their time-to-live (TTL) values, the UDP monitor (or the TCP monitor, if the appliance has fallen back to using the TCP monitor) queries the name server for domain resolution and includes any new IP addresses in the service group. If an IP address that is part of the service group is not present in the DNS response, the appliance removes that address from the service group after gracefully closing existing connections to the group member, a process during which it does not allow any new connections to be established with the member. If a domain name that resolved successfully in the past results in an NXDOMAIN response, all the service group members associated with that domain are removed.

Static (IP-address based) members and dynamically scaling domain based members can coexist in a service group. You can also bind members with different domain names to a service group with the automatic scaling option set. However, each domain name associated with a service group must be unique within the service group. You must enable the automatic scaling option for each domain based server that you want to use for automatic service group scaling. If an IP address is common to one or more domains, the IP address is added to the service group only once.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure a service group to scale automatically by using the command line interface

At the command prompt, type the following commands to configure the service group and verify the configuration:

- add serviceGroup <serviceGroupName> <serverName> <port> -autoScale (**YES** | **NO**)
- show serviceGroup <serviceGroupName>

Example

In the following example, server1 is a domain based server. The DNS response contains multiple IP addresses. Five addresses are available and are added to the service group.

```
> add serviceGroup servGroup server1 80 -autoScale YES
 Done
> sh servicegroup servGroup
     servGroup - HTTP
      State: ENABLED  Monitor Threshold : 0
          . . .
          . . .
     1)   192.0.2.31:80   State: UP      Server Name: server1 (Auto scale)   Server ID: None Weight: 1

          Monitor Name: tcp-default      State: UP
          Probes: 2      Failed [Total: 0 Current: 0]
          Last response: Success - TCP syn+ack received.

     2)   192.0.2.32:80   State: UP      Server Name: server1 (Auto scale)   Server ID: None Weight: 1

          Monitor Name: tcp-default      State: UP
          Probes: 2      Failed [Total: 0 Current: 0]
          Last response: Success - TCP syn+ack received.

     3)   192.0.2.36:80   State: UP      Server Name: server1 (Auto scale)   Server ID: None Weight: 1

          Monitor Name: tcp-default      State: UP
          Probes: 2      Failed [Total: 0 Current: 0]
          Last response: Success - TCP syn+ack received.

     4)   192.0.2.55:80   State: UP      Server Name: server1 (Auto scale)   Server ID: None Weight: 1

          Monitor Name: tcp-default      State: UP
          Probes: 2      Failed [Total: 0 Current: 0]
          Last response: Success - TCP syn+ack received.

     5)   192.0.2.80:80   State: UP      Server Name: server1 (Auto scale)   Server ID: None Weight: 1

          Monitor Name: tcp-default      State: UP
          Probes: 2      Failed [Total: 0 Current: 0]
          Last response: Success - TCP syn+ack received.
 Done
```

To configure a service group to scale automatically by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Create a service group, and set the autoscale mode to DNS.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Translating the IP Address of a Domain-Based Server

Nov 12, 2013

To simplify maintenance on the NetScaler appliance and on the domain-based servers that are connected to it, you can configure IP address masks and translation IP addresses. These functions work together to parse incoming DNS packets and substitute a new IP address for a DNS-resolved IP address.

When configured for a domain-based server, IP address translation enables the appliance to locate an alternate server IP address in the event that you take the server down for maintenance or if you make any other infrastructure changes that affect the server.

When configuring the mask, you must use standard IP mask values (a power of two, minus one) and zeros, for example, 255.255.0.0. Non-zero values are only permitted in the starting octets.

When you configure a translation IP for a server, you create a 1:1 correspondence between a server IP address and an alternate server that shares leading or trailing octets in its IP address. The mask blocks particular octets in the original server's IP address. The DNS-resolved IP address is transformed to a new IP address by applying the translation IP address and the translation mask.

For example, you can configure a translation IP address of 10.20.0.0 and a translation mask of 255.255.0.0. If a DNS-resolved IP address for a server is 40.50.27.3, this address is transformed to 10.20.27.3. In this case, the translation IP address supplies the first two octets of the new address, and the mask passes through the last two octets from the original IP address. The reference to the original IP address, as resolved by DNS, is lost. Monitors for all services to which the server is bound also report on the transformed IP address.

When configuring a translation IP address for a domain-based server, you specify a mask and an IP address to which the DNS-resolved IP address is to be translated.

Note: Translation of the IP address is only possible for domain-based servers. You cannot use this feature for IP-based servers. The address pattern can be based on IPv4 addresses only.

To configure a translation IP address for a server by using the command line interface

At the command prompt, type:

add server <name>@ <serverDomainName> -translationIp <translationIPAddress> -translationMask <netMask> -state <ENABLED|DISABLED>

Example

add server myMaskedServer www.example.com -translationIp 10.10.10.10 -translationMask 255.255.0.0 -state ENABLED

To configure a translation IP address for a server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Servers, create a domain-based server, and specify a translation IP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Masking a Virtual Server IP Address

Dec 27, 2017

You can configure a mask and a pattern instead of a fixed IP address for a virtual server. This enables traffic that is directed to any of the IP addresses that match the mask and pattern to be rerouted to a particular virtual server. For example, you can configure a mask that allows the first three octets of an IP address to be variable, so that traffic to 111.11.11.198, 22.22.22.198, and 33.33.33.198 is all sent to the same virtual server.

By configuring a mask for a virtual server IP address, you can avoid reconfiguration of your virtual servers due to a change in routing or another infrastructure change. The mask allows the traffic to continue to flow without extensive reconfiguration of your virtual servers.

The mask for a virtual server IP address works somewhat differently from the IP pattern definition for a server described in Translating the IP Address of a Domain-Based Server. For a virtual server IP address mask, a non-zero mask is interpreted as an octet that is considered. For a service, the non-zero value is blocked.

Additionally, for a virtual server IP address mask, either leading or trailing values can be considered. If the virtual server IP address mask considers values from the left of the IP address, this is known as a forward mask. If the mask considers the values to the right side of the address, this is known as a reverse mask.

Note: The NetScaler appliance evaluates all forward mask virtual servers before evaluating reverse mask virtual servers.
When masking a virtual server IP address, you also need to create an IP address pattern for matching incoming traffic with the correct virtual server. When the appliance receives an incoming IP packet, it matches the destination IP address in the packet with the bits that are considered in the IP address pattern, and after it finds a match, it applies the IP address mask to construct the final destination IP address.

Consider the following example:

- Destination IP address in the incoming packet: 10.102.27.189
- IP address pattern: 10.102.0.0
- IP mask: 255.255.0.0
- Constructed (final) destination IP address: 10.102.27.189.

In this case, the first 16 bits in the original destination IP address match the IP address pattern for this virtual server, so this incoming packet is routed to this virtual server.

If a destination IP address matches the IP patterns for more than one virtual server, the longest match takes precedence. Consider the following example:

- Virtual Server 1: IP pattern 10.10.0.0, IP mask 255.255.0.0
- Virtual Server 2: IP pattern 10.10.10.0, IP mask 255.255.255.0
- Destination IP address in the packet: 10.10.10.45.
- Selected virtual server: Virtual Server 2.

The pattern associated with Virtual Server 2 matches more bits than that associated with Virtual Server 1, so IPs that match it will be sent to Virtual Server 2.

Note: Ports are also considered if a tie-breaker is required.

To configure a virtual server IP address mask by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

add lb vserver <name>@ http -ipPattern <ipAddressPattern> -ipMask <ipMask> <listenPort>

Example

Pattern matching based on prefix octets:

add lb vserver myLBVserver http -ippattern 10.102.0.0 -ipmask 255.255.0.0 80

Pattern matching based on trailing octets:

add lb vserver myLBVserver1 http -ippattern 0.0.22.74 -ipmask 0.0.255.255 80

Modify a pattern-based virtual server:

set lb vserver myLBVserver1 -ippattern 0.0.22.74 -ipmask 0.0.255.255

To configure a virtual server IP address mask by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the Address Type list, select IP Pattern, and specify an IP pattern and IP mask.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Load Balancing for Commonly Used Protocols

Nov 02, 2017

In addition to Web sites and Web-based applications, other types of network-deployed applications that use other common protocols often receive large amounts of traffic and therefore benefit from load balancing. Several of these protocols require specific configurations for load balancing to work properly. Among them are FTP, DNS, SIP, and RTSP.

If you configure your NetScaler appliance to use domain names for your servers rather than IPs, you may also need to set up IP translation and masking for those servers.

To configure load balancing for commonly used protocols, see the following sections:

- Load Balancing for a Group of FTP Servers
- Load Balancing DNS Servers
- Load Balancing Domain-Name Based Services
- Load Balancing a Group of SIP Servers
- Load Balancing RTSP Servers
- Load Balancing of Remote Desktop Protocol (RDP) Servers

© 1999-2017 Citrix Systems, Inc. All rights reserved.

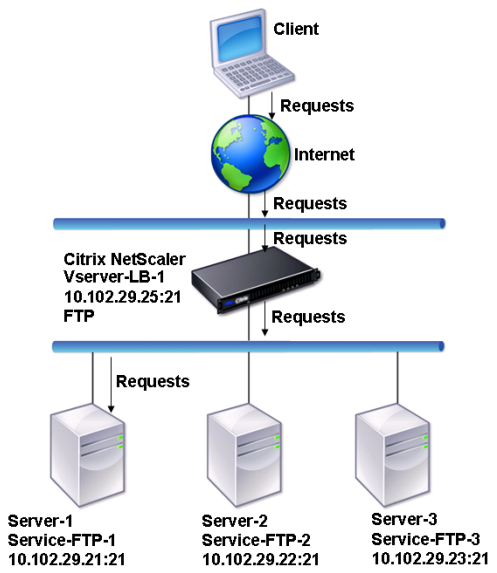
Load Balancing for a Group of FTP Servers

Nov 02, 2017

The NetScaler appliance can be used to load balance FTP servers. FTP requires that the user initiate two connections on two different ports to the same server: the control connection, through which the client sends commands to the server, and the data connection, through which the server sends data to the client. When the client initiates an FTP session by opening a control connection to the FTP server, the appliance uses the configured load balancing method to select an FTP service, and forwards the control connection to it. The load balanced FTP server then opens a data connection to the client for information exchange.

The following diagram describes the topology of a load balancing configuration for a group of FTP servers.

Figure 1. Basic Load Balancing Topology for FTP Servers



In the diagram, the services Service-FTP-1, Service-FTP-2, and Service-FTP-3 are bound to the virtual server Vserver-LB-1. Vserver-LB-1 forwards the client's connection request to one of the services using the least connection load balancing method. Subsequent requests are forwarded to the service that the appliance initially selected for load balancing.

The following table lists the names and values of the basic entities configured on the appliance.

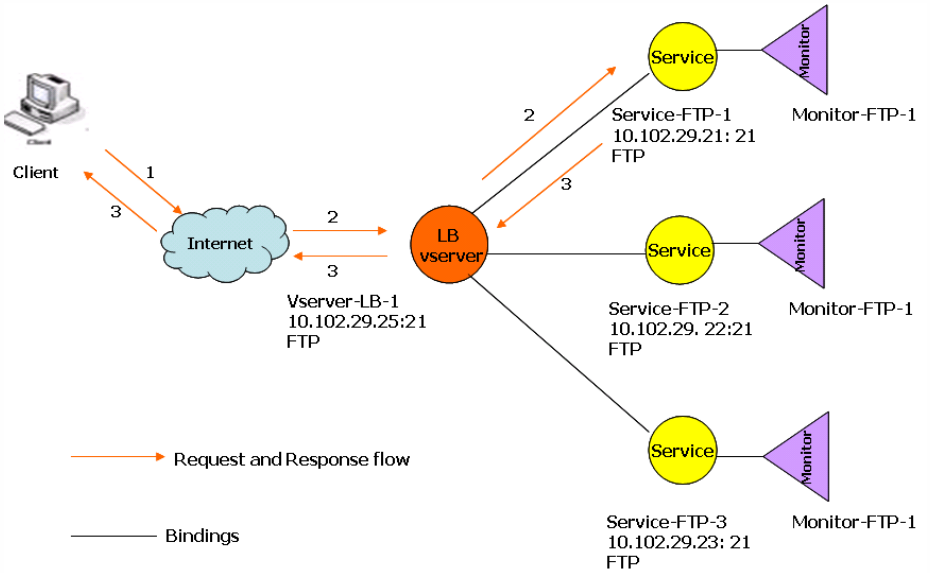
Entity type	Name	IP address	Port	Protocol
Vserver	Vserver-LB-1	10.102.29.25	21	FTP
Services	Service-FTP-1	10.102.29.21	21	FTP
	Service-FTP-2	10.102.29.22	21	FTP
	Service-FTP-3	10.102.29.23	21	FTP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity type	Name	IP address	Port	Protocol
Monitors	FTP	None	None	None

The following diagram shows the load balancing entities, and the values of the parameters that need to be configured on the appliance.

Figure 2. Load Balancing FTP Servers Entity Model



The appliance can also provide a passive FTP option to access FTP servers from outside of a firewall. When a client uses the passive FTP option and initiates a control connection to the FTP server, the FTP server also initiates a control connection to the client. It then initiates a data connection to transfer a file through the firewall.

To create services and virtual servers of type FTP, see Setting Up Basic Load Balancing. Name the entities and set the parameters to the values described in the columns of the previous table. When you configure a basic load balancing setup, a default monitor is bound to the services.

Next, bind the FTP monitor to the services by following the procedure described in the section Binding Monitors to Services.

To create FTP monitors by using the command line interface

At the command prompt, type:

add lb monitor <MonitorName> FTP -interval <Interval> -userName <UserName> -password <Password>

Example

add lb monitor monitor-FTP-1 FTP -interval 360 -userName User -password User

To create FTP monitors by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Create a monitor of type FTP, and in Special Parameters, specify a user name and password.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

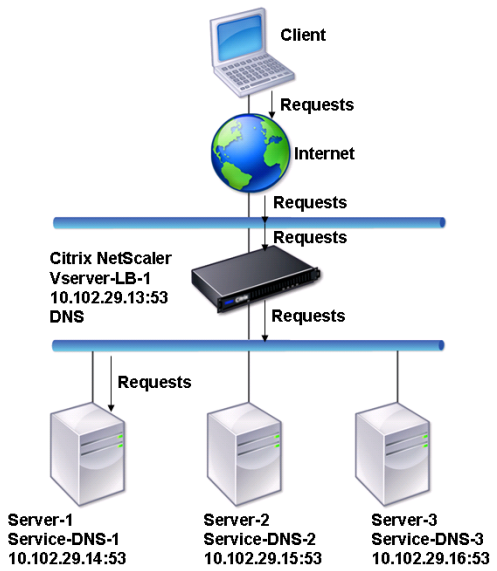
Load Balancing DNS Servers

Nov 02, 2017

When you request DNS resolution of a domain name, the NetScaler appliance uses the configured load balancing method to select a DNS service. The DNS server to which the service is bound then resolves the domain name and returns the IP address as the response. The appliance can also cache DNS responses and use the cached information to respond to future requests for resolution of the same domain name. Load balancing DNS servers improves DNS response times.

The following diagram describes the topology of a load balancing configuration that load balances a group of DNS services.

Figure 1. Basic Load Balancing Topology for DNS Servers



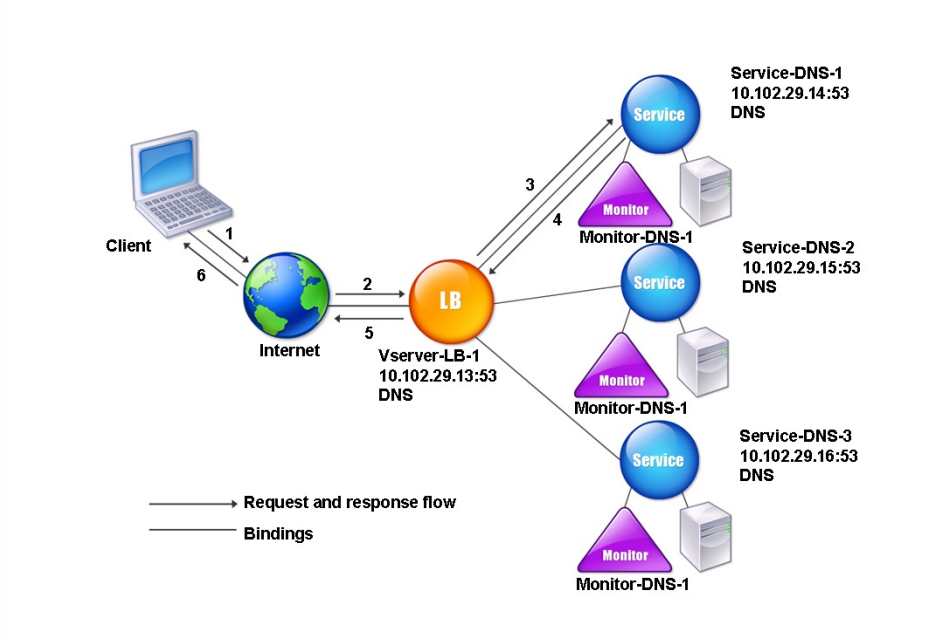
In the diagram, the services Service-DNS-1, Service-DNS-2, and Service-DNS-3 are bound to the virtual server Vserver-LB-1. The virtual server Vserver-LB-1 forwards client requests to a service using the least connection load balancing method. The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.13	53	DNS
Services	Service-DNS-1	10.102.29.14	53	DNS
	Service-DNS-2	10.102.29.15	53	DNS
	Service-DNS-3	10.102.29.16	53	DNS
Monitors	monitor-DNS-1	None	None	None

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following diagram shows the load balancing entities and the values of the parameters that need to be configured on the appliance.

Figure 2. Load Balancing DNS Servers Entity Model



To configure a basic DNS load balancing setup, see Setting Up Basic Load Balancing. Follow the procedures to create services and virtual servers of type DNS, naming the entities and setting the parameters using the values described in the previous table. When you configure a basic load balancing setup, the default ping monitor is bound to the services. For instructions on binding a DNS monitor to DNS services, you can also see Binding Monitors to Services.

The following procedure describes the steps to create a monitor that maps a domain name to the IP address based on a query.

To configure DNS monitors by using the command line interface

At the command prompt, type:

add lb monitor <monitorName> DNS -query <domainName> -queryType <**Address**|**ZONE**> -IPAddress <ipAddress>
Example

```
add lb monitor monitor-DNS-1 DNS -query www.citrix.com -queryType Address -IPAddress 10.102.29.66

add lb monitor monitor-DNS-2 DNS -query www.citrix2.com -queryType Address -IPAddress
1000:0000:0000:0000:0005:0600:700a::888b-888d
```

To configure DNS monitors by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Create a monitor of type DNS, and in Special Parameters, specify a query and query type.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing Domain-Name Based Services

Aug 06, 2017

When you create a service for load balancing, you can provide an IP address. Alternatively, you can create a server using a domain name. The server name (domain name) can be resolved using an IPv4 or IPv6 name server, or by adding an authoritative DNS record (A record for IPv4 or AAAA record for IPv6) to the NetScaler configuration.

When you configure services with domain names instead of IP addresses, and if the name server resolves the domain name to a new IP address, the monitor bound to the service runs a health check on the new IP address, and updates the service IP address only when the IP address is found to be healthy. The monitor could be the default monitor bound to the service or you can bind any other supported monitor. It probes the service at regular intervals defined in the monitor parameters. If the domain name resolves to a new IP address, the monitor sends a fresh probe to check the health of the service. All subsequent probes are at the predefined interval.

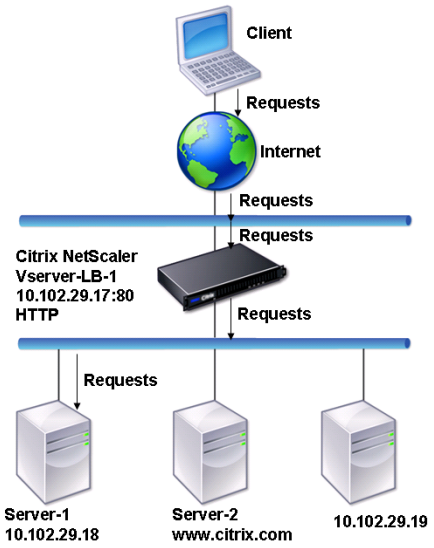
Note: When you change the IP address of a server, the corresponding service is marked down for the first client request. The name server resolves the service IP address to the changed IP address for the next request, and the service is marked UP.
Domain-name based services have the following restrictions:

- The maximum domain name length is 255 characters.
- The Maximum Client parameter is used to configure a service that represents the domain name-based server. For example, a maxClient of 1000 is set for the services bound to a virtual server. When the connection count at the virtual server reaches 2000, the DNS resolver changes the IP address of the services. However, because the connection counter on the service is not reset, the virtual server cannot take any new connections until all the old connections are closed.
- When the IP address of the service changes, persistence is difficult to maintain.
- If the domain name resolution fails due to a timeout, the appliance uses the old information (IP address).
- When monitoring detects that a service is down, the appliance performs a DNS resolution on the service (representing the domain name-based server) to obtain a new IP address.
- Statistics are collected on a service and are not reset when the IP address changes.
- If a DNS resolution returns a code of “name error” (3), the appliance marks the service down and changes the IP address to zero.

When the appliance receives a request for a service, it selects the target service. This way, the appliance balances load on your services. The following diagram describes the topology of a load balancing configuration that load balances a group of domain-name based servers (DBS).

Figure 1. Basic Load Balancing Topology for DBS Servers

© 1999-2017 Citrix Systems, Inc. All rights reserved.



The services Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 are bound to the virtual server Vserver-LB-1. The vserver Vserver-LB-1 uses the least connection load balancing method to choose the service. The IP address of the service is resolved using the name server Vserver-LB-2.

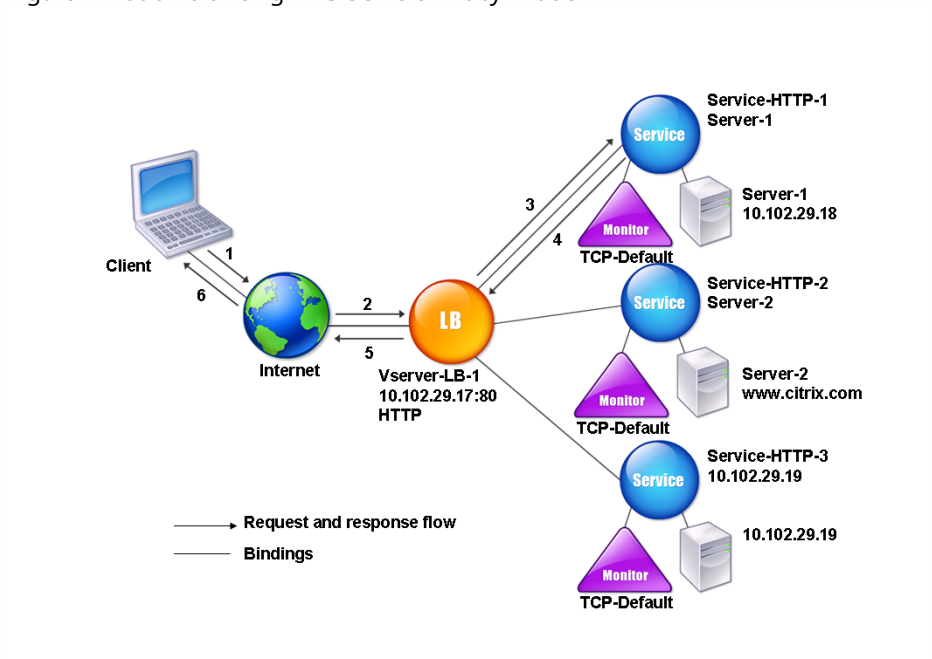
The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.17	80	HTTP
	Vserver-LB-2	10.102.29.20	53	DNS
Servers	server-1	10.102.29.18	80	HTTP
	server-2	www.citrix.com	80	HTTP
Services	Service-HTTP-1	server-1	80	HTTP
	Service-HTTP-2	server-2	80	HTTP
	Service-HTTP-2	10.102.29.19	80	HTTP
Monitors	Default	None	None	None
Name Server	None	10.102.29.19	None	None

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following diagram shows the load balancing entities and the values of the parameters that need to be configured on the appliance.

Figure 2. Load Balancing DBS Servers Entity Model



To configure a basic load balancing setup, see Setting Up Basic Load Balancing. Create the services and virtual servers of type HTTP, and name the entities and set the parameters using the values described in the previous table.

You can add, remove, enable, and disable external name servers. You can create a name server by specifying its IP address, or you can configure an existing virtual server as the name server.

To add a name server by using the command line interface

At the command prompt, type:

add dns nameServer <dnsVserverName>

Example

add dns nameServer Vserver-LB-2

To add a name server by using the configuration utility

1. Navigate to **Traffic Management** > **DNS** > **Name Servers**.
2. Create a DNS name server of type DNS Virtual Server, and select a server from the DNS Virtual Server list.

You can also add an authoritative name server that resolves the domain name to an IP address.

> **Note**
>
> You can add a name server of type TCP, UDP or UDP_TCP to resolver DBS probes. However, if TCP and UDP name servers coexists, and a UDP name server receives a response with truncated bit, this response is not retried over TCP name server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

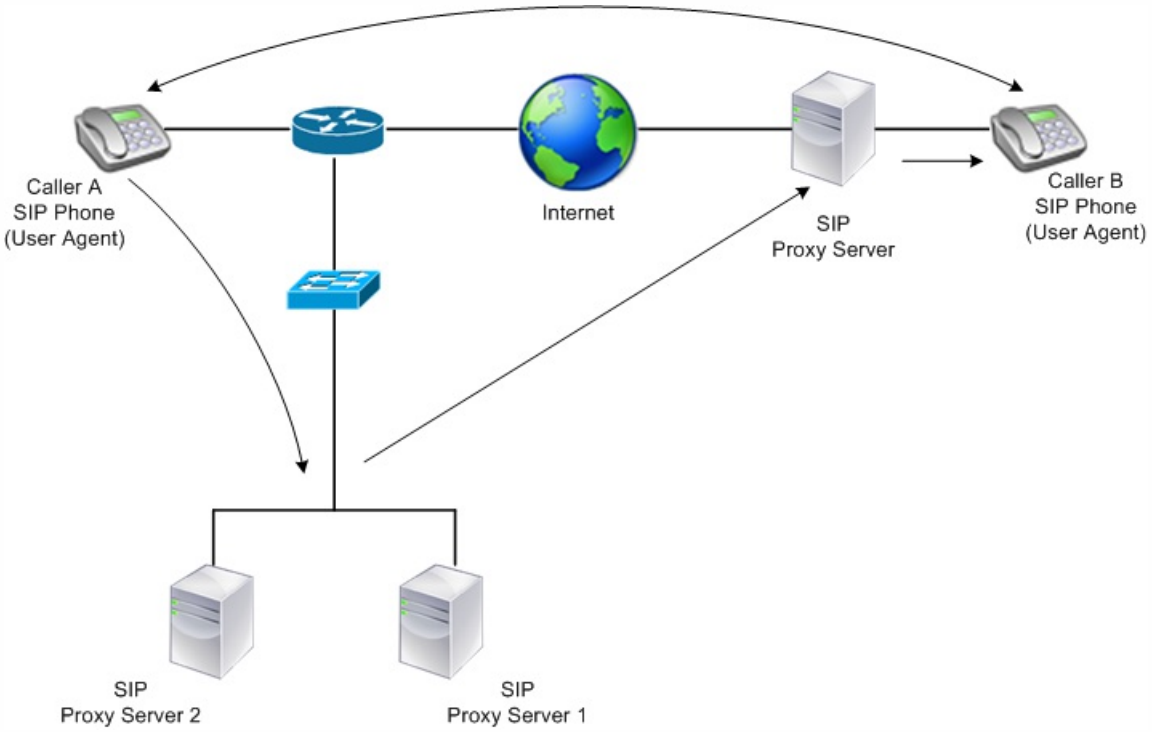
Load Balancing a Group of SIP Servers

Dec 09, 2014

The Session Initiation Protocol (SIP) is designed to initiate, manage, and terminate multimedia communications sessions. It has emerged as the standard for Internet telephony (VoIP). SIP messages can be transmitted over TCP or UDP. SIP messages are of two types: request messages and response messages.

The traffic in a SIP based communication system is routed through dedicated devices and applications (entities). In a multimedia communication session, these entities exchange messages. The following figure shows a basic SIP based communication system:

Figure 1. SIP Based Communication System



A NetScaler ADC enables you to load balance SIP messages over UDP or over TCP (including TLS). You can configure the NetScaler ADC to load balance SIP requests to a group of SIP proxy servers. To do so, you create a load balancing virtual server with the load balancing method and the type of persistence set to one of the following combinations:

- Call-ID hash load balancing method with no persistence setting
- Call-ID based persistence with least connection or round robin load balancing method
- Rule based persistence with least connection or round robin load balancing method

Also, by default, the NetScaler ADC appends RPORT to the via header of the SIP request, so that the server sends the response back to the source IP address and port from which the request originated.

Note: For load balancing to work, you must configure the SIP proxies so that they do not add private IP addresses or private domains to the SIP header/payload. SIP proxies must add to the SIP header a domain name that resolves to the IP address of the SIP virtual server. Also, the SIP proxies must communicate with a common database to share registration information.

Server Initiated Traffic

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For SIP-server initiated outbound traffic, configure RNAT on the NetScaler ADC so that the private IP addresses used by the clients are translated into public IP addresses.
If you have configured SIP parameters that include the RNAT source or destination port, the appliance compares the values of the source and destination ports of the request packets with the RNAT source port and RNAT destination port. If one of the values matches, the appliance updates the VIA header with RPORT. The SIP response from the client then traverses the same path as the request.

For server-initiated SSL traffic, the NetScaler ADC uses a built-in certificate-key pair. If you want to use a custom certificate-key pair, bind the custom certificate-key pair to the NetScaler internal service named **nsrnatsip-127.0.0.1-5061**.

Support for Policies and Expressions

The NetScaler default expressions language contains a number of expressions that operate on Session Initiation Protocol (SIP) connections. These expressions can be bound only to SIP based (sip_udp, sip_tcp or sip_ssl) virtual servers, and to global bind points. You can use these expressions in content switching, rate limiting, responder, and rewrite policies.
For more information, see SIP Expressions.

Configuring Load Balancing for SIP Signaling Traffic over TCP or UDP

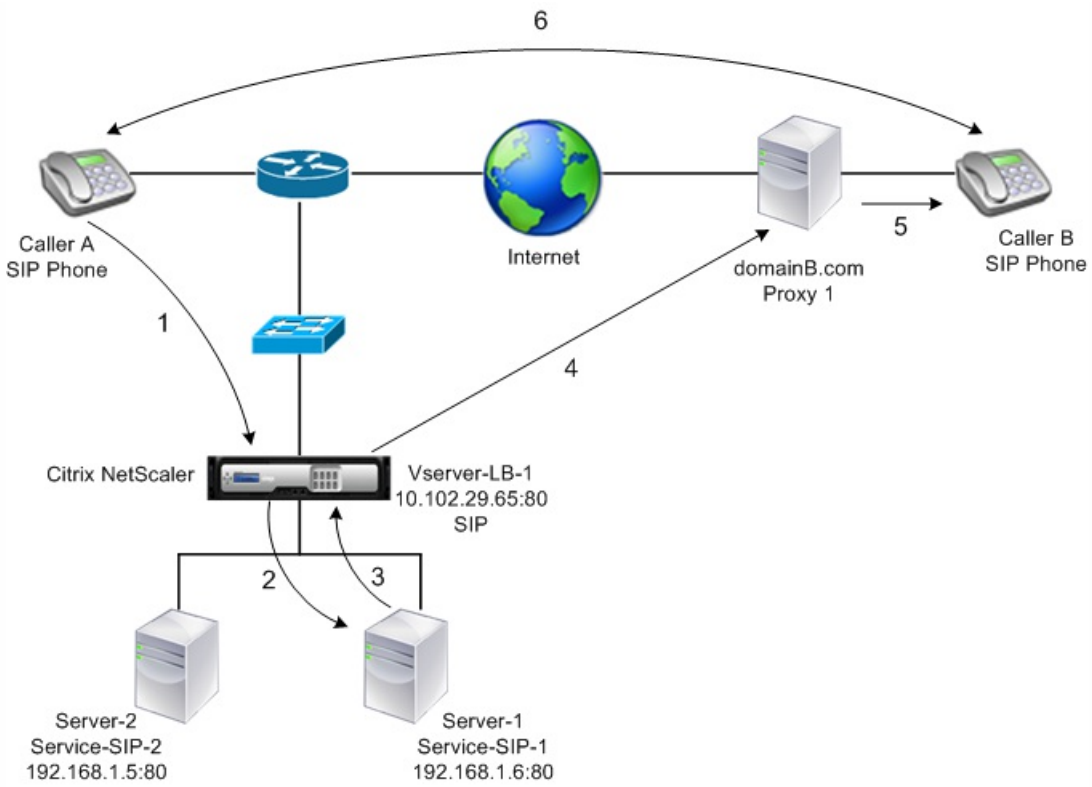
The NetScaler ADC can load balance SIP servers that send requests over UDP or TCP, including TCP traffic secured by TLS. The ADC provides the following service types to load balance the SIP servers:

- SIP_UDP – Used when SIP servers send SIP messages over UDP.
- SIP_TCP – Used when SIP servers send SIP messages over TCP.
- SIP_SSL – Used to secure SIP signaling traffic over TCP by using SSL or TLS. The NetScaler ADC supports the following modes:
  - End-to-end TLS connection between the client, the ADC, and the SIP server.
  - TLS connection between the client and the ADC, and TCP connection between the ADC and the SIP server.
  - TCP connection between the client and the ADC, and TLS connection between the ADC and the SIP server.

The following figure shows the topology of a setup configured to load balance a group of SIP servers sending SIP messages over TCP or UDP.

Figure 2. SIP Load Balancing Topology

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Entity type	Name	IP address	Port	Service type / Protocol
Virtual Server	Vserver-LB-1	10.102.29.65	80	SIP_UDP / SIP_TCP / SIP_SSL
Services	Service-SIP-1	192.168.1.6	80	SIP_UDP / SIP_TCP / SIP_SSL
	Service-SIP-2	192.168.1.5	80	SIP_UDP / SIP_TCP / SIP_SSL
Monitors	Default	None	80	SIP_UDP / SIP_TCP / SIP_SSL

Following is an overview of configuring basic load balancing for SIP traffic:

1. Configure services, and configure a virtual server for each type of SIP traffic that you want to load balance:
   - **SIP_UDP** – If you are load balancing the SIP traffic over UDP.
   - **SIP_TCP** – If you are load balancing the SIP traffic over TCP.
   - **SIP_SSL** – If you are load balancing and securing the SIP traffic over TCP.

   Note: If you use SIP_SSL, be sure to create an SSL certificate-key pair. For more information, see Adding a Certificate Key Pair.

2. Bind the services to the virtual servers.
3. If you want to monitor the states of the services with a monitor other than the default (**tcp-default**), create a custom monitor and bind it to the services. The NetScaler ADC provides two custom monitor types, **SIP-UDP** and **SIP-TCP**, for monitoring SIP services.
4. If using a SIP_SSL virtual server, bind an SSL certificate-key pair to the virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. If you are using the NetSCaler ADC as the gateway for the SIP servers in your deployment, configure RNAT.
6. If you want to append RPORT to the SIP messages that are initiated from the SIP server, configure the SIP parameters.

To configure a basic load balancing setup for SIP traffic by using the command line interface

1. Create one or more services. At the command prompt, type:
   add service <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>

   Example
   add service Service-SIP-UDP-1 192.0.2.5 SIP_UDP 80

2. Create as many virtual servers as necessary to handle the services that you created. The virtual server type must match the type of services that you will bind to it. At the command prompt, type:
   add lb vserver <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>

   Example
   add lb vserver Vserver-LB-1 SIP_UDP 10.102.29.60 80

3. Bind each service to a virtual server. At the command prompt, type:
   bind lb vserver <name> <serverName>

   Example
   bind lb vserver Vserver-LB-1 Service-SIP-UDP-1

4. (Optional) Create a custom monitor of type SIP-UDP or SIP-TCP, and bind the monitor to the service. At the command prompt, type:
   add lb monitor <monitorName> <monitorType> [<interval>]

   bind lb monitor <monitorName> <ServiceName>

   Example
   add lb monitor mon1 sip-UDP -sipMethod REGISTER -sipuRI sip:mon@test.com -sipregURI sip:mon@test.com -respcode 200

   bind monitor mon1 Service-SIP-UDP-1

5. If you created a SIP_SSL virtual server, bind an SSL certificate key pair to the virtual server. At the command prompt, type: At the command prompt, type:
   bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -CA –skipCAName

   Example
   bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1

6. Configure RNAT as required by your network topology. At the command prompt, type one of the following commands to create, respectively, an RNAT entry that uses a network address as the condition and a MIP or SNIP as the NAT IP address, an RNAT entry that uses a network address as the condition and a unique IP address as the NAT IP address, an RNAT entry that uses an ACL as the condition and a MIP or SNIP as the NAT IP address, or an RNAT entry that uses an ACL as a condition and a unique IP address as the NAT IP address:
   set rnat <IPAddress> <netmask>

   set rnat <IPAddress> <netmask> -natip <NATIPAddress>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

set rnat <aclname> [-redirectPort <port>]

set rnat <aclname> [-redirectPort <port>] -natIP <NATIPAddress>

Example
set rnat 192.168.1.0 255.255.255.0 -natip 10.102.29.50

If you want to use a custom certificate-key pair, bind the custom certificate-key pair to the NetScaler internal service named **nsrnatsip-127.0.0.1-5061**.

add ssl certKey <certkeyName> -cert <string> [-key <string>]

bind ssl service <serviceName> -certkeyName <string>

Example
add ssl certKey c1 -cert cert.epm -key key.ky

bind ssl service nsrnatsip-127.0.0.1-5061 -certkeyName c1

7. If you want to append RPORT to the SIP messages that the SIP server initiates, type the following command at the command prompt:
   set lb sipParameters -rnatSrcPort <rnatSrcPort> -rnatDstPort<rnatDstPort> -retryDur <integer> -addRportVip <addRportVip> - sip503RateThreshold <sip503_rate_threshold_value>

Sample Configuration for load balancing the SIP traffic over UDP

> add service service-UDP-1 10.102.29.5 SIP_UDP 80

Done

> add lb vserver vserver-LB-1 SIP_UDP 10.102.29.60 80

Done

> bind lb vserver vserver-LB-1 service-UDP-1

Done

> add lb mon mon1 sip-udp -sipMethod REGISTER -sipuRI sip:mon@test.com -sipregURI sip:mon@test.com -respcode 200

Done

> bind mon mon1 service-UDP-1

Done

> set rnat 192.168.1.0 255.255.255.0

Done

> set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000 -addRportVip ENABLED -sip503RateThreshold 1000

Done

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Sample Configuration for load balancing the SIP traffic over TCP

```
> add service service-TCP-1 10.102.29.5 SIP_TCP 80

Done

> add lb vserver vserver-LB-1 SIP_TCP 10.102.29.60 80

Done

> bind lb vserver vserver-LB-1 service-TCP-1

Done

> add lb mon mon1 sip-tcp -sipMethod REGISTER -sipuRI sip:mon@test.com -sipregURI sip:mon@test.com -respcode 200

Done

> bind mon mon1 service-TCP-1

Done

> set rnat 192.168.1.0 255.255.255.0

Done

> set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000 -addRportVip ENABLED -sip503RateThreshold
1000

Done
```

Sample Configuration for load balancing and securing SIP traffic over TCP

```
> add service service-SIP-SSL-1 10.102.29.5 SIP_SSL 80

Done

> add lb vserver vserver-LB-1 SIP_SSL 10.102.29.60 80

Done

> bind lb vserver vserver-LB-1 service-SIP-SSL

Done

> add lb mon mon1 sip-tCP -sipMethod REGISTER -sipuRI sip:mon@test.com -sipregURI sip:mon@test.com -respcode 200

Done

> bind mon mon1 service-SIP-SSL

Done

> bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1

Done
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> set rnat 192.168.1.0 255.255.255.0

Done

> set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000 -addRportVip ENABLED -sip503RateThreshold 1000

Done
```

To configure a basic load balancing setup for SIP traffic by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and add a virtual server of type SIP_UDP, SIP_TCP, or SIP_SSL.
2. Click the Service section, and add a service of type SIP_UDP, SIP_TCP, or SIP_SSL.
3. (Optional) Click the Monitor section, and add a monitor of type: SIP-UDP or SIP-TCP.
4. Bind the monitor to the service, and bind the service to the virtual server.
5. If you created a SIP_SSL virtual server, bind an SSL certificate key pair to the virtual server. Click the Certificates section, and bind a certificate key pair to the virtual server.
6. Configure RNAT as required by your network topology. To configure RNAT:
   1. Navigate to System > Network > Routes.
   2. On the Routes page, click the RNAT tab.
   3. In the details pane, click Configure RNAT.
   4. In the Configure RNAT dialog box, do one of the following:
      - If you want to use the network address as a condition for creating an RNAT entry, click Network and set the following parameters:
        - Network
        - Netmask
      - If you want to use an extended ACL as a condition for creating an RNAT entry, click ACL and set the following parameters:
        - ACL Name
        - Redirect Port
   5. To set a MIP or SNIP address as a NAT IP address, skip to step 7.
   6. To set a unique IP address as a NAT IP, in the Available NAT IP (s) list, select the IP address that you want to set as the NAT IP, and then click Add. The NAT IP you selected appears in the Configured NAT IP(s) list.
   7. Click Create, and then click Close.

   If you want to use a custom certificate-key pair, bind the custom certificate-key pair to the NetScaler internal service named **nsrnatsip-127.0.0.1-5061**. To bind the pair:

   1. Navigate to Traffic Management > Load Balancing > Services and click the Internal Services tab.
   2. Select nsrnatsip-127.0.0.1-5061 and click **Edit**.
   3. Click the **Certificates** section and bind a certificate key pair to the internal service.
7. If you want to append RPORT to the SIP messages that the SIP server initiates, configure the SIP parameters. Navigate to Traffic Management > Load Balancing and click Change SIP settings, set the various SIP parameters.

SIP Expression and Policy Example: Compression Enabled in Client Requests

A NetScaler ADC cannot process compressed client SIP requests, so the client SIP request fails.

You can configure a responder policy that intercepts the SIP NEGOTIATE message from the client and looks for the compression header. If the message includes a compression header, the policy responds with "400 Bad Request," so that

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

the client resends the request without compressing it.

At the command prompt, type the following commands to create the responder policy:

```
> add responder action sipaction1 respondwith q{"SIP/2.0 400 Bad Request\r\n\r\n"}

Done.

> add responder policy sippol1

> add responder policy sippol1 "SIP.REQ.METHOD.EQ(\"NEGOTIATE\")&&SIP.REQ.HEADER(\"Compression\").EXISTS"
sipaction1
```

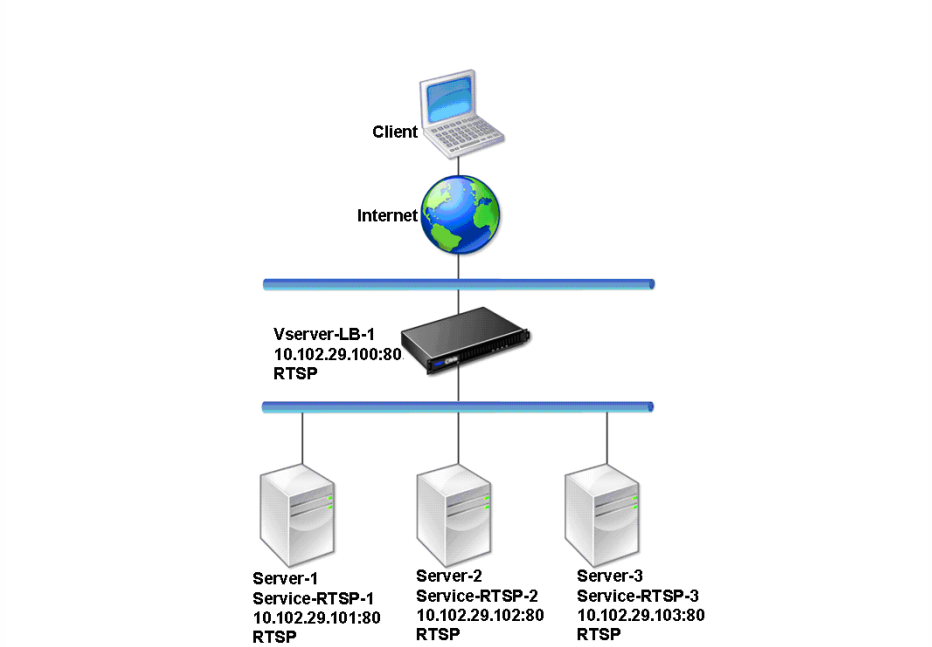

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing RTSP Servers

Nov 07, 2017

The NetScaler appliance can balance load on RTSP servers to improve the performance of audio and video streams over networks. The following diagram describes the topology of an load balancing setup configured to load balance a group of RTSP servers.

Figure 1. Load Balancing Topology for RTSP



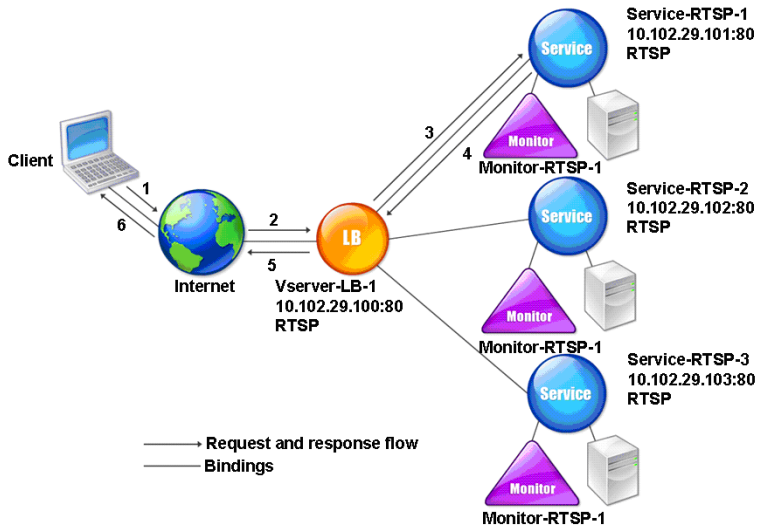
In the example, the services Service-RTSP-1, Service-RTSP-2, and Service-RTSP-3 are bound to the virtual server Vserver-LB-1. The following table lists the names and values of the example entities.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.100	554	RTSP
Services	Service-RTSP-1	10.102.29.101	554	RTSP
	Service-RTSP-2	10.102.29.102	554	RTSP
	Service-RTSP-3	10.102.29.103	554	RTSP
Monitors	Monitor-RTSP-1	None	554	RTSP

The following diagram shows the load balancing entities used in RTSP configuration.

Figure 2. Load Balancing RTSP Servers Entity Model

© 1999-2017 Citrix Systems, Inc. All rights reserved.



To configure a basic load balancing setup for RTSP servers, see Setting Up Basic Load Balancing. Create services and virtual servers of type RTSP. When you configure a basic load balancing setup, the default TCP-default monitor is bound to the services. To bind an RTSP monitor to these services, see Binding Monitors to Services. The following procedure describes how create a monitor that checks RTSP servers.

To configure RTSP monitors by using the command line interface

At the command prompt, type:

add lb monitor <monitorName> <type>

Example

add lb monitor Monitor-RTSP-1 RTSP

To configure RTSP monitors by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor of type RTSP.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing of Remote Desktop Protocol (RDP) Servers

Jun 27, 2017

Remote Desktop Protocol (RDP) is a multichannel-capable protocol that allows for separate virtual channels for carrying presentation data, serial device communication, licensing information, highly encrypted data (keyboard and mouse activity), and so on.

RDP is used for providing a graphical user interface to another computer on the network. RDP is used with Windows terminal servers for providing fast access with almost real-time transmission of mouse movements and key presses even over low-bandwidth connections.

When multiple terminal servers are deployed to provide remote desktop services, the NetScaler appliance provides load balancing of the terminal servers (Windows 2003 and 2008 Server Enterprise Editions). In some cases, a user who is accessing an application remotely may want to leave the application running on the remote machine but shut down the local machine. The user therefore closes the local application without logging out of the remote application. After reconnecting to the remote machine, the user should be able to continue with the remote application. To provide this functionality, the NetScaler RDP implementation honors the routing token (cookie) set by the Terminal Services Session Directory or Broker so that the client can reconnect to the same terminal server to which it was connected previously. The Session Directory, implemented on Windows 2003 Terminal Server, is referred to as Broker on Windows 2008 Terminal Server.

When a TCP connection is established between the client and the load balancing virtual server, the NetScaler applies the specified load balancing method and forwards the request to one of the terminal servers. The terminal server checks the session directory to determine whether the client has a session running on any other terminal server in the domain.

If there is no active session on any other terminal server, the terminal server responds by serving the client request, and the NetScaler forwards the response to the client.

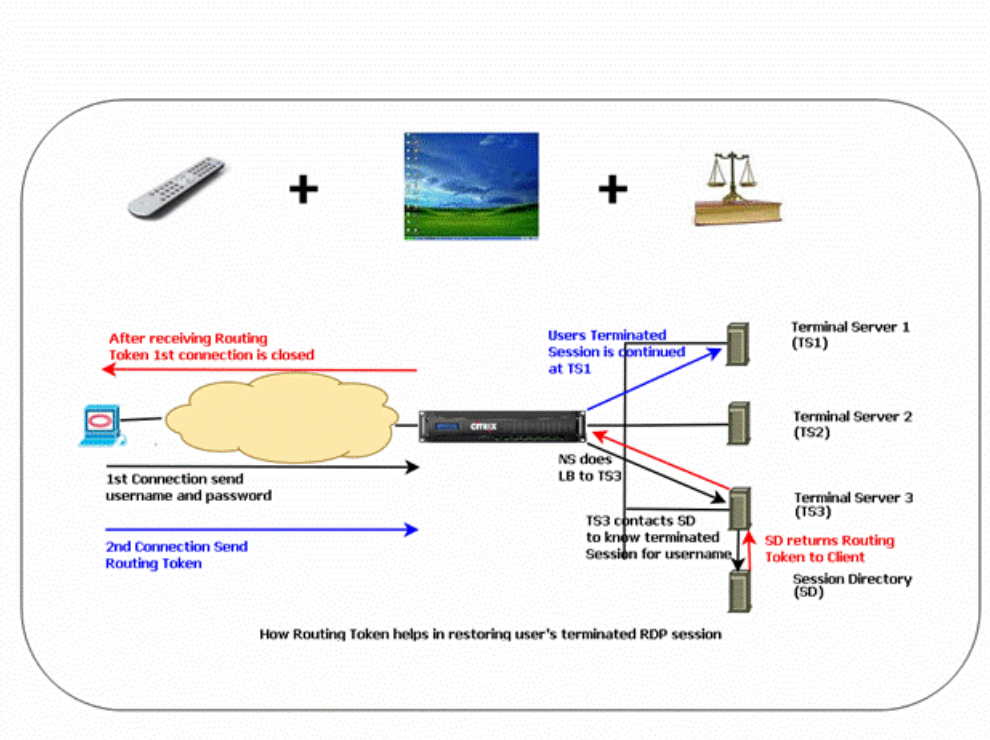
If there is an active session on any other terminal server, the terminal server that receives the request inserts a cookie (referred to as routing token) with the details of the active session and returns the packets to the NetScaler, which returns the packet to the client. The server closes the connection with the client. When the client retries to connect, the NetScaler reads the cookie information and forwards the packet to the terminal server on which the client has an active session.

The user on the client machine experiences a continuation of the service and does not have to take any specific action.

Note: The Windows Session Directory feature requires the Remote Desktop client that was first released with Windows XP. If a session with a Windows 2000 or Windows NT 4.0 Terminal Server client is disconnected and the client reconnects, the server with which the connection is established is selected by the load balancing algorithm.
The following diagram describes RDP load balancing.

Figure 1. Load Balancing Topology for RDP

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Note

- When an RDP service is configured, persistence is automatically maintained by using a routing token. You need not enable persistence explicitly.
- The NetScaler appliance supports only IP-based cookies.

Ensure that the disconnected RDP sessions are cleared on the terminal servers at the backend to avoid flapping between two terminal servers when an RDP session is disconnected without logging out. For more information, see http://technet.microsoft.com/en-us/library/cc758177(WS.10).aspx#BKMK_2

When you add an RDP service, by default, NetScaler adds a monitor of the type TCP and binds it to the service. The default monitor is a simple TCP monitor that checks whether or not a listening process exists at the 3389 port on the server specified for the RDP service. If there is a listening process at 3389, NetScaler marks this service as UP and if there is no listening process, it marks the service as DOWN.

For more efficient monitoring of an RDP service, in addition to the default monitor, you can configure a script monitor that is meant for the RDP protocol. When you configure the scripting monitor, the NetScaler opens a TCP connection to the specified server and sends an RDP packet. The monitor marks the service as UP only if it receives a confirmation of the connection from the physical server. Therefore, from the scripting monitor, the NetScaler can know whether the RDP service is ready to service a request.

The monitor is a user-type monitor and the script is located on the NetScaler at /nsconfig/monitors/nsrdp.pl. When you configure the user monitor, the NetScaler runs the script automatically. To configure the scripting monitor, add the monitor and bind it to the RDP service.

To configure RDP load balancing, create services of type RDP and bind them to an RDP virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure RDP load balancing services by using the command line interface

At the command prompt, type the following commands to configure an RDP load balancing setup and verify the configuration:

add service <name>@ <serverName> <serviceType> <port>

Note: Repeat the above command to add more services.
Example

```
> add service ser1 10.102.27.182 RDP 3389
Done
> add service ser2 10.102.27.183 RDP 3389
Done
>show service ser1
ser1 (10.102. 27.182:3389) - RDP
      State: UP
…
  Server Name: 10.102.27.182
  Server ID : 0    Monitor Threshold : 0
      Down state flush: ENABLED
…
1)     Monitor Name: tcp-default
            State: UP       Weight: 1
…
            Response Time: 4.152 millisec
 Done
```

To configure RDP load balancing services by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, and create services of type RDP.

To configure an RDP load balancing virtual server by using the command line interface

At the command prompt, type the following commands to configure an RDP load balancing virtual server and verify the configuration:

- add lb vserver <name>@ <serviceType> <ipAddress> <port>
- bind lb vserver <name>@ <serviceName>

  Bind all the RDP services to be load balanced to the virtual server.

Example

This example has two RDP services bound to the RDP virtual server.

```
> add lb vs v1 rdP 10.102.27.186 3389
Done

> bind lb vs v1 ser1
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
service "ser1" bound
> bind lb vs v1 ser2
service "ser2" bound
Done

>sh lb vs v1
v1 (10.102.27.186:3389) - RDP   Type: ADDRESS
State: UP
…
No. of Bound Services :  2 (Total)      2 (Active)
Configured Method: LEASTCONNECTION
  Current Method: Round Robin, Reason: A new service is bound
Mode: IP
Persistence: NONE
  L2Conn: OFF

1) ser1 (10.102.27.182: 3389) - RDPState: UP   Weight: 1
2) ser2 (10.102.27.183: 3389) - RDPState: UP   Weight: 1
Done
```

To configure an RDP load balancing virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers, create a virtual server of type RDP, and bind RDP services to this virtual server.

To configure a scripting monitor for RDP services by using the command line interface

At the command prompt, type the following commands:

- add lb monitor <monitorName> USER -scriptName nsrdp.pl
- bind lb monitor <monitorName> <rdpServiceName>

Example

```
add service ser1 10.102.27.182 RDP 3389
add lb monitor RDP_MON USER -scriptName nsrdp.pl
bind lb monitor RDP_MON ser1
```

To configure a scripting monitor for RDP services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor of type USER.
2. In Special Parameters, in the Script Name list, select nsrdp.pl, and then bind this monitor to an RDP service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 1: SMPP Load Balancing

Feb 17, 2016

Millions of short messages are exchanged daily between individuals and value-added service providers, such as banks, advertisers, and directory services, by using the short message peer to peer (SMPP) protocol. Often, message delivery is delayed because servers are overloaded and traffic is not optimally distributed among the servers. The NetScaler ADC supports SMPP load balancing and provides optimal distribution of messages across your servers, preventing poor performance and outages.

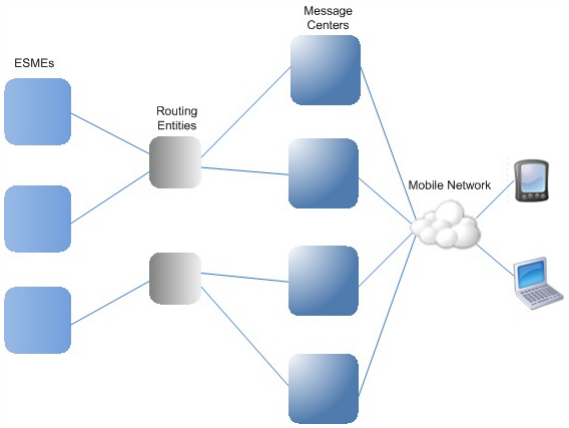
The NetScaler ADC performs load balancing on the server side when messages are received from clients and on the client side when messages are received from servers.

Load balancing of SMPP messages by the NetScaler ADC provides the following benefits:

- Better load distribution on servers, which translates to faster response time to end users
- Server health monitoring and better failover capabilities
- Quick and easy addition of new servers (message centers) without changing the client configuration
- High availability

Introduction to SMPP

SMPP is an application layer protocol for transfer of short messages between External Short Message Entities (ESME), Routing Entities (RE) and Message Centers (MC) over long-lived TCP connections. It is used for sending short message service (SMS) messages between friends, contacts, and third parties such as banks (mobile banking), advertisers (mobile commerce), and directory services. Messages from an ESME (non-mobile entity) arrive at the MC, which distributes them to short message entities (SMEs) such as mobile phones. SMPP is also used by SMEs to send short messages to third parties (for example, for purchase of products, bill payment, and funds transfer). These messages arrive at the MC and are forwarded to the destination MC or ESME.
The following diagram shows the different SMPP entities: ESMEs, REs, and MCs, in a mobile network.



Architecture Overview of the Different SMPP Entities in a Mobile Network

Note: The terms client and ESME are used interchangeably throughout the document.
An ESME (client) opens a connection to the MC in one of the three modes: as a transmitter, a receiver, or a transceiver. As a transmitter, it can only submit messages for delivery. As a receiver, it can only receive messages. As a transceiver, the ESME can both submit and receives messages. The ESME sends the MC one of the three messages (also known as PDUs): bind_transmitter, bind_receiver, or bind_transceiver. The MC responds with a bind_transmitter_resp, bind_receiver_resp, or bind_transceiver_resp, as appropriate for the request.

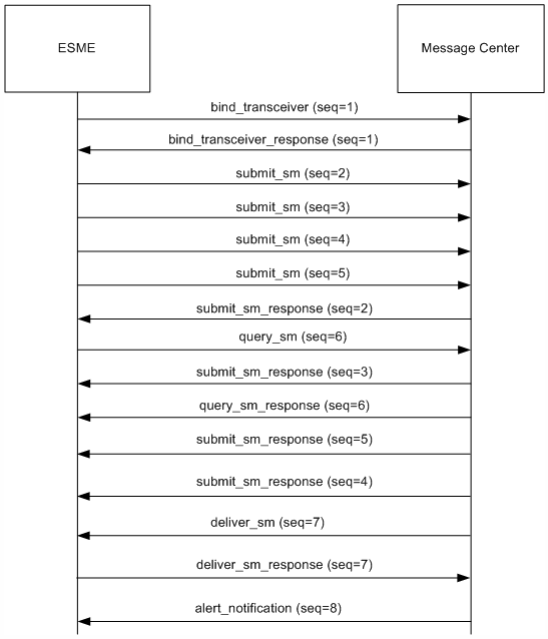
After the connection is established, the ESME can, depending on the mode in which it is bound to the MC, send a submit_sm or data_sm message, receive a deliver_sm or data_sm message, or send and receive any of these types of messages. The ESME can also send ancillary messages, such as query_sm, replace_sm, and cancel_sm, to query the status of an earlier message delivery, replace an earlier message with a new message, or cancel an undelivered message.

If a message is not delivered because an ESME is not available or a mobile subscriber is not online, the message is queued. Later, when the MC detects that the mobile subscriber is now reachable, it sends an alert_notification PDU to the ESME over a receiver or transceiver session, requesting delivery of any queued messages.

Each request PDU has a unique sequence number. The response PDU has the same sequence number as the original request. Because message exchange over SMPP can be in asynchronous mode, an ESME or an MC can send multiple requests at a time. The sequence number plays a crucial role in returning the response in the same SMPP session. In other words, the sequence number makes request and response matching possible.

The following diagram shows how the traffic flow uses the various PDUs when the ESME binds as a transceiver.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Limitation

The NetScaler appliance does not support outbind operations. That is, a message center cannot initiate an SMPP session with an ESME through the NetScaler appliance.

How SMPP Load Balancing Works on the NetScaler ADC

An ESME (client) sends a bind message to open a connection to the NetScaler ADC. The ADC authenticates each ESME and, if successful, responds with an appropriate message. The NetScaler ADC establishes a connection with each message center and load balances all the messages among these message centers. When the ADC receives a message from a client, it reuses an open connection to the message center or sends a bind request to a message center if an open connection is not available.

The ADC can load balance messages originating from the clients and from the servers. It can monitor the health of the message centers and handle concatenated messages. It also provides content switching support for the message centers.

Messages Originating from the ESMEs

Each ESME must be added as a user on the NetScaler ADC for authentication. The client establishes a TCP connection with an SMPP virtual server configured on the ADC by sending a bind request. The ADC authenticates the client and, if successful, parses the bind message. The ADC then sends the request to the message center selected by the configured load balancing method. If a connection to the message center is not available for reuse, the ADC opens a TCP connection with the message center by sending a new bind request to the message center.
Before forwarding the response (submit_sm_resp or data_sm_resp) from the message center to the client, the ADC adds a custom server ID to the message ID to identify the message center for ancillary operations, such as query, replace, or cancel requests for a message, by the client. Requests from other clients are load balanced in the same way.

In the original bind request, a client specifies the address range that it can serve. This range is used for forwarding deliver_sm or data_sm messages from the message centers to the clients.

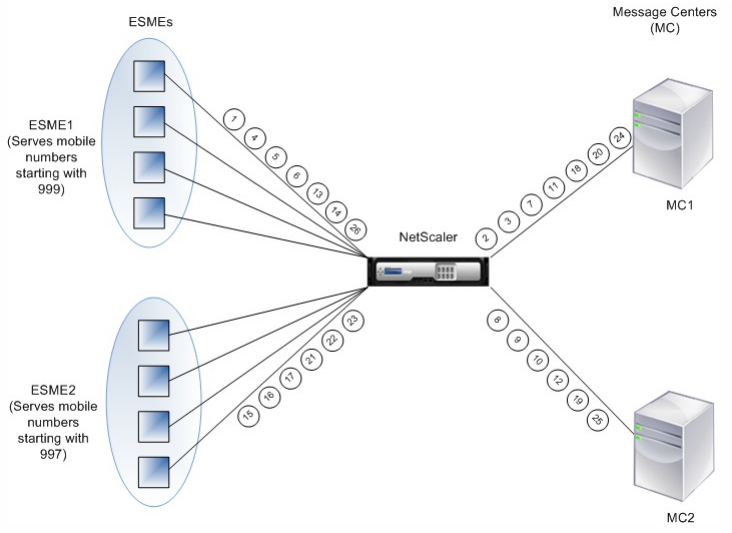
Messages Originating from a Message Center

ESMEs that can handle a specific address range are grouped into a cluster. All the nodes in a cluster provide the same credentials. Within a cluster, only the round robin method is used for load balancing. To deliver mobile originated (MO) messages, the message center sends a deliver_sm message to the NetScaler ADC. If a cluster that can serve the destination address range (for example, numbers starting with 998) is bound to the ADC, it selects that cluster, and then load balances the message among the ESME nodes in that cluster.

If an ESME that can serve deliver_sm messages for the address range is not bound to the ADC, and message queuing is enabled, the message is queued until such a client binds to the ADC in a receiver or transceiver mode. You can specify the size of the queue.

The following diagram illustrates the internal flow of PDUs between ESMEs, NetScaler ADC, and the message centers. For simplicity, only two ESMEs and two message centers are shown.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Flow of messages (PDUs):

1. ESME1 sends bind request to NetScaler
2. NetScaler sends bind request to MC1
3. MC1 sends bind response to NetScaler
4. NetScaler sends bind response to ESME1
5. ESME1 sends submit_sm(1) to NetScaler
6. ESME1 sends submit_sm(2) to NetScaler
7. NetScaler forwards submit_sm(1) to MC1
8. NetScaler sends bind request to MC2
9. MC2 sends bind response to NetScaler
10. NetScaler forwards submit_sm(2) to MC2
11. MC1 sends submit_sm_resp(1) to NetScaler
12. MC2 sends submit_sm_resp(2) to NetScaler
13. NetScaler forwards submit_sm_resp(1) to ESME1
14. NetScaler forwards submit_sm_resp(2) to ESME1
15. ESME2 sends bind request to NetScaler
16. NetScaler sends bind response to ESME2
17. ESME2 sends submit_sm(3) to NetScaler
18. NetScaler forwards submit_sm(3) to MC1
19. MC2 sends deliver_sm to NetScaler (ESME2 serves the address range specified in the message)
20. MC1 sends submit_sm_resp(3) to NetScaler
21. NetScaler forwards submit_sm_resp(3) to ESME2
22. NetScaler forwards deliver_sm to ESME2
23. ESME2 sends deliver_sm_resp to NetScaler
24. MC1 sends alert_notification to NetScaler (ESME1 serves the address range specified in the message)
25. NetScaler forwards deliver_sm_resp to MC2
26. NetScaler forwards the alert_notification to ESME1

Health Monitoring of Message Centers

By default, a TCP_default monitor is bound to an SMPP service, but you can bind a custom monitor of type SMPP. The custom monitor opens a TCP connection to the message center and sends an enquire_link packet. Depending on the success or failure of the probe, the service is marked UP or DOWN.

Content Switching on Message Centers

Message centers can accept multiple connections (or bind requests) from ESMEs. You can configure the NetScaler ADC to content switch these requests on the basis of the SMPP bind parameters. Following are some common expressions for configuring methods to select a message center:

- Based on the address range: In the following sample expression, the ADC selects a specific message center if the address range starts at 988.
  Example

  ```
  SMPP.BINDINFO.ADDRESS_RANGE.CONTAINS(\"^988\")
  ```

- Based the ESME ID: In the following sample expression, the ADC selects a specific message center if the ESME ID equals ESME1.
  Example

  ```
  SMPP.BINDINFO.SYSTEM_ID.EQ("ESME1")
  ```

- Based on the ESME type: In the following sample expression, the ADC selects a specific message center if the ESME type is VMS. VMS stands for voice mail system.
  Example

  ```
  SMPP.BINDINFO.SYSTEM_TYPE.EQ("VMS")
  ```

- Based on the type of number (TON) of the ESME: In the following sample expression, the ADC selects a specific message center if TON equals 1 (1 stands for an international number.)
  Example

  ```
  SMPP.BINDINFO.ADDR_TON.EQ(1)
  ```

- Based on the number plan indicator (NPI) of the ESME: In the following sample expression, the ADC selects a specific message center if NPI equals 0 (0 stands for an unknown connection.)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

SMPP.BINDINFO.ADDR_NPI.EQ(0)

- Based on the bind type: In the following sample expression, the ADC selects a specific message center if the bind type is TRANSCEIVER. (A transceiver can send and receive messages.)
  Example

  SMPP.BINDINFO.TYPE.EQ(TRANSCEIVER)

Concatenated Message Handling

An SMS can hold a maximum of 140 bytes. Longer messages must be broken down into smaller parts. If the destination mobile is capable, the messages are combined and delivered as one long SMS. The NetScaler ADC forwards the fragments of a message to the same message center. Each message contains a reference number, a sequence number, and the total number of fragments. The reference number is the same for each fragment of a long message. The sequence number specifies that position of the particular fragment in the complete message. After all the fragments are received, the ESME combines the fragments into one long message and delivers the message to the mobile subscriber.

If a client disconnects from an active connection, the connection to the message center is not closed. It is reused for requests from other clients.

Limitation

Message IDs, from the message center, longer than 59 bytes are not supported. If the message ID length returned by the message center is more than 59 bytes, ancillary operations fail and the NetScaler ADC responds with an error message.

Configuring SMPP Load Balancing on the NetScaler ADC

Perform the following tasks to configure SMPP load balancing on the ADC:

1. Add an SMPP user. The ADC authenticates the user before it accepts a bind request from the user. The user is typically an ESME.
2. Add a load balancing virtual server, specifying the protocol as SMPP.
3. Add a service, specifying the protocol as SMPP, and a custom server ID that is unique for each server. Bind the service to the load balancing virtual server created earlier.
4. Optionally, create a service group and add services to the service group.
5. Optionally, add a monitor of type SMPP-ECV and bind it to the service. A TCP-default monitor is bound by default.
6. Set the SMPP parameters, such as client mode and message queue.

To configure SMPP load balancing by using the command line

At the command prompt, type:

- add smpp user <username> -password <password>
- add service <name> <IP> SMPP <port> –customserverID <customserverID>
- add lb vserver <name> <IP> SMPP <port>
- bind lb vserver <name> <service name>
- set smpp param

Example

```
add smpp user smppclient1 -password c03ebb540695b6110eb31172f32245a1 -encrypted -encryptmethod ENCMTHD_2
add smpp user smppclient2 -password c03ebb540695b6110eb31172f32245a1 -encrypted -encryptmethod ENCMTHD_2
add service smmpsvc 10.102.84.140 SMPP 2775 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -svrTimeout 360 -CustomServerID ab -CKA N
add service smmpsvc2 10.102.81.175 SMPP 2775 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -svrTimeout 360 -CustomServerID xy -CKA
add lb vserver smppvs SMPP 10.102.239.179 2775 -persistenceType NONE -cltTimeout 180
bind lb vserver smppvs smmpsvc2
bind lb vserver smppvs smmpsvc
set smpp param -addrrange "\\d*"
```

To configure SMPP load balancing by using the configuration utility

1. Navigate to System > User Administration > SMPP Users, and add an SMPP user.
2. Navigate to Traffic Management > Load Balancing > Configure SMPP Parameters, and set the parameters as required by your deployment.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers, and add a virtual server of type SMPP.
4. Click in the Service section, add a service of type SMPP, and specify a Server ID.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 2: Configuring Rule Based Persistence Based on a Name-Value Pair in a TCP Byte Stream

Jun 08, 2015

Some protocols transmit name-value pairs in a TCP byte stream. The protocol in the TCP byte stream in this example is the Financial Information eXchange (FIX) protocol. In its traditional, non-XML implementation, the FIX protocol enables two hosts communicating over a network to exchange business or trade-related information as a list of name-value pairs (called "FIX fields"). The field format is <tag>=<value><delimiter>. This traditional tag-value format makes the FIX protocol ideal for the use case.

The tag in a FIX field is a numeric identifier that indicates the meaning of the field. For example, the tag 35 indicates the message type. The value after the equal sign holds a specific meaning for the given tag and is associated with a data type. For example, a value of A for the tag 35 indicates that the message is a logon message. The delimiter is the nonprinting "Start of Header" (SOH) ASCII character (0x01), which is the caret symbol (^). Each field is also assigned a name. For example, the field with tag 35 is the msgType field. Following is an example of a logon message.

```
8=FIX.4.1 9=61 35=A 49=INVMGR 56=BRKR 34=1 52=20000426-12:05:06 98=0 108=30 10=157
```

Your choice of persistence type for a tag-value list such as the one shown above is determined by the options that are available to you for extracting a particular string from the list. Token-based persistence methods require you to specify the offset and length of the token that you want to extract from the payload. The FIX protocol does not allow you to do that, because the offset of a given field and the length of its value can vary from one message to another (depending on the message type, the preceding fields, and the lengths of the preceding values) and from one implementation to another (depending on whether custom fields have been defined). Such variations make it impossible to predict the exact offset of a given field or to specify the length of the value that is to be extracted as the token. In this case, therefore, rule based persistence is the preferred persistence type.

Assume that a virtual server fixlb1 is load balancing TCP connections to a farm of servers hosting instances of a FIX-enabled application, and that you want to configure persistence for connections on the basis of the value of the SenderCompID field, which identifies the firm sending the message. The tag for this FIX field is 49 (shown in the earlier logon message example).

To configure rule based persistence for the load balancing virtual server, set the persistence type for the load balancing virtual server to RULE and configure the rule parameter with an expression. The expression must be one that extracts the portion of the TCP payload in which you expect to find the SenderCompID field, typecasts the resulting string to a name-value list based on the delimiters, and then extracts the value of the SenderCompID field (tag 49), as follows:

```
set lb vserver fixlb1 -persistenceType RULE -rule
"CLIENT.TCP.PAYLOAD(300).TYPECAST_NVLIST_T('=','^').VALUE(\"49\")"
```

Note: Backslash characters have been used in the expression because this is a CLI command. If you are using the configuration utility, do not enter the backslash characters.
If the client sends a FIX message that contains the name-value list in the earlier logon message example, the expression extracts the value INVMGR, and the NetScaler appliance creates a persistence session based on this value.

The argument to the PAYLOAD() function can be as large as you deem is necessary to include the SenderCompID field in the string extracted by the function. Optionally, you can use the SET_TEXT_MODE(IGNORECASE) function if you want

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the appliance to ignore case when extracting the value of the field, and the HASH function to create a persistence session based on a hash of the extracted value. The following expression uses the SET_TEXT_MODE(IGNORECASE) and HASH functions:

```
CLIENT.TCP.PAYLOAD(500).TYPECAST_NVLIST_T('=','^').SET_TEXT_MODE(IGNORECASE).VALUE("49").HASH
```

Following are more examples of rules that you can use to configure persistence for FIX connections (replace <tag> with the tag of the field whose value you want to extract):

- To extract the value of any FIX field in the first 300 bytes of the TCP payload, you can use the expression CLIENT.TCP.PAYLOAD(300).BEFORE_STR("^").AFTER_STR("<tag>=").
- To extract a string that is 20 bytes long at offset 80, cast the string to a name-value list, and then extract the value of the field that you want, use the expression CLIENT.TCP.PAYLOAD(100).SUBSTR(80,20).TYPECAST_NVLIST_T('=','^').VALUE("<tag>").
- To extract the first 100 bytes of the TCP payload, cast the string to a name-value list, and extract the value of the third occurrence of the field that you want, use the expression CLIENT.TCP.PAYLOAD(100).TYPECAST_NVLIST_T('=','^').VALUE("<tag>",2).
  Note: If the second argument that is passed to the VALUE() function is n, the appliance extracts the value of the $(n+1)^{th}$ instance of the field because the count starts from zero (0).

Following are more examples of rules that you can use to configure persistence. Only the payload-based expressions can evaluate data being transmitted through the FIX protocol. The other expressions are more general expressions for configuring persistence based on lower networking protocols.

- CLIENT.TCP.PAYLOAD(100)
- CLIENT.TCP.PAYLOAD(100).HASH
- CLIENT.TCP.PAYLOAD(100).SUBSTR(5,10)
- CLIENT.TCP.SRCPORT
- CLIENT.TCP.DSTPORT
- CLIENT.IP.SRC
- CLIENT.IP.DST
- CLIENT.IP.SRC.GET4
- CLIENT.IP.DST.GET4
- CLIENT.ETHER.SRCMAC.GET6
- CLIENT.ETHER.DSTMAC.GET5
- CLIENT.VLAN.ID

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 3: Configuring Load Balancing in Direct Server Return Mode

Jun 20, 2017

Load balancing in direct server return (DSR) mode allows the server to respond to clients directly by using a return path that does not flow through the NetScaler appliance. In DSR mode, however, the appliance can continue to perform health checks on services. In a high-data volume environment, sending server traffic directly to the client in DSR mode increases the overall packet handling capacity of the appliance because the packets do not flow through the appliance.

DSR mode has the following features and limitations:

- It supports one-arm mode and inline mode.
- The appliance ages out sessions based on idle timeout.
- Because the appliance does not proxy TCP connections (that is it does not send SYN-ACK to the client), it does not completely shut out SYN attacks. By using the SYN packet rate filter, you can control the rate of SYNs to the server. To control the rate of SYNs, set a threshold for the rate of SYNs. To get protection from SYN attacks, you must configure the appliance to proxy TCP connections. However, that requires the reverse traffic to flow through the appliance.
- In a DSR configuration, the NetScaler appliance does not replace the load balancing virtual server's IP address with the destination server's IP address. Instead, it forwards packets to a service by using the server's MAC address. The VIP must be configured on the server and ARP must be disabled for the VIP which is configured on the server to prevent the client request from bypassing the appliance when it is configured in one-arm mode. For example, a user needs to configure VIP in the loopback interface and disable the ARP for the same VIP.
- The appliance obtains the server's MAC address from the monitor bound to the service. However, custom user monitors (monitors of type USER), which use scripts stored on the NetScaler appliance, do not learn a server's MAC address. If you use only custom monitors in a DSR configuration, for each request the virtual server receives, the appliance attempts to resolve the destination IP address to a MAC address (by sending ARP requests). Because the destination IP address is a virtual IP address owned by the NetScaler appliance, the ARP requests always resolve to the MAC address of the NetScaler interface. Consequently, all traffic received by the virtual server is looped back to the appliance. If you use user monitors in a DSR configuration, you must also configure another monitor of a different type (for example, a PING monitor) for the services, ideally with a longer interval between probes, so that the MAC address of the servers can be learned.

In the example scenario, the services Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to a service, and the service responds to clients directly, bypassing the NetScaler. The following table lists the names and values of the entities configured on the NetScaler in DSR mode.

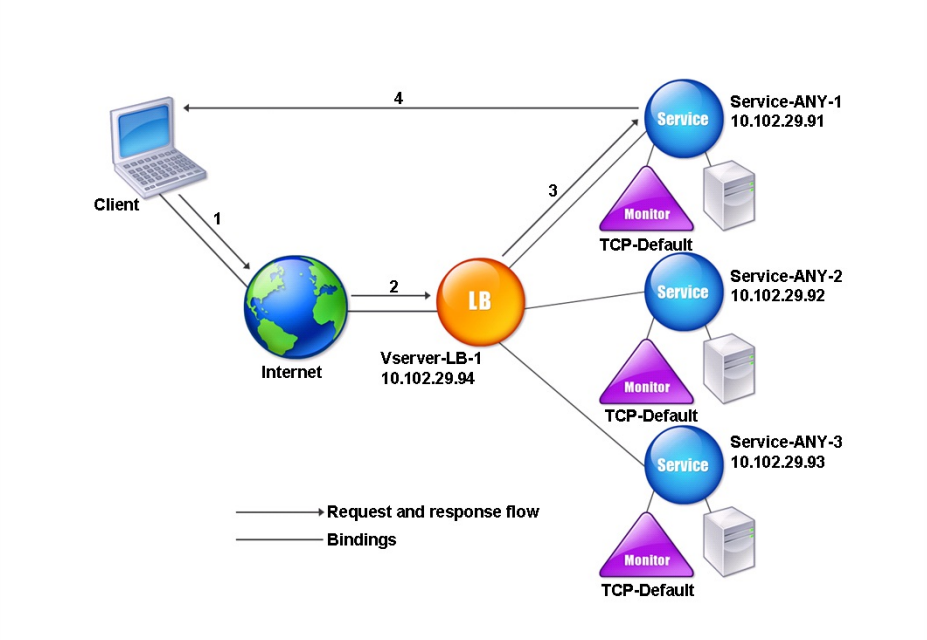
Entity type	Name	IP address	Protocol
Virtual server	Vserver-LB-1	10.102.29.94	ANY
Services	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity type	Name Service-ANY-3	IP address 10.102.29.93	Protocol ANY
Monitors	TCP	None	None

The following diagram shows the load balancing entities and values of the parameters to be configured on the appliance.

Figure 1. Entity Model for Load Balancing in DSR Model



For the appliance to function correctly in DSR mode, the destination IP in the client request must be unchanged. Instead, the appliance changes the destination MAC to that of the selected server. This setting enables the server to determine the client MAC address for forwarding requests to the client while bypassing the server.

Next, you configure a basic load balancing setup as described in Setting Up Basic Load Balancing, naming the entities and setting the parameters using the values described in the previous table.

After you configure the basic load balancing setup, you must customize it for DSR mode. To do this, you configure a supported load balancing method, such as the Source IP Hash method with a sessionless virtual server. You also need to set the redirection mode to allow the server to determine the client MAC address for forwarding responses and bypass the appliance.

After you configure the load balancing method and redirection mode, you need to enable the USIP mode on each service. The service then uses the source IP address when forwarding responses.

To configure the load balancing method and redirection mode for a sessionless virtual server by using the command line interface

At the command prompt, type:

set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <RedirectionMode> -sessionless <Value>

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPHash -m MAC -sessionless enabled
```

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

To configure the load balancing method and redirection mode for a sessionless virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server, select Redirection Mode as MAC Based, and method as SOURCEIPHASH.
3. In Traffic Settings, select Sessionless Load Balancing.

To configure a service to use source IP address by using the command line interface

At the command prompt, type:

set service <ServiceName> -usip <Value>

Example

```
set service Service-ANY-1 -usip yes
```

To configure a service to use source IP address by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open a service, and in Traffic Settings, select Use Source IP Address.

Some additional steps are required in certain situations, which are described in the succeeding sections.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 4: Configuring LINUX Servers in DSR Mode

Jun 08, 2015

The LINUX operating system requires that you set up a loopback interface with the NetScaler appliance virtual IP address (VIP) on each load balanced server in the DSR cluster.

To configure LINUX server in DSR mode

To create a loop back interface with the NetScaler appliance's VIP on each load balanced server, at the Linux OS prompt type the following commands:

ifconfig dummy0 up

ifconfig dummy0:0 inet <netscaler vip> netmask 255.255.255.255 up

echo 1 > /proc/sys/net/ipv4/conf/dummy0/arp_ignore

echo 2 > /proc/sys/net/ipv4/conf/dummy0/arp_announce

Then, run the software that re-maps the TOS id to VIP.

Note: Add the correct mappings to the software before running it. In the preceding commands, the LINUX server uses dummy0 to connect to the network. When you use this command, type the name of the interface that your LINUX server uses to connect to the network.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

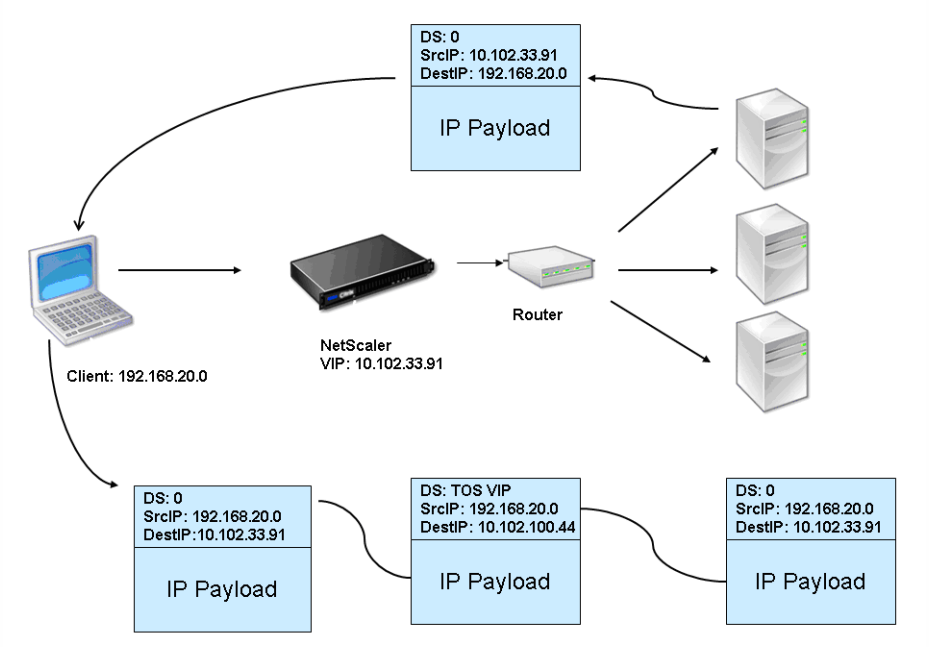
Use Case 5: Configuring DSR Mode When Using TOS

Nov 07, 2017

Differentiated services (DS), also known as TOS (Type of Service), is a field that is part of the TCP packet header. TOS is used by upper layer protocols for optimizing the path for a packet. The TOS information encodes the NetScaler appliance virtual IP address (VIP), and the load balanced servers extract the VIP from it.

In the following scenario, the appliance adds the VIP to the TOS field in the packet and then forwards the packet to the load balanced server. The load balanced server then responds directly to the client, bypassing the appliance, as illustrated in the following diagram.

Figure 1. The NetScaler Appliance in DSR mode with TOS



The TOS feature is specifically customized for a controlled environment, as described below:

- The environment must not have any stateful devices, such as stateful firewall and TCP gateways, in the path between the appliance and the load balanced servers.
- Routers at all the entry points to the network must remove the TOS field from all incoming packets to make sure that the load balanced server does not confuse another TOS field with that added by the appliance.
- Each server can have only 63 VIPs.
- The intermediate router must not send out ICMP error messages regarding fragmentation. The client will not understand the message, as the source IP address will be the IP address of the load balanced server and not the NetScaler VIP.
- TOS is valid only for IP-based services. You cannot use domain name based services with TOS.

In the example, Service-ANY-1 is created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to the service, and the service responds to clients directly, bypassing the appliance. The following table lists the names and values of the entities configured on the appliance in DSR mode.

Entity Type	Name	IP Address	Protocol

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity Type	Name	IP Address	Protocol
Virtual server	Vserver-LB-1	10.102.33.91	ANY
Services	Service-ANY-1	10.102.100.44	ANY
Monitors	PING	None	None

DSR with TOS requires that load balancing be set up on layer 3. To configure a basic load balancing setup for Layer 3, see Setting Up Basic Load Balancing. Name the entities and set the parameters using the values described in the previous table.

After you configure the load balancing setup, you must customize the load balancing setup for DSR mode by configuring the redirection mode to allow the server to decapsulate the data packet and then respond directly to the client and bypass the appliance.

After specifying the redirection mode, you can optionally enable the appliance to transparently monitor the server. This enables the appliance to transparently monitor the load balanced servers.

To configure the redirection mode for the virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -m <Value> -tosId <Value>
```

Example

```
set lb vserver Vserver-LB-1 -m TOS -tosId 3
```

To configure the redirection mode for the virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server, and in Redirect Mode, select TOS ID.

To configure the transparent monitor for TOS by using the command line interface

At the command prompt, type:

```
add monitor <MonitorName> <Type> -destip <DestinationIP> -tos <Value> -tosId <Value>
```

Example

```
add monitor mon1 PING -destip 10.102.33.91 -tos Yes  -tosId 3
```

To create the transparent monitor for TOS by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Create a monitor, select TOS, and type the TOS ID that you specified for the virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 6: Configuring Load Balancing in DSR Mode for IPv6 Networks by Using the TOS Field

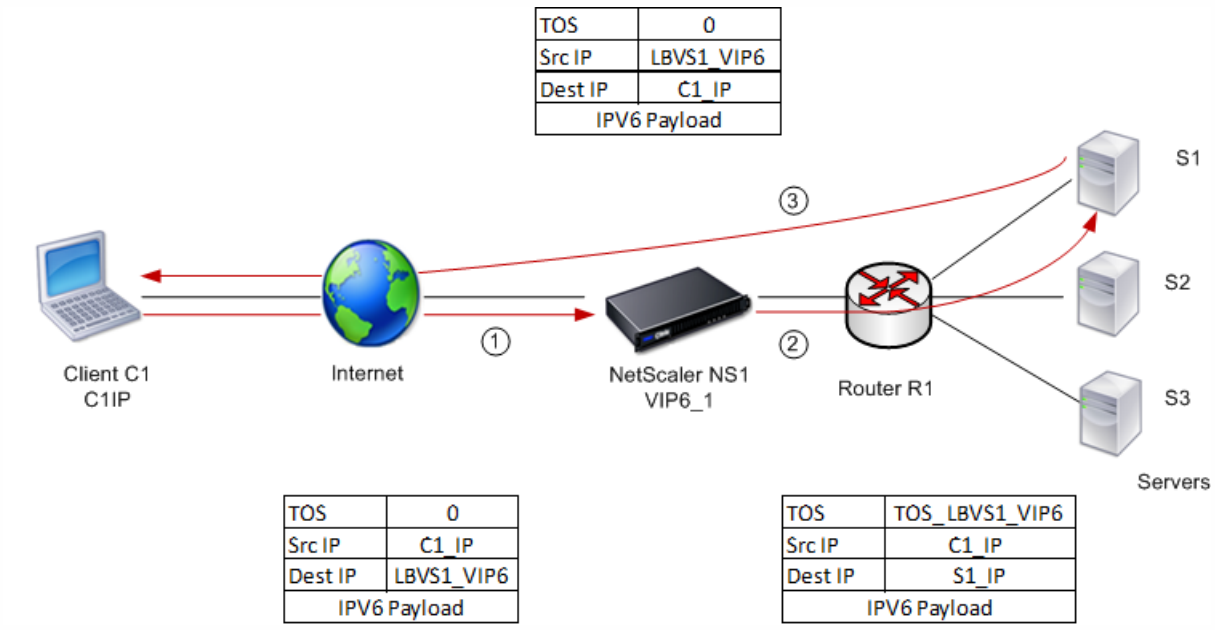
Jun 08, 2015

You can configure load balancing in Direct Server Return (DSR) mode for IPv6 networks by using the Type of Service (TOS) field when the NetScaler appliance and the servers are in different networks.

Note: The TOS field is also called the Traffic Class field.

In DSR mode, when a client sends a request to a VIP6 address on a NetScaler appliance, the appliance forwards this request to the server by changing the destination IPv6 address of the packet to the IPv6 address of the server and sets an encoded value of the VIP6 address in the TOS (also called traffic class) field of the IPv6 header. You can configure the server to use the information in the TOS field to derive the VIP6 address from the encoded value, which is then used as source IP address in response packets. Response traffic directly goes to the client, bypassing the NetScaler.

Consider an example where a load balancing virtual server LBVS1, configured on a NetScaler appliance NS1, is used to load balance traffic across servers S1, S2, and S3. The NetScaler appliance NS1 and the servers S1, S2, and S3 are in different networks so router R1 is deployed between NS1 and the servers.



The following table lists the settings used in this example.

Entities	Name
IPv6 address of client C1	C1_IP (for reference purposes only)
Load balancing virtual server on NS1	LBVS1
IPv6 address of LBVS1	LBVS1_VIP6 (for references purpose only)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entities	Name
TOS value	TOS_ LBVS1_VIP6 (for references purpose only)
Service for server S1 on NS1	SVC_S1
IPv6 address for server S1	S1_IP (for references purpose only)
Service for server S2 on NS1	SVC_S2
IPv6 address for server S1	S2_IP (for references purpose only)
Service for server S3 on NS1	SVC_S3
IPv6 address for server S1	S3_IP (for references purpose only)

Following is the traffic flow in the example scenario:

1. Client C1 sends a request to virtual server LBVS1.
2. LBVS1's load balancing algorithm selects server S1 and the appliance opens a connection to S1. NS1 sends the request to S1 with:
   - TOS field set to TOS_ LBVS1_VIP6.
   - Source IP address as C1_IP.
3. The server S1, on receiving the request, uses the information in the TOS field to derive the LBVS1_VIP6 address, which is the IP address of the virtual server LBVS1 on NS1. The server directly sends the response to C1, bypassing the NetScaler, with:
   - Source IP address set to the derivedLBVS1_VIP6 address so that the client communicates to the virtual server LBVS1 on NS1 and not to server S1.

To configure load balancing in DSR Mode using TOS, perform the following steps on the appliance:

1. Enable USIP mode globally.
2. Add the servers as services.
3. Configure a load balancing virtual server with a TOS value.
4. Bind the services to the virtual server.

To configure load balancing in DSR Mode using TOS by using the command line interface

At the command prompt, type:

- enable ns mode USIP
- add service <serviceName> <IP> <serviceType> <port>
  Repeat the above command as many times as necessary to add each server as a service on the NetScaler appliance.
- add lb vserver <name> <serviceType> <ip> <port> -m <redirectionMode> -tosId <positive_integer>
- bind lb vserver <vserverName> <serviceName>

To enable USIP mode by using the configuration utility

Navigate to System > Settings > Configure Modes, and select Use Source IP Address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To create services by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, and create a service.

To create a load balancing virtual server and bind services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and create a virtual server.
2. Click in the Service section to bind a service to this virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 7: Configuring Load Balancing in DSR Mode by Using IP Over IP

Jul 18, 2017

You can configure your NetScaler appliance to use direct server return (DSR) mode across Layer 3 networks by using IP tunneling, also called *IP over IP* configuration. As with standard load balancing configurations for DSR mode, this allows servers to respond to clients directly instead of using a return path through the NetScaler appliance, improving response times and throughput. As with standard DSR mode, the NetScaler appliance monitors the servers and performs health checks on the application ports.

With IP over IP configuration, the NetScaler appliance and the servers do not need to be on the same Layer 2 subnet. Instead, the NetScaler appliance encapsulates the packets before sending them to the destination server. After the destination server receives the packets, it decapsulates the packets, and then sends its responses directly to the client.

To configure IP over IP DSR mode on your NetScaler appliance, you must do the following:

- **Create a load balancing virtual server.** Set the protocol to ANY and set the mode to IPTUNNEL.
- **Create services.** Create a service for each of your back-end applications. Bind the services that you created to the virtual server.
- **Configure for decapsulation:** You can configure either a NetScaler appliance or a backend server to act as a decapsulator.

Configuring a Load Balancing Virtual Server

Updated: 2013-11-29

Configure a virtual server to handle requests to your applications. Assign a service type of ANY and set the forwarding method to IPTUNNEL. Optionally, configure the virtual server to operate in sessionless mode. You can configure any load balancing method that you want to use.

To create and configure a load balancing virtual server for IP over IP DSR by using the command line interface

At the command prompt type the following command to configure a load balancing virtual server for IP over IP DSR and verify the configuration:

- add lb vserver <name> serviceType <serviceType> IPAddress <ip> Port <port> -lbMethod <method> -m <ipTunnelTag> -sessionless <sessionless>
- show lb vserver <name>

Example

In the following example, we have selected the load balancing method as sourceIPhash and configured sessionless load balancing.

```
add lb vserver Vserver-LB-1 ANY 10.102.29.60 * -lbMethod SourceIPHash -m IPTUNNEL -sessionless enabled
```

To create and configure a load balancing virtual server for IP over IP DSR by using

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Create a virtual server, and specify Redirection Mode as IP Tunnel Based.

Configuring Services for IP over IP DSR

Updated: 2013-11-29

After creating your load-balanced server, You must configure one service for each of your applications. The service handles traffic from the NetScaler appliance to those applications, and allows the NetScaler appliance to monitor the health of each application.

You assign a service type of ANY and configure it for USIP mode. Optionally, you can also bind a monitor of type IPTUNNEL to the service for tunnel-based monitoring.

To create and configure a service for IP over IP DSR by using the command line interface

At the command prompt, type the following commands to create a service and optionally, create a monitor and bind it to the service:

- add service <serviceName> <serverName> <serviceType> <port> -usip <usip>
- add monitor <monitorName> <monitorType> -destip <ip> -iptunnel <iptunnel>
- bind service <serviceName> -monitorName <monitorName>

Example

In the following example, we are creating a monitor of type IPTUNNEL:

```
add monitor mon-1 PING -destip 10.102.29.60 -iptunnel yes
add service Service-DSR-1 10.102.30.5 ANY * -usip yes
bind service Service-DSR-1 -monitorName mon-1
```

To configure a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Create a monitor, and select IP Tunnel.

To create and configure a service for IP over IP DSR by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Create a service and, in Settings, select Use Source IP Address.

To bind a service to a load balancing virtual server by using the command line interface

At the command prompt type the following command:

bind lb vserver <name> <serviceName>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

bind lb vserver Vserver-LB-1 Service-DSR-1

To bind a service to a load balancing virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server, and click in the Services section to bind a service to the virtual server.

Decapsulator configuration

When a NetScaler appliance is used as a decapsulator, an IP tunnel must be created in the NetScaler appliance. For details, see Configuring IP Tunnels.

Example configuration:

NetScaler as encapsulator COPY

```
add lb vserver v1 any 1.1.1.1 * -m IPTUNNEL

add service s1 2.2.2.2 ANY *

bind lb vserver v1 s1
```

NetScaler as decapsulator COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add iptunnel tun1 <snip_in_encap> netmask *

add ns ip 1.1.1.1 255.255.255.255 –type vip –arp disabled

add lb vserver v1 any 1.1.1.1 *

add service s1 <actualserverip> ANY *

bind lb vserver v1 s1
```

- When a backend server is used as a decapsulator, the backend configuration varies depending on the server type. The steps involved in configuring a backend server as a decapsulator are;

1. Configure a loop back interface.
2. Add a route through tunnel interface.

Note: Make sure that the tunnel modules are installed in the system.

Example configuration:

In this example, 1.1.1.1 is the NetScaler virtual IP (VIP) address and 2.2.2.2 is the backend server IP address.

The VIP address is configured in the loopback interface and a route is added through the tunnel interface. The modprobe ipip command is used for enabling the tunnel interface.

NetScaler as encapsulator COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add lb vserver v1 ANY 1.1.1.1 80 -m IPTUNNEL

add service svc1 2.2.2.2 ANY 80 -usip YES -useproxyport NO

bind lb vserver v1 svc1
```

Backend server as decapsulator

COPY

```
ifconfig lo inet 1.1.1.1 netmask 255.255.255.255

modprobe ipip

echo 1 > /proc/sys/net/ipv4/conf/tunl0/arp_ignore

echo 2 > /proc/sys/net/ipv4/conf/tunl0/arp_announce

ifconfig tunl0 1.1.1.1 netmask 255.255.255.255 up

route add -host 1.1.1.1 dev tunl0
```

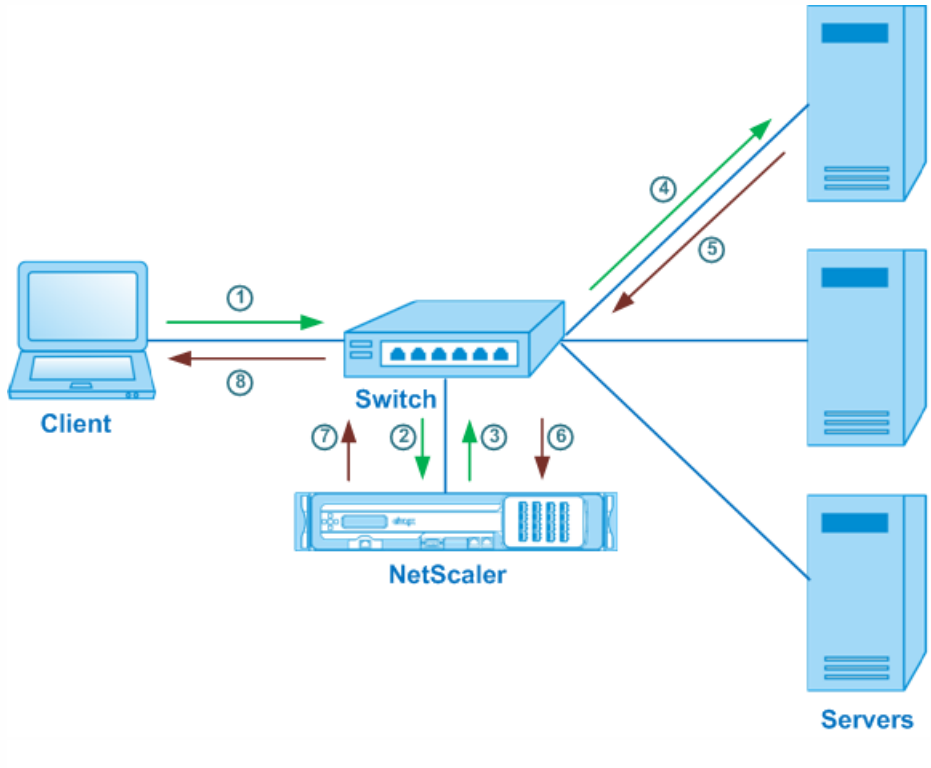

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 8: Configuring Load Balancing in One-arm Mode

Feb 25, 2016

In a one-arm setup, you connect the NetScaler appliance to the network through a single VLAN. The appliance receives the request from the client on a single VLAN and it sends the request to the server on the same VLAN. This is one of the simplest deployment scenarios, where the router, the servers and the appliance are all connected to the same switch. Client requests at the switch are forwarded to the appliance, and the appliance uses the configured load balancing method to select the service, as is shown in the following diagram.

Figure 1. Load Balancing in One-Arm Mode



In the example scenario, the services Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to a service. The following table lists the names and values of the entities configured on the appliance in one-arm mode.

Entity type	Name	IP address	Protocol
Virtual server	Vserver-LB-1	10.102.29.94	ANY
Services	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity type	Name Service-ANY-3	IP address 10.102.29.93	Protocol ANY
Monitors	TCP	None	None

To configure a load balancing setup in one-arm mode, see "Setting Up Basic Load Balancing."

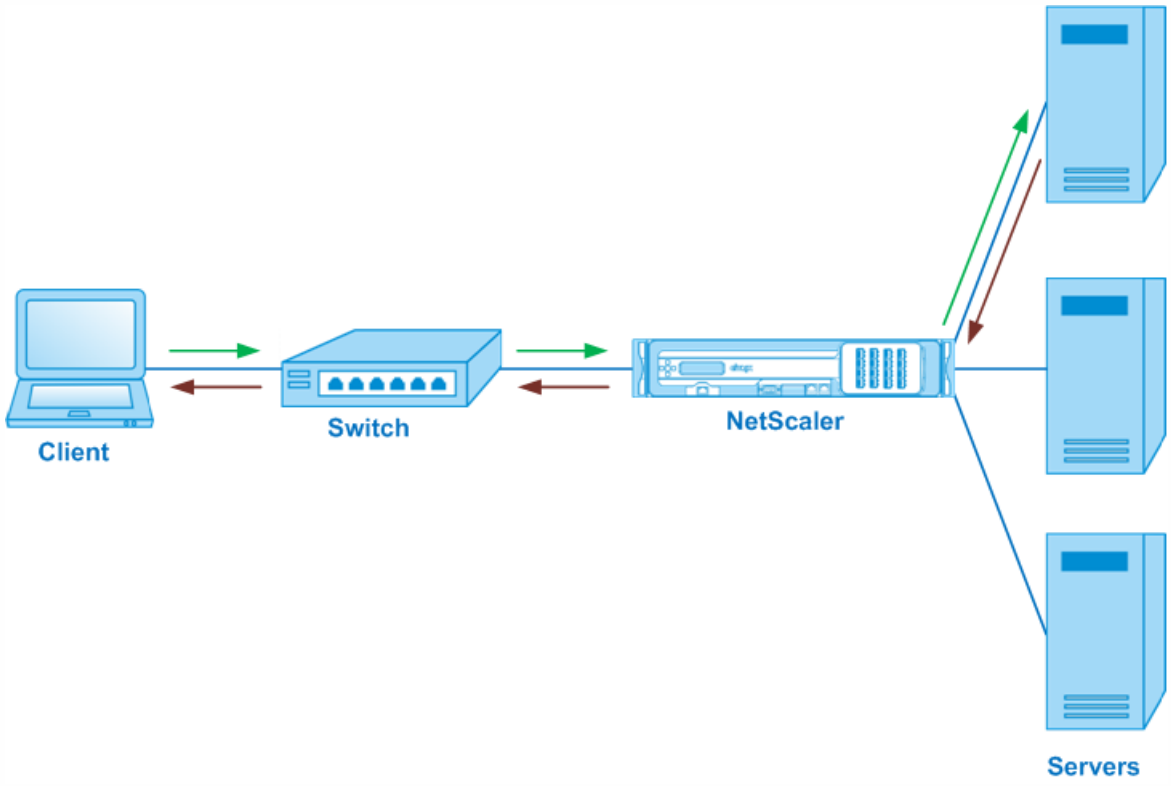
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 9: Configuring Load Balancing in the Inline Mode

Feb 19, 2016

In an inline mode (also called two-arm mode) setup, you connect the NetScaler appliance to the network through multiple VLANs. The appliance receives the request from the client on one VLAN and it sends the request to the server on another VLAN. In the two-arm setup, the appliance is connected between the servers and the client. Client requests at the switch are forwarded to the appliance, and the appliance uses the configured load balancing method to select the service. This is shown in the following diagram.

Figure 1. Load Balancing in Inline Mode



The configuration and the entity diagram for inline mode are the same as described in "Configuring Load Balancing in One-arm Mode."

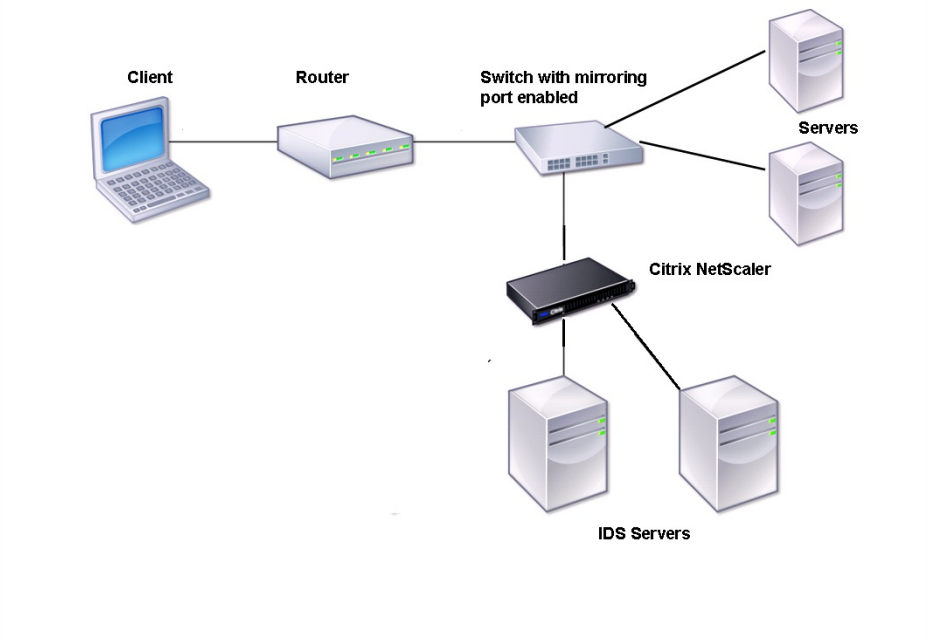
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 10: Load Balancing of Intrusion Detection System Servers

Dec 27, 2017

To enable the NetScaler appliance to support load balancing of intrusion detection system (IDS) servers, the IDS servers and clients must be connected through a switch that has port mirroring enabled. The client sends a request to the server. Because port mirroring is enabled on the switch, the request packets are copied or sent to the NetScaler appliance virtual server port. The appliance then uses the configured load balancing method to select an IDS server, as shown in the following diagram.

Figure 1. Topology of Load Balanced IDS Servers



Note: Currently, the appliance supports load balancing of passive IDS devices only.
As illustrated in the preceding diagram, the IDS load balancing setup functions as follows:

1. The client request is sent to the IDS server, and a switch with a mirroring port enabled forwards these packets to the IDS server. The source IP address is the IP address of the client, and the destination IP address is the IP address of the server. The source MAC address is the MAC address of the router, and the destination MAC address is the MAC address of the server.
2. The traffic that flows through the switch is mirrored to the appliance. The appliance uses the layer 3 information (source IP address and destination IP address) to forward the packet to the selected IDS server without changing the source IP address or destination IP address. It modifies the source MAC address and the destination MAC address to the MAC address of the selected IDS server.

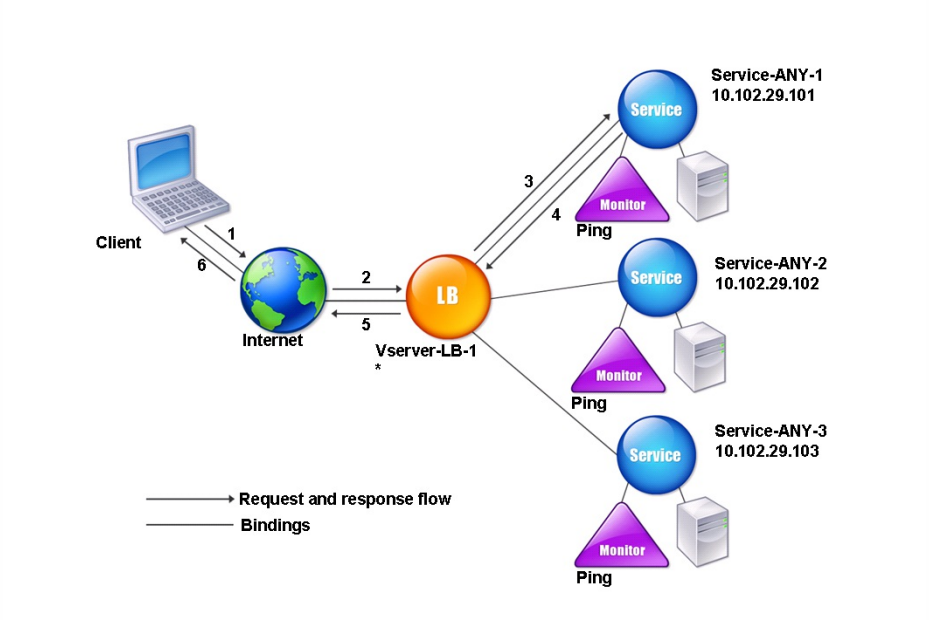
Note: When load balancing IDS servers, you can configure the SRCIPHASH, DESTIPHASH, or SRCIPDESTIPHASH load balancing methods. The SRCIPDESTIPHASH method is recommended because packets flowing from the client to a service on the appliance must be sent to a single IDS server.
Suppose Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to Vserver-LB-1. The virtual server balances the load on the services. The following table lists the names and values of the entities configured on the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity type	Name	IP address	Port	Protocol
Virtual server	Vserver-LB-1	*	*	ANY
Services	Service-ANY-1	10.102.29.101	*	ANY
	Service-ANY-2	10.102.29.102	*	ANY
	Service-ANY-3	10.102.29.103	*	ANY
Monitors	Ping	None	None	None

Note: You can use inline mode or one-arm mode for an IDS load balancing setup.
The following diagram shows the load balancing entities and values of the parameters to be configured on the appliance.

Figure 2. Entity Model for Load Balancing IDS Servers



To configure an IDS load balancing setup, you must first enable MAC-based forwarding. You must also disable layer 2 and layer 3 modes on the appliance.

To enable MAC-based forwarding by using the command line interface

At the command prompt, type:

enable ns mode <ConfigureMode>

Example

enable ns mode MAC

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To enable MAC-based forwarding by using the configuration utility

Navigate to System > Settings > Configure Modes, and select MAC Based Forwarding.

Next, see "Setting Up Basic Load Balancing", to configure a basic load balancing setup.

After you configure the basic load balancing setup, you must customize it for IDS by configuring a supported load balancing method (such as the SRCIPDESTIP Hash method on a sessionless virtual server) and enabling MAC mode. The appliance does not maintain the state of the connection and only forwards the packets to the IDS servers without processing them. The destination IP address and port remains unchanged because the virtual server is in the MAC mode.

To configure LB method and redirection mode for a sessionless virtual server by using the command line interface

At the command prompt, type:

set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <RedirectionMode> -sessionless <Value>

Example

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPDestIPHash -m MAC -sessionless enabled
```

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

To configure LB method and redirection mode for a sessionless virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server and, in Redirection Mode, select MAC Based.
3. In Advanced Settings, click Methods, and select SRCIPDESTIPHASH. Click Traffic Settings, and select Sessionless Load Balancing.

To set a service to use source IP address by using the command line interface

At the command prompt, type:

set service <ServiceName> -usip <Value>

Example

```
set service Service-ANY-1 -usip yes
```

To set a service to use source IP address by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open a service, and in Settings select Use Source IP Address.

For USIP to function correctly, you must set it globally. For more information about configuring USIP globally, see "Enabling Use Source IP Mode."
."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 11: Isolating Network Traffic using Listen Policies

Jan 01, 2018

A very common security requirement in a data center is to maintain network path isolation between the traffic of various applications or tenants. One application or tenant's traffic must be isolated from the traffic of other applications or tenants. For example, a financial services company would want to keep the traffic of its insurance department's applications separate from that of its financial services applications. In the past, this was easily achieved through physical separation of network service devices such as firewalls, load balancers, and IDP, and network monitoring and logical separation in the switching fabric.

As data center architectures evolve toward multi-tenant virtualized data centers, networking services in the aggregation layer of a data center are getting consolidated. This development has made network path isolation a critical component for network service devices and is driving the requirement for ADCs to be able to isolate traffic at the L4 to L7 levels. Furthermore, all the traffic of a particular tenant must go through a firewall before reaching the service layer.

To address the requirement of isolating the network paths, a NetScaler appliance identifies network domains and controls the traffic across the domains. The NetScaler solution has two main components: listen policies and shadow virtual servers.

Each network path to be isolated is assigned a virtual server on which a listen policy is defined so that the virtual server listens to traffic only from a specified network domain.

To isolate the traffic, listen policies can be based on a number of client parameters or their combinations, and the policies can be assigned priorities. The following table lists the parameters that can be used in listen policies for identifying the traffic.

Table 1. Client Parameters Used to Define Listen Policies

Category	Parameters
Ethernet protocol	Source MAC address, destination MAC address
Network interface	Network ID, receiving throughput, sending throughput, transmission throughput
IP protocol	Source IP address, destination IP address
IPv6 protocol	Source IPv6 address, destination IPv6 address
TCP protocol	Source port, destination port, maximum segment size, payload, and other options
UDP protocol	Source port, destination port
VLAN	ID

On the NetScaler appliance, a virtual server is configured for each domain, with a listen policy specifying that the virtual server is to listen only to traffic for that domain. Also configured for each domain is a shadow load balancing virtual server, which listens to traffic destined for any domain. Each of the shadow load balancing virtual servers has a wildcard (*) IP address and port, and its service type is set to ANY.

In each domain, a firewall for the domain is bound as a service to the shadow load balancing virtual server, which forwards all traffic through the firewall. Local traffic is forwarded to its destination, and traffic destined for another domain is forwarded to the firewall for that domain. The shadow load balancing virtual servers are configured for MAC mode redirection.

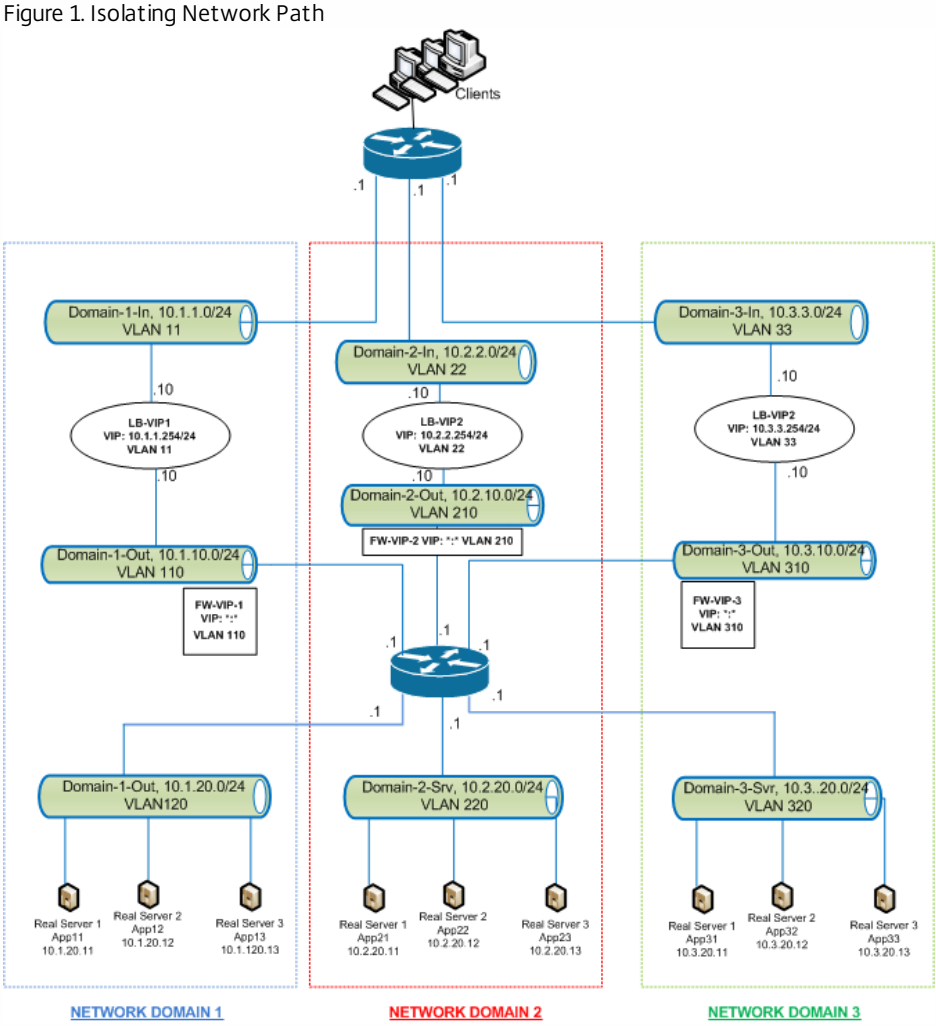
How Network Paths Are Isolated

The following figure shows a typical traffic flow across domains. Consider the traffic flow within Network Domain 1, and between Network

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Domain 1 and Network Domain 2.

Figure 1. Isolating Network Path



Traffic within Network Domain 1

Network Domain 1 has three VLANs: VLAN 11, VLAN110, and VLAN120. The following steps describe the traffic flow.

- A client from VLAN 11 sends a request for a service available from the service pool in VLAN 120.
- The load balancing virtual server LB-VIP1, which is configured to listen to traffic from VLAN 11, receives the request and forwards the request to VLAN 110. The virtual server in VLAN 110 forwards the request to shadow load balancing virtual server FW-VIP-1.
- FW-VIP-1, which is configured to listen to traffic from VLAN 110, receives the request and forwards it to VLAN 120.
- The load balancing virtual server in VLAN 120 load balances the request to one of the physical servers, App11, App12, or App13.
- The response sent by the physical server returns by the same path to the client in VLAN 11.

This configuration ensures that traffic is always segregated inside the NetScaler for all the traffic that originates from a client.

Traffic between Network Domain 1 and Network Domain 2

Network Domain 1 has three VLANs: VLAN 11, VLAN 110, and VLAN 120. Network Domain 2 also has three VLANS: VLAN 22, VLAN 210, and VLAN 220. The following steps describe the traffic flow from VALN 11 to VLAN 22.

- A client from VLAN 11, which belongs to Network Domain 1, sends a request for a service available from the service pool in VLAN 220, which belongs to the Network Domain 2.
- In Network Domain 1, the load balancing virtual server LB-VIP1, which is configured to listen to traffic from VLAN 11, receives the request and forwards the request to VLAN 110.
- Shadow load balancing virtual server FW-VIP-1, which is configured to listen to VLAN 110 traffic destined to any other domain, receives the request and forwards it to firewall virtual server FW-VIP-2 because the request is destined to a physical server in Network Domain 2.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- In Network Domain 2, FW-VIP-2 forwards the request to VLAN 220.
- The load balancing virtual server in VLAN 220 load balances the request to one of the physical servers, App21, App22, or App23.
- The response sent by the physical server returns by the same path through the firewall in Network Domain 2 and then to Network Domain 1 to reach the client in VLAN 11.

Configuration Steps

To configure network path isolation by using listen policies, do the following:

- Add listen policy expressions. Each expression specifies a domain to which traffic is destined. You can use the VLAN ID or other parameters to identify the traffic.
- For each network domain, configure two virtual servers as follows:
  - Create a load balancing virtual server for which you specify a listen policy that identifies the traffic destined for this domain. You can specify the name of an expression created earlier, or you can create a new expression while creating the virtual server.
  - Create another load balancing virtual server, referred to as shadow virtual server, for which you specify a listen policy expression that applies to traffic destined for any domain. On this virtual server, set the service type to ANY and the IP address and port to an asterisk (*). Enable MAC-based forwarding on this virtual server.
  - Enable the L2 Connection option on both the virtual servers.
    Generally, to identify a connection, the NetScaler uses the 4-tuple of client IP address, client port, destination IP address, and destination port. When you enable the L2 Connection option, the Layer 2 parameters of the connection (channel number, MAC address, and VLAN ID) are used in addition to the normal 4-tuple.
- Add services representing the server pools in the domain, and bind them to the virtual server.
- Configure the firewall for each domain as a service, and bind all of the firewall services to the shadow virtual server.

To isolate network traffic by using the command line interface

At the command prompt, type the following commands:

- add policy expression <expressionName> <listenPolicyExpression>
- add lb vserver <name> <serviceType> <ip> <port> -l2conn ON -listenPolicy <expressionName>
  Add a load balancing virtual server for each domain. This virtual server is for traffic of the same domain.
- add lb vserver <name> ANY * * -l2conn ON -m MAC -listenPolicy <expressionName>
  Add a shadow load balancing virtual server for each domain. This virtual server is for traffic of other domains.

Example

```
add policy expression e110 client.vlan.id==110
add policy expression e210 client.vlan.id==210
add policy expression e310 client.vlan.id==310
add policy expression e11 client.vlan.id==11
add policy expression e22 client.vlan.id==22
add policy expression e33 client.vlan.id==33

add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -persistenceType NONE -listenPolicy e11
-cltTimeout 180 -l2Conn ON

add lb vserver LB-VIP2 HTTP 10.2.2.254 80 -persistenceType NONE - listenPolicy e22
-cltTimeout 180 -l2Conn ON

add lb vserver LB-VIP3 HTTP 10.3.3.254 80 -persistenceType NONE - listenPolicy e33
-cltTimeout 180 -l2Conn ON


add lb vserver FW-VIP-1 ANY * * -persistenceType NONE -lbMethod ROUNDROBIN - listenPolicy e110 -Listenpriority 1 -m MAC -cltTimeout 120
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add lb vserver FW-VIP-2 ANY * * -persistenceType NONE -lbMethod ROUNDROBIN - listenPolicy e210 -Listenpriority 2 -m MAC -cltTimeout 120

add lb vserver FW-VIP-3 ANY * * -persistenceType NONE -lbMethod ROUNDROBIN - listenPolicy e310 -Listenpriority 3 -m MAC -cltTimeout 120


add service RD-1 10.1.1.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED
-usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA NO -TCPB NO -CMP NO

add service RD-2 10.2.2.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED
-usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA NO -TCPB NO -CMP NO

add service RD-3 10.3.3.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED
-usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA NO -TCPB NO -CMP NO


bind lb vserver FW-VIP-1 RD-1

bind lb vserver FW-VIP-2 RD-2

bind lb vserver FW-VIP-3 RD-3
```

To isolate network traffic by using the configuration utility

1. Add services representing the servers, as described in the "Creating a Service" section in Setting Up Basic Load Balancing.
2. Add each firewall as a service:
   1. Navigate to Traffic Management > Load Balancing > Services
   2. Create a service, specifying protocol as ANY, server as firewall's IP address, and port as 80.
3. Configure a load balancing virtual server.
4. Configure the shadow load balancing virtual server.
5. For each network domain, repeat steps 3 and 4.
6. From the Load Balancing Virtual Servers pane, open the virtual servers that you created and verify the settings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 12: Configuring XenDesktop for Load Balancing

Dec 27, 2016

For an improved performance in the delivery of virtual desktop applications, you can integrate the NetScaler appliance with Citrix XenDesktop and use the NetScaler load balancing feature to distribute the load across the Web Interface servers and the Desktop Delivery Controller (DDC) servers.

Generally, you use XenDesktop in situations where applications are not compatible with running on a terminal server or XenApp, or if each virtual desktop has unique requirements. In such cases, you need one desktop host for each user that connects. However, the hosts can be pooled so that you need only one host for each currently connected user.

The core application service deployed for XenDesktop is the Desktop Delivery Controller (DDC). The DDC is installed on a server, and its main function is to register desktop hosts and broker client connections to them.

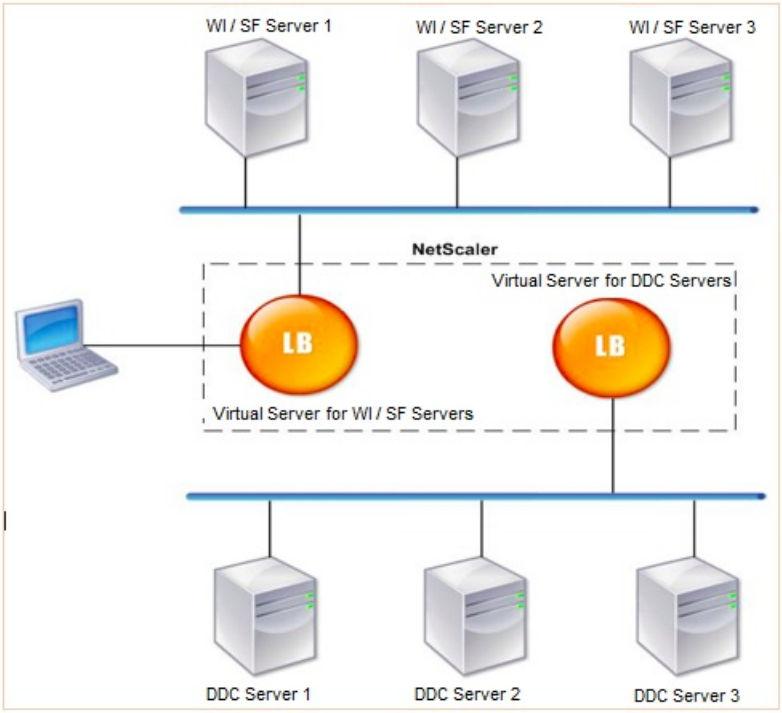
The DDC also authenticates users and manages the assembly of the users' virtual desktop environments by controlling the state of the desktops, and starting and stopping the desktops.

Generally, multiple DDCs are installed to enhance availability.

The Web Interface servers provide secure access to virtual desktops. The Web Interface is the initial connection portal to the Desktop Delivery Controller (DDC). The Web browser on the user's device sends information to the Web server, which communicates with the server farm to provide the user with access to the virtual desktop.

The following figure shows the topology of a NetScaler appliance working with XenDesktop.

Figure 1. **Load Balancing of XenDesktop**



© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Although you can use the HTTP protocol, Citrix recommends that you use SSL for communication between the client and the NetScaler. You can use the HTTP protocol for communication between the NetScaler and the DDC servers even though you use the SSL protocol for communication with the client.

To configure load balancing for XenDesktop by using the NetScaler GUI

1. Create a service.
   1. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** and click **Add**.
   2. Create a service by specifying a name, an IP address, a port, and a protocol type and then click **OK**.
2. Create a load balancing virtual server.
   1. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and click **Add**.
   2. Create a virtual server by specifying a name, an IP address, a port, and a protocol type and then click **OK**.
3. Bind the service to the load balancing virtual server.
4. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and select a server.
   1. Click **Edit**.
   2. In the **Services and Service Groups**, click **>** and click **Add Binding**.
   3. Select the service you want to bind and enter the weight value.
   4. Click **Bind**.

To configure load balancing for XenDesktop by using the command line interface

- To create a service, at the command prompt, type:

  add service <name> <serverName> <serviceType> <port>

  Example

  add service Service-HTTP-1 192.0.2.5 HTTP 80

- To create a virtual server, at the command prompt, type:

  add lb vserver <name> <serviceType> <ip> <port>

  Example

  add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

- To bind a service to a load balancing virtual server, at the command prompt, type:

  bind lb vserver <name> <serviceName>

  Example

  bind lb vserver Vserver-LB-1 Service-HTTP-1

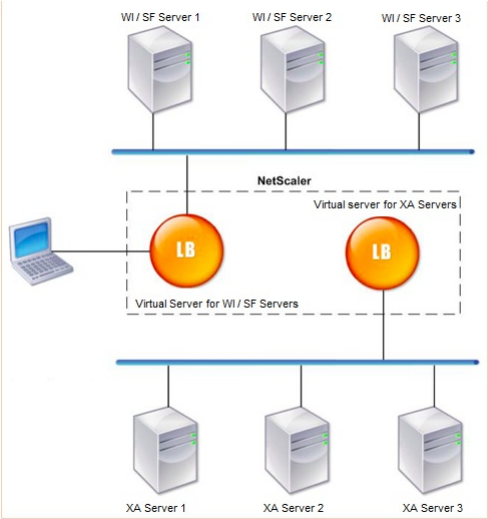
© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 13: Configuring XenApp for Load Balancing

Dec 27, 2016

For efficient delivery of applications, you can integrate the NetScaler appliance with Citrix XenApp and use the NetScaler load balancing feature to distribute the load across the XenApp server farms. The following figure is a topology diagram of such a setup.

Figure 1. **Load Balancing of XenApp**



The Web Interface servers provide secure access to XenApp application resources through the user's Web browser. The Web Interface client presents to the users all the resources, such as applications, content, and desktops that are made available in the XenApp server farms. Users can access the published resources through a standard Web browser or through the Citrix online plug-in.

The Web browser on the user's device sends information to the Web server, which communicates with the servers on the server farm to provide the user with access to the resources.

The Web Interface and the XML Broker are complementary services. The Web Interface provides users with access to applications, and the XML Broker evaluates the user's permissions to determine which applications appear in the Web Interface.

The XML service is installed on all the servers in the server farm. The XML service specified in the Web Interface functions as an XML broker. On the basis of the user credentials passed by the Web Interface server, the XML Broker server sends a list of applications accessible to the user.

In large enterprises where multiple Web Interface servers and XML Broker servers are deployed, Citrix recommends load balancing these servers by using NetScaler. Configure one virtual server to load balance all of the Web Interface servers and another for all of the XML Broker servers. The load balancing method and other features can be configured on the virtual server as required.

The Web Interface servers provide secure access to XenApp application resources through the user's Web browser. The Web Interface client presents to the users all the resources, such as application

The Web browser on the user's device sends information to the Web server, which communicates with the servers on the server farm to provide the user with access to the resources.

The Web Interface and the XML Broker are complementary services. The Web Interface provides users with access to applications, and the XML Broker evaluates the user's permissions to determine v

The XML service is installed on all the servers in the server farm. The XML service specified in the Web Interface functions as an XML broker. On the basis of the user credentials passed by the Web Inte

In large enterprises where multiple Web Interface servers and XML Broker servers are deployed, Citrix recommends load balancing these servers by using NetScaler. Configure one virtual server to load b

Note: Although you can use the HTTP protocol, Citrix recommends that you use SSL for communication between the client and the NetScaler. You can use the HTTP protocol for communication bet

The Web Interface servers provide secure access to XenApp application resources through the user's Web browser. The Web Interface client presents to the users all the resources, such as application

The Web browser on the user's device sends information to the Web server, which communicates with the servers on the server farm to provide the user with access to the resources.

The Web Interface and the XML Broker are complementary services. The Web Interface provides users with access to applications, and the XML Broker evaluates the user's permissions to determine v

The XML service is installed on all the servers in the server farm. The XML service specified in the Web Interface functions as an XML broker. On the basis of the user credentials passed by the Web Inte

In large enterprises where multiple Web Interface servers and XML Broker servers are deployed, Citrix recommends load balancing these servers by using NetScaler. Configure one virtual server to load b

Note: Although you can use the HTTP protocol, Citrix recommends that you use SSL for communication between the client and the NetScaler. You can use the HTTP protocol for communication bet

> **Note**
>
> Although you can use the HTTP protocol, Citrix recommends that you use SSL for communication between the client and the NetScaler. You can use the HTTP protocol for communication between the NetScaler and the WI servers even though you use the SSL protocol for communication with the client.

To configure load balancing for XenApp by using the NetScaler GUI

1. Create a service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** and click **Add**.
2. Create a service by specifying a name, an IP address, a port, and a protocol type and then click **OK**.

2. Create a load balancing virtual server.
   1. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and click **Add**.
   2. Create a virtual server by specifying a name, an IP address, a port, and a protocol type and then click **OK**.
3. Bind the service to the load balancing virtual server.
4. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and select a server.
   1. Click **Edit**.
   2. In the S**ervices and Service Groups,** click **>** and click **Add Binding**.
   3. Select the service you want to bind and enter the weight value.
   4. Click **Bind**.

To configure load balancing for XenApp by using the command line interface

- To create a service, at the command prompt, type:

  add service <name> <serverName> <serviceType> <port>

  Example

  add service Service-HTTP-1 192.0.2.5 HTTP 80

- To create a virtual server, at the command prompt, type:

  add lb vserver <name> <serviceType> <ip> <port>

  Example

  add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

- To bind a service to a load balancing virtual server, at the command prompt, type:

  bind lb vserver <name> <serviceName>

  Example

  bind lb vserver Vserver-LB-1 Service-HTTP-1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

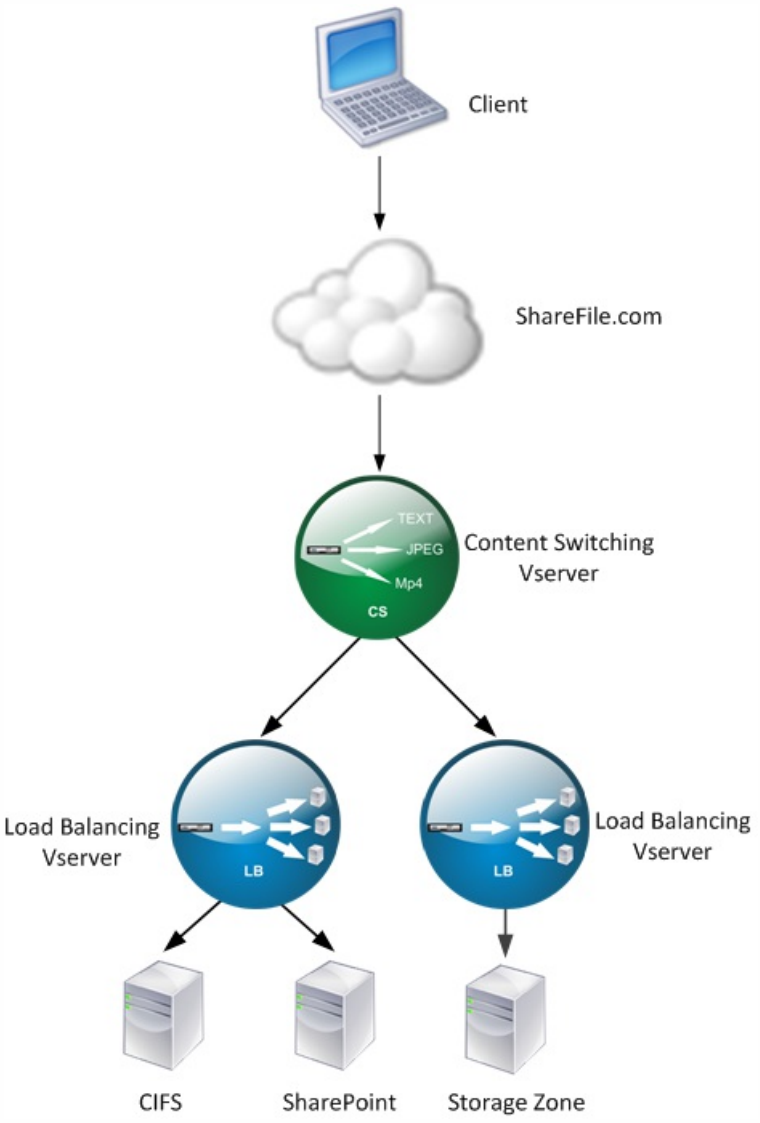
Use Case 14: ShareFile Wizard for Load Balancing Citrix ShareFile

Jun 08, 2015

You can configure load balancing for Citrix ShareFile using the wizard. The Citrix ShareFile wizard helps in setting up load balancing configuration for ShareFile site based on the type of content requested. The content switching server directs the request based on whether it is a StorageZone, CIFS or a SharePoint request. The content switching is based on policies. The wizard auto generates the policies to identify whether the request is for StorageZone, CIFS or SharePoint. The content switching virtual server uses these policies to direct the request to the correct load balancing server.

A typical data flow can be depicted as shown in the diagram below.

Figure 1. ShareFile Data Load Balancing



You can view the load balancing virtual servers that the ShareFile wizard creates by navigating to Traffic Management >Virtual Servers and Services > Virtual Servers. You cannot manually remove the virtual servers created using the ShareFile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

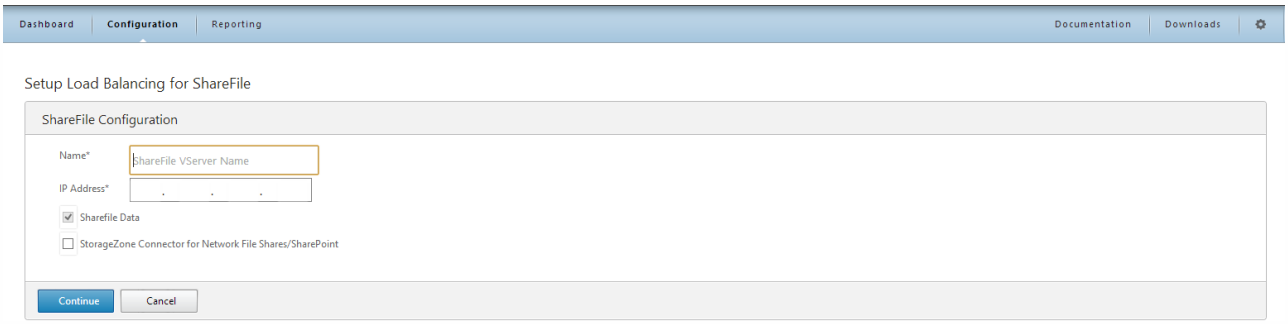
wizard. Use the wizard to remove the virtual servers.

NetScaler uses the LDAP authentication for SharePoint or CIFS request. Hash authentication is used for authenticating requests for StorageZones.

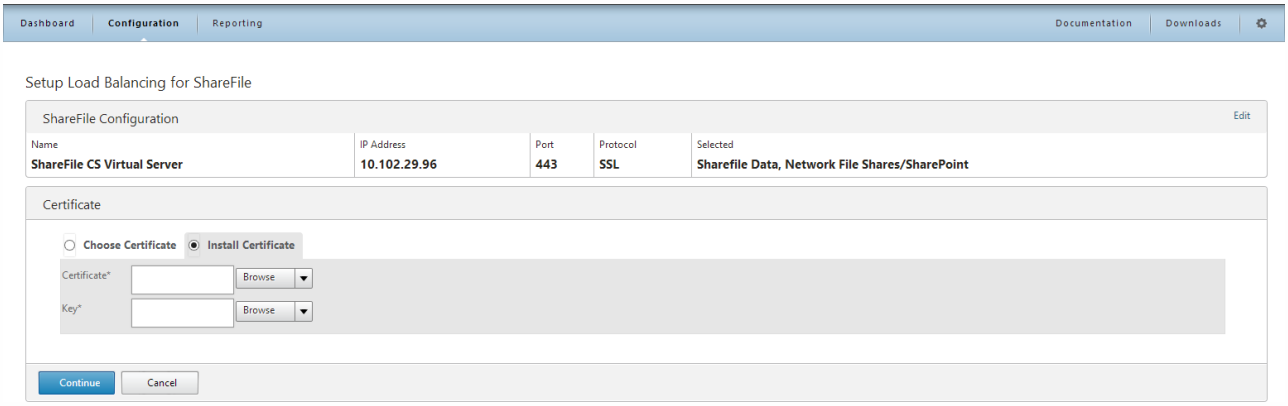
To configure a NetScaler appliance for load balancing Citrix ShareFile

Updated: 2015-06-04

1. In the navigation pane, click Load Balancing.
2. Navigate to Traffic Management > Load Balancing.
3. Under Citrix ShareFile, click Setup NetScaler for ShareFile.
4. On the Setup Load Balancing for ShareFile page, provide the following information:
   - Name: Name of the content switching virtual server.
   - IP Address: IP address of the content switching virtual server.
   - If you want to setup load balancing for CIFS or SharePoint, click the StorageZone Connector for Network File Shares/SharePoint checkbox and then click Continue. By default ShareFile Data checkbox is selected.



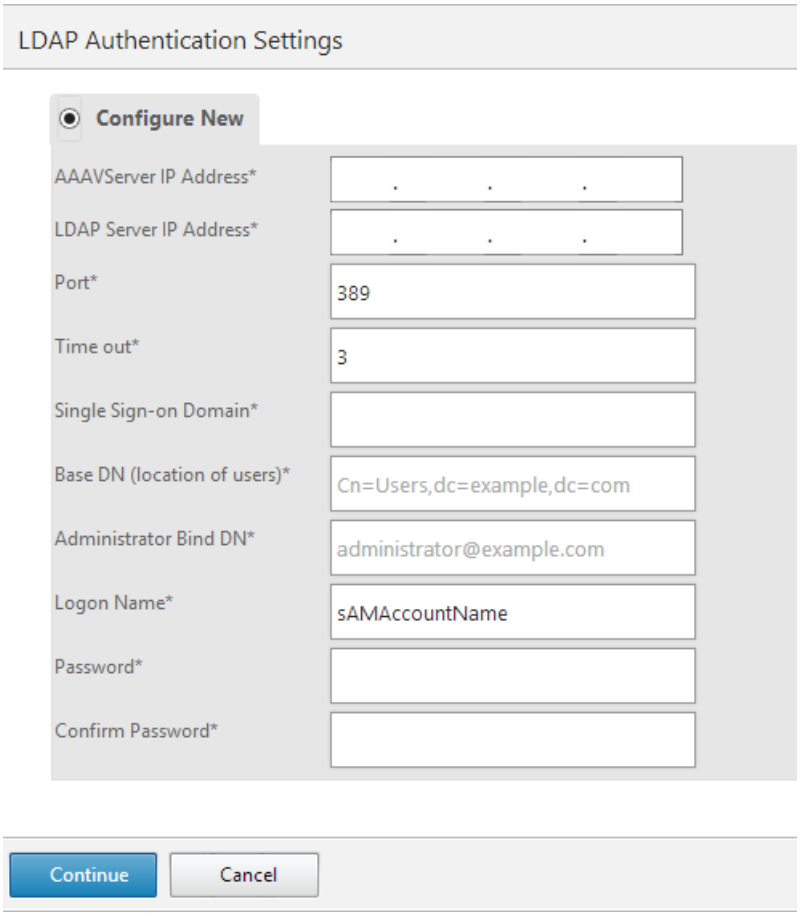
5. Provide a valid certificate. If you have a certificate, click Choose Certificate and from the drop-down list select the certificate. If you have to install a certificate, click Install Certificate and provide the Certificate-Key pair.



6. Click Continue.
7. In the Add New StorageZone Controller dialog box, specify the values of the following parameters:
   - StorageZone Controller IP Address— IP address
   - Port— Port number. The default value is 443.
   - Protocol— Select from HTTPS or HTTP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

8. Click Create and then click Done. The wizard automatically creates a service and autogenerate the name of the service.
9. If you chose load balancing for CIFS or SharePoint in step 4.c, then specify the values for LDAP Authentication Settings:
   - AAAVServer IP Address— IP address of AAA virtual server
   - LDAP Server IP Address— IP Address of the LDAP server
   - Port— Port number. The default value is 389
   - Time out— The time out value in minutes
   - Single Sign-on Domain— Single sign-on domain name
   - Base DN— Base domain name
   - Administrator Bind DN— LDAP account name with the domain name, for example, adminstrator@domainname.com
   - Logon Name— Logon name is the sAMAccount name
   - Password and Confirm Password— Enter the password and confirm the password



10. Click Continue and then click Done.

To remove load balancing configuration for ShareFile

1. Click on Configuration > Traffic Management > Load Balancing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. On the Load Balancing page, under Citrix ShareFile click on Remove ShareFile Configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Aug 01, 2013

If the load balancing does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for Troubleshooting Load Balancing

Updated: 2013-08-01

For best results, use the following resources to troubleshoot a content switching issue on a NetScaler appliance:

- Latest ns.conf file
- Relevant newnslog files
- Ethereal packet traces recorded on the appliance and relevant client, if possible
- The ns.log file

In addition to the above resources, the following tools expedite troubleshooting:

- A browser add-on tool that can display HTTP headers. This can be used to troubleshoot persistency related issues.
- The Wireshark application customized for the NetScaler trace files.

Troubleshooting Load Balancing Issues

Updated: 2015-06-11

- **Issue**

  CPU usage reaches 100% when a user monitor is bound to a service that is bound to a virtual server on which -m MAC option is enabled.

- **Resolution**

  Bind a non-user monitor to the service.

- **Issue**
  I created a user script for monitoring, but it is not working.

  Resolution

  Check the number of arguments in the script. The limit is 512. A script with more than 512 arguments might not work properly. Use the nsumon-debug.pl script from the NetScaler command line to debug the script.

- **Issue**
  I see a lot of monitor probes, and they seem to be increasing the network traffic unnecessarily. Is there a way to turn off the monitor probes?

  Resolution

  You can turn off the monitor probe connections, by disabling the monitor or setting the value of the healthMonitor parameter in the set service command to NO. With the NO option, the appliance shows the service as UP at all times.

- **Issue**
  I have set up monitors for services, but connections are still directed to servers that are DOWN.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Resolution

You probably need to decrease the monitor probe intervals. The NetScaler appliance does not detect the DOWN state until the monitor sends a probe.

- **Issue**
  A metric bound to the monitor is present in the local and custom metric tables.

  Resolution

  Add the local prefix to the metric name if the metric is chosen from the local metric table. However, if the metric is chosen from the custom table, you don't need to add any prefix.

- **Issue**
  The monitor probes to a service are not reaching the service.

  Resolution

  Check whether you have set a limit on the number of connections for a service. If yes, exempt monitor-probe connections from this limit by setting the monitorSkipMaxClient parameter to ENABLED.

- **Issue**
  I am able to ping the servers, but the state of the services is always shown as DOWN.

  Resolution

  Check the type of monitors configured. For example, if a server is not configured for SSL and you use an HTTPS monitor, the state of the service is marked as DOWN. In this case using a TCP monitor should change the state of the service to UP.

- **Issue**
  Setting a weight for load monitors does not help in deciding the state of the service.

  Resolution

  Load monitors cannot decide the state of the service. Therefore, setting a weight on the load monitors is inappropriate.

- **Issue**
  A service is not stable.

  Resolution

  Consider troubleshooting the following components:
  - Verify that a correct server is bound to the service.
  - Verify the type of monitor bound to the service.
  - Verify the reasons for the monitor failures. You can open service from the Services page and verify the details for the number of probes, failures, and last response status for the monitor in the Monitors tab of the Configure service dialog box. To display the details, click the monitor configured.
  - If it is a custom monitor, bind a TCP or ping monitor to the service and verify the status of the monitor. If this resolves the issue, there is some problem with the custom monitor and the monitor requires further investigation.
  - You can record packet traces on the NetScaler appliance and verify the monitor probes and server response for

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

further investigation.

- **Issue**
  The virtual IP (VIP) address is not stable or its status is displayed as DOWN.

  Resolution

  Consider troubleshooting the following components:
  - Verify that the load balancing feature is licensed.
  - Verify that the feature is enabled.
  - Verify that an appropriate service is bound to the virtual server.
  - If the status of the VIP address is displayed as DOWN, verify that an administrator has enabled the service. If it is not, the status of the service should be Out-Of-Service. In such as case, you must enable the service and verify if the issue is resolved.
  - Verify the service(s) bound to the virtual server and complete the troubleshooting steps mentioned for service not stable issue.
  - If the VIP address is not stable, all the services bound to the virtual server should fail. Therefore, verify if all the services are failing at the same time. If it is so, there is a network issue between the NetScaler appliance and the servers.

- **Issue**
  The site is experiencing uneven load balancing.

  Resolution

  Consider troubleshooting the following components:
  - Verify the load balancing method configured on the appliance.
  - Verify weights associated with the services are as expected.
  - If the load balancing method is other than round robin, verify the number of connections to the server logged in the newnslog file. You can run the following command to verify the number on the newnslog file:
    # nsconmsg –K <newnslog_file> -s ConLb=2 –d oldconmsg

    Verify the services for the specific virtual server and check for the Response time, Open Established connections (OE), Hits, Persistent Hits and persistent rate (P) to troubleshoot the issue further.

  - If the load balancing method is round robin, verify the persistent Hits as mentioned in the preceding step. Additionally, verify if the service is not stable. If it is not, complete the troubleshooting steps mentioned for service not stable issue
  - Verify if persistency is configured on the appliance.
  - Verify is any service is not stable. If yes, complete the troubleshooting steps mentioned for service not stable issue.

- **Issue**
  The service status is displayed as DOWN.

  Resolution

  Consider troubleshooting the following components:
  - Verify whether a SNIP or MIP address is configured.
  - Verify that appropriate monitors are bound to the service.
  - If custom monitors are bound to the service, bind a TCP or ping monitor to the service and verify the status of the monitor. If this resolves the issue, there is some problem with the custom monitor and the monitor requires further investigation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Verify if the status of service is displayed as DOWN for the server that is in another subnet. If yes, verify if Use Subnet IP (USNIP) resolves the issue because this could be due to the MIP address being unable to communicate to the server.

- **Issue**
  There is an issue with the response time.

  Resolution

  Consider troubleshooting the following components:
  - Verify the server response time from the service stats either by running the following command:
    # nsconmsg –K <newnslog_file> -s ConLb=2 –d oldconmsg
  - Check for service not stable and service status being displayed as DOWN issues.

- **Issue**
  One of the servers is serving more requests than the other load balanced servers.

  Resolution

  Consider troubleshooting the following components:

  - Verify the load balancing method. Use the round robin method to distribute the client request equally regardless of the load on the servers.
  - Determine whether persistence is enabled for the load balancing configuration. If persistence is enabled, a given servers might be carrying a heavier load to maintain its session, especially If the persistence sessions are long.
  - Verify whether weights are assigned to each service. Assigning proper weights helps in proper load distribution.

- **Issue**
  Connections to a specific load balanced server are stalled. For example, all connections to one Outlook server might be stalled.

  Resolution

  Consider troubleshooting the following components:
  - Verify the load balance method. If it is round robin, consider changing the method to least connections.
  - Consider reducing the monitor time-out period. A shorter timeout period helps in marking a service as DOWN sooner, which would help in directing the traffic to server which is functional.
  - If the connections are stalled for a long period, surge-queue might build. Consider flushing the surge-queue to avoid a sudden spike in load on the server.
  - If the servers are working at their maximum level, consider adding a new server for better performance.

- **Issue**
  A majority of the connections are directed to a particular server, even when the least connections method for load balancing is configured.

  Resolution

  Determine whether persistence is configured and is of type source IP. If source IP persistence is configured even with the least connections method, the requests go to a specific server. The server's IP address is required for maintaining the session information. Consider using HTTP Cookies based persistence.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Troubleshooting Tips**

  For other issues, consider following tips to troubleshoot an issue not listed above:
  - If multiple load monitors are bound to a service, the load on the service is the sum of all the values on the load monitors bound to it. For load balancing to work properly, you must bind the same set of monitors to all the services.
  - If you disable a load monitor bound to the service and the service is bound to a virtual server, the virtual server uses the round robin method for load balancing.
  - When you bind a service to a virtual server where the load balancing method is CUSTOMLOAD and the service status is UP, the virtual server uses the initial round robin method for load balancing. It continues to be in round robin if the service has no custom load monitors, or if status of at least one of the custom load monitors is not UP.
  - All the services that are bound to a virtual server where the load balancing method is CUSTOMLOAD, the services must have load monitors bound to them.
  - The CUSTOMLOAD load balancing method also follows startup round robin.
  - If you disable a metric-based binding and this is the last active metric, the specific virtual server uses the round robin method for load balancing. A metric is disabled by setting the metric threshold to zero.
  - When a metric bound to a monitor crosses the threshold value, that particular service is not considered for load balancing. If all the services have reached the threshold, the virtual server uses the round robin method for load balancing and an error message “5xx - server busy error” is displayed.
  - A maximum of 10 metrics from a custom table can be bound to the monitor.
  - The OIDs must be scalar variables.
  - For successful load balancing, the interval must be as low as possible. If the interval is high, the time period for retrieving the load value increases. As a result, load balancing takes place using improper values.
  - A user cannot modify the local table.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing

Feb 26, 2014

What are the various load balancing policies I can create on the NetScaler appliance?

You can create the following types of load balancing policies on the NetScaler appliance:

- Least Connections
- Round Robin
- Least response time
- Least bandwidth
- Least packets
- URL hashing
- Domain name hashing
- Source IP address hashing
- Destination IP address hashing
- Source IP - Destination IP hashing
- Token
- LRTM

Can I achieve the Web farm security by implementing load balancing using the NetScaler appliance?

Yes. You can achieve Web farm security by implementing load balancing using the NetScaler appliance. NetScaler appliance enables you to implement the following options of the load balancing feature:

- IP Address hiding: Enables you to install the actual servers to be on private IP address space for security reasons and for IP address conservation. This process is transparent to the end-user because the NetScaler appliance accepts requests on behalf of the server. While in the address hiding mode, the appliance completely isolates the two networks. Therefore, a client can access a service running on the private subnet, such as FTP or a Telnet server, through a different VIP on the appliance for that service.
- Port Mapping: Enables the actual TCP services to be hosted on non-standard ports for security reasons. This process is transparent to the end-user as the NetScaler appliance accepts requests on behalf of the server on the standard advertised IP address and port number.

What are various devices that I can use to load balance with a NetScaler appliance?

You can load balance following devices with a NetScaler appliance:

- Server farms
- Caches or Reverse Proxies
- Firewall devices
- Intrusion detection systems
- SSL offload devices
- Compression devices
- Content Inspection servers

Why should I implement the load balancing feature for the website?

You can implement the Load balancing feature for the website to take the following advantages:

- Reduce the response time: When you implement the load balancing feature for the website, one of the major benefits is the boost you can look forward to in load time. With two or more servers sharing the load of the web traffic, each of the servers runs less traffic load than a single server alone. This means there are more resources available to fulfill the client requests. This results in a faster website.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Redundancy: Implementing the load balancing feature introduces a bit of redundancy. For example, if the website is balanced across three servers and one of them does not respond at all, the other two can keep running and the website visitors do not even notice any downtime. Any load balancing solution immediately stops sending traffic to the backend server that is not available.

Why do I need to disable the Mac Based Forwarding (MBF) option for Link Load Balancing (LLB)?

- If you enable the MBF option, the NetScaler appliance considers that the incoming traffic from the client and the outgoing traffic to the same client flow through the same upstream router. However, the LLB feature requires that the best path be chosen for the return traffic.
- Enabling the MBF option breaks this topology design by sending the outgoing traffic through the router that forwarded the incoming client traffic.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Offload and Acceleration

Dec 08, 2014

A Citrix® NetScaler® appliance configured for SSL acceleration transparently accelerates SSL transactions by offloading SSL processing from the server. To configure SSL offloading, you configure a virtual server to intercept and process SSL transactions, and send the decrypted traffic to the server (unless you configure end-to-end encryption, in which case the traffic is re-encrypted). Upon receiving the response from the server, the appliance completes the secure transaction with the client. From the client's perspective, the transaction seems to be directly with the server. A NetScaler configured for SSL acceleration also performs other configured functions, such as load balancing.

Configuring SSL offloading requires an SSL certificate and key pair, which you must obtain if you do not already have an SSL certificate. Other SSL-related tasks that you might need to perform include managing certificates, managing certificate revocation lists, configuring client authentication, and managing SSL actions and policies.

A non-FIPS NetScaler appliance stores the server's private key on the hard disk. On a FIPS appliance, the key is stored in a cryptographic module known as a hardware security module (HSM). Only the MPX 9700/10500/12500/15500 appliances support a FIPS card, so other NetScaler models cannot be equipped with an HSM.

Beginning with release 10.5, build 52.1115.e, all NetScaler appliances that do not support a FIPS card (including virtual appliances) support the Thales nShield® Connect external HSM. (MPX 9700/10500/12500/15500 appliances do not support an external HSM.)

Note: FIPS-related options for some of the SSL configuration procedures described in this document are specific to a FIPS-enabled NetScaler.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SSL Offloading

Jun 03, 2015

To configure SSL offloading, you must enable SSL processing on the NetScaler appliance and configure an SSL based virtual server that will intercept SSL traffic, decrypt the traffic, and forward it to a service that is bound to the virtual server. To secure time-sensitive traffic, such as media streaming, you can configure a DTLS virtual server.To enable SSL offloading, you must import a valid certificate and key and bind the pair to the virtual server.

To configure SSL offloading, see the following sections:

- Enabling SSL Processing
- Configuring Services
- Configuring an SSL-Based Virtual Server
- Binding Services to the SSL-Based Virtual Server
- Adding or Updating a Certificate-Key Pair
- Binding the Certificate-Key Pair to the SSL-Based Virtual Server
- SSL Profiles
- Configuring an SSL Virtual Server for Secure Hosting of Multiple Sites
- Configuring a DTLS Virtual Server
- DTLS Profile
- Importing SSL Files from Remote Hosts
- SSL Profiles
- Enabling Stricter Control on Client Certificate Validation
- Graceful Cleanup of SSL Sessions

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling SSL Processing

Nov 11, 2013

To process SSL traffic, you must enable SSL processing. You can configure SSL based entities, such as virtual servers and services, without enabling SSL processing, but they will not work until SSL processing is enabled.

To enable SSL processing by using the command line interface

At the command prompt, type:

- enable ns feature ssl
- show ns feature

Example

```
> enable ns feature SSL
 Done
> show ns feature

     Feature              Acronym        Status
     -------              -------        ------
 1)  Web Logging          WL             OFF
 2)  Surge Protection     SP             ON
 3)  Load Balancing       LB             ON
 .
 .
 .
 9)  SSL Offloading       SSL            ON
 .
 .
 .
 24) NetScaler Push       push           OFF
 Done
```

To enable SSL processing by using the configuration utility

Navigate to System > Settings and, in the Modes and Features group, select Configure Basic Features, and select SSL Offloading.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Services

Oct 12, 2016

On the NetScaler appliance, a service represents a physical server or an application on a physical server. Once configured, services are in the disabled state until the appliance can reach the physical server on the network and monitor its status.

To add a service by using the command line interface

At the command prompt, type the following commands to add a service and verify the configuration:

- **add service** <name> (<IP> | <serverName>) <serviceType> <port>
- **show service** <serviceName>

Example COPY

```
> add service SSL1 192.168.0.12 SSL 443

 Done

> sh service SSL1

        SSL1 (192.168.0.12:443) - SSL

        State: DOWN

        Last state change was at Thu Oct  6 17:15:41 2016

        Time since last state change: 0 days, 00:00:07.990

        Server Name: 192.168.0.12

        Server ID : None        Monitor Threshold : 0

        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits

        Use Source IP: NO

        Client Keepalive(CKA): NO

        Access Down Service: NO

        TCP Buffering(TCPB): NO
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        HTTP Compression(CMP): NO

        Idle timeout: Client: 180 sec   Server: 360 sec

        Client IP: DISABLED

        Cacheable: NO

        SC: OFF

        SP: ON

        Down state flush: ENABLED

        Monitor Connection Close : NONE

        Appflow logging: ENABLED

        Process Local: DISABLED

        Traffic Domain: 0



1)      Monitor Name: tcp-default

            State: DOWN     Weight: 1       Passive: 0

            Probes: 2       Failed [Total: 1 Current: 1]

            Last response: Failure - Time out during TCP connection establishment stage

            Response Time: 0.0 millisec

 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To modify or remove a service by using the command line interface

To modify a service, use the set service command, which is just like using the add service command, except that you enter the name of an existing service. To remove a service, use the rm service command, which accepts only the <name> argument.

To configure a service by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, create a service, and specify the protocol as SSL.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring an SSL-Based Virtual Server

Jan 01, 2018

Secure sessions require establishing a connection between the client and an SSL-based virtual server on the NetScaler appliance. The SSL virtual server intercepts SSL traffic, decrypts it and processes it before sending it to services that are bound to the virtual server.

Note: The SSL virtual server is marked as down on the NetScaler appliance until a valid certificate / key pair and at least one service are bound to it. An SSL based virtual server is a load balancing virtual server of protocol type SSL or SSL_TCP. The load balancing feature must be enabled on the NetScaler.

To add an SSL-based virtual server by using the command line interface

At the command prompt, type the following commands to create an SSL-based virtual server and verify the configuration:

- add lb vserver <name> (serviceType) <IPAddress> <port>
- show lb vserver <name>

Example

```
> add lb vserver vssl  SSL 10.102.29.133 443
 Done
> show ssl vserver vssl

      Advanced SSL configuration for VServer vssl:
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 0
      Session Reuse: ENABLED        Timeout: 120 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: DISABLED
      ClearText Port: 0
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)    Cipher Name: DEFAULT
      Description: Predefined Cipher Alias
 Done
```

To modify or remove an SSL-based virtual server by using the command line interface

To modify the load balancing properties of an SSL virtual server, use the set lb vserver command, which is just like using the add lb vserver command, except that you enter the name of an existing vserver. To modify the SSL properties of an SSL-based virtual server, use the set ssl vserver command. For more information, see Customizing the SSL Configuration.

To remove an SSL virtual server, use the rm lb vserver command, which accepts only the <name> argument.

To configure an SSL-based virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers, create a virtual server, and specify the protocol as SSL.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding Services to the SSL-Based Virtual Server

Jan 01, 2018

For the NetScaler appliance to forward decrypted SSL data to servers in the network, services representing these physical servers must be bound to the virtual server that receives the SSL data.

Because the link between the NetScaler and the physical server is typically secure, data transfer between the appliance and the physical server does not have to be encrypted. However, you can provide end-to end-encryption by encrypting data transfer between the NetScaler and the server. For details, see Use Case 1: Configuring SSL Offloading with End-to-End Encryption.

Note: The Load Balancing feature should be enabled on the NetScaler appliance before you bind services to the SSL based virtual server.

To bind a service to a virtual server by using the command line interface

At the command prompt, type the following commands to bind the service to the virtual server and verify the configuration:

- bind lb vserver <name> <serviceName>
- show lb vserver <name>

Example

```
> bind lb vserver vssl ssl1
 Done
> show lb vserver vssl
      vssl (10.102.29.133:443) - SSL  Type: ADDRESS
      State: DOWN[Certkey not bound]
      Last state change was at Thu Nov 12 05:31:17 2009 (+485 ms)
      Time since last state change: 0 days, 00:08:52.130
      Effective State: DOWN
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      No. of Bound Services :  1 (Total)       1 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Vserver IP and Port insertion: OFF
      Push: DISABLED  Push VServer:
      Push Multi Clients: NO
      Push Label Rule: none

1) ssl1 (10.102.29.252: 80) - HTTP State: UP    Weight: 1
 Done
```

To unbind a service from a virtual server by using the command line interface

At the command prompt, type the following command:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

unbind lb vserver <name> <serviceName>

Example

```
unbind lb vserver vssl ssl1
```

To bind a service to a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server, and click in the Services section to bind a service to the virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding or Updating a Certificate-Key Pair

Aug 08, 2016

For any SSL transaction, the server needs a valid certificate and the corresponding private and public key pair. The SSL data is encrypted with the server's public key, which is available through the server's certificate. Decryption requires the corresponding private key.

Because the NetScaler appliance offloads SSL transactions from the server, the server's certificate and private key must be present on the appliance, and the certificate must be paired with its corresponding private key. This certificate-key pair must then be bound to the virtual server that processes the SSL transactions.
Note: From release 11.0, the default certificate on a NetScaler appliance is 2048-bits. In earlier builds, the default certificate was 512-bits or 1024-bits. After upgrading to release 11.0, you must delete all your old certificate-key pairs starting with "ns-", and then restart the appliance to automatically generate a 2048-bit default certificate.
Both the certificate and the key must be in local storage on the NetScaler appliance before they can be added to the appliance. If your certificate or key file is not on the appliance, upload it to the appliance before you create the pair.
Important: Certificates and keys are stored in the /nsconfig/ssl directory by default. If your certificates or keys are stored in any other location, you must provide the absolute path to the files on the NetScaler appliance. The NetScaler FIPS appliances do not support external keys (non-FIPS keys). On a FIPS appliance, you cannot load keys from a local storage device such as a hard disk or flash memory. The FIPS keys must be present in the Hardware Security Module (HSM) of the appliance.
On a NetScaler MPX appliance and a NetScaler FIPS appliance, only RSA private keys are supported. On a VPX virtual appliance, both RSA and DSA private keys are supported. On an SDX appliance if SSL chips are assigned to an instance, then only RSA private keys are supported. However, if SSL chips are not assigned to an instance, then both RSA and DSA private keys are supported. In all the cases, you can bind a CA certificate with either RSA or DSA keys.

Set the notification period and enable the expiry monitor to issue a prompt before the certificate expires.

The NetScaler appliance supports the following input formats of the certificate and the private-key files:
- PEM - Privacy Enhanced Mail
- DER - Distinguished Encoding Rule
- PFX - Personal Information Exchange

The format is automatically detected by the software. Therefore, you are no longer required to specify the format in the inform parameter. If you do specify the format (correct or incorrect), it is ignored by the software. The format of the certificate and the key file must be the same.
Note: A certificate must be signed by using one of the following hash algorithms:
- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384 (supported only on the front end)
- SHA-512 (supported only on the front end)

An MPX appliance supports certificates of 512 or more bits, up to the following sizes:
- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service
- 4096-bit CA certificate (includes intermediate and root certificates)
- 4096-bit certificate on the back-end server
- 4096-bit client certificate (if client authentication is enabled on the virtual server)

A VPX virtual appliance supports certificates of 512 or more bits, up to the following sizes:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service
- 4096-bit CA certificate (includes intermediate and root certificates)
- 2048-bit certificate on the back-end server
- 2048-bit client certificate (if client authentication is enabled on the virtual server)

> **Note**
>
> A NetScaler SDX appliance supports certificates of 512 or more bits. Each NetScaler VPX instance hosted on the appliance supports the certificate sizes listed above for a VPX virtual appliance. However, if an SSL chip is assigned to an instance, that instance supports the certificate sizes supported by an MPX appliance.

To add a certificate-key pair by using the command line interface

At the command prompt, type the following commands to add a certificate-key pair and verify the configuration:

- add ssl certKey <certkeyName> -cert <string>[(-key <string> [-password]) | -fipsKey <string>] [-inform (DER | PEM)] [<passplain>] [-expiryMonitor (ENABLED | DISABLED) [-notificationPeriod <positive_integer>]]
- show ssl certKey [<certkeyName>]

Example

```
> add ssl certKey sslckey -cert server_cert.pem -key server_key.pem -password ssl -expiryMonitor ENABLED -notificationPeriod 30
 Done
Note: For FIPS appliances, replace -key with -fipskey
> show ssl certKey sslckey
      Name: sslckey          Status: Valid,   Days to expiration:8418
      Version: 3
      Serial Number: 01
      Signature Algorithm: md5WithRSAEncryption
      Issuer:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.root.com
      Validity
            Not Before: Jul 15 02:25:01 2005 GMT
            Not After : Nov 30 02:25:01 2032 GMT
      Subject:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.server.com
      Public Key Algorithm: rsaEncryption
      Public Key size: 2048
 Done
```

To update or remove a certificate-key pair by using the command line interface

To modify the expiry monitor or notification period in a certificate-key pair, use the set ssl certkey command. To replace the certificate or key in a certificate-key pair, use the update ssl certkey command. The update ssl certkey command has an additional parameter for overriding the domain check. For both commands, enter the name of an existing certificate-key pair. To remove an SSL certificate-key pair, use the rm ssl certkey command, which accepts only the <certkeyName> argument.

To add or update a certificate-key pair by using the configuration utility

Navigate to Traffic Management > SSL > Certificates, and configure a certificate-key pair.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding the Certificate-Key Pair to the SSL-Based Virtual Server

Jan 01, 2018

An SSL certificate is an integral element of the SSL encryption and decryption process. The certificate is used during an SSL handshake to establish the identity of the SSL server.

The certificate being used for processing SSL transactions must be bound to the virtual server that receives the SSL data. If you have multiple virtual servers receiving SSL data, a valid certificate-key pair must be bound to each of them.

You can use a valid, existing SSL certificate that you have uploaded to the NetScaler appliance. As an alternative for testing purposes, you can create your own SSL certificate on the appliance. Intermediate certificates created by using a FIPS key on the NetScaler cannot be bound to an SSL virtual server.

As a part of the SSL handshake, in the certificate request message during client authentication, the server lists the distinguished names (DNs) of all the certificate authorities (CAs) bound to the server from which it will accept a client certificate. If you do not want the DN name of a specific CA certificate to be sent to the SSL client, set the skipCA flag. This setting indicates that the particular CA certificate's distinguished name should not be sent to the SSL client.

For details on how to create your own certificate, see Managing Certificates.

Note: Citrix recommends that you use only valid SSL certificates that have been issued by a trusted certificate authority.

To bind an SSL certificate-key pair to a virtual server by using the command line interface

At the command prompt, type the following commands to bind an SSL certificate-key pair to a virtual server and verify the configuration:

- bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -CA -skipCAName
- show ssl vserver <vServerName>

Example

```
> bind ssl vs vs1 -certkeyName cert2 -CA -skipCAName
Done
> sh ssl vs vs1
Advanced SSL configuration for VServer vs1:
DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Push Encryption Trigger: Always
Send Close-Notify: YES
1) CertKey Name: cert1 CA Certificate OCSPCheck: Optional CA_Name Sent
2) CertKey Name: cert2 CA Certificate OCSPCheck: Optional CA_Name Skipped
1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
```

To unbind an SSL certificate-key pair from a virtual server by using the command line interface

If you try to unbind a certificate-key pair from a virtual server by using the unbind ssl certKey <certkeyName> command, an error message appears because the syntax of the command has changed. At the command prompt, type the following command:

```
unbind ssl vserver <vServerName> -certkeyName <string>
```

Example

```
unbind ssl vserver vssl -certkeyName sslckey
```

To bind an SSL certificate-key pair to a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open an SSL virtual server and, in Advanced Settings, click SSL Certificate.
3. Bind a server certificate or CA certificate to the virtual server. To add a server certificate as an SNI certificate, select Server Certificate for SNI.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Profiles

Mar 24, 2016

> **Note**
>
> This feature is available in release 11 build 64.x and later.

The SSL infrastructure on the NetScaler appliance is continually updated to address the ever growing requirements for security and performance. Vulnerabilities in SSLv3 and RC4 implementation have emphasized the need to use the latest ciphers and protocols to negotiate the security settings for a network connection. Implementing any changes to the configuration, such as disabling SSLv3, across thousands of SSL end points is a cumbersome process. Therefore, settings that were part of the SSL end points configuration have been moved to the SSL profile, along with the default ciphers. To implement any change in the configuration, including cipher support, you only need to modify the profile. If the profile is enabled, the change is immediately reflected in all the end points that the profile is bound to.

> **Important**
>
> After the upgrade, if you enable the profile, you cannot reverse the changes. That is, the profile cannot be disabled.

The default SSL profiles contain all the default ciphers and ECC curves in addition to the settings that were part of the old profile. Sample outputs for the default profiles are provided later in this document. You can modify the default profile to suit your deployment. The default profile is bound to any new SSL entity. However, you can also create custom profiles and bind them to SSL entities.

By default, some SSL parameters, called *global parameters*, apply to all the SSL end points. If no SSL profile is bound to the configuration, the global parameters apply to all the SSL virtual servers, services, and service groups. If a profile is bound, the global parameters do not apply. The settings specified in the profile apply instead.

Differences between the Old SSL Profile and the New SSL Profile

The new SSL profile is different from previous versions in the following ways:

	Old Profile	**New Profile**
Ciphers and ECC Curves included in the profile	No	Yes
Inserting a cipher or cipher group in the middle of an existing list	Unbind all the ciphers and bind again in the order of the required priority.	Add a cipher and assign it a priority. If a priority is not specified, the cipher is assigned the lowest priority in the list.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Unbinding all the ciphers	> unbind ssl vserver <name> ciphername –ALL	unbind ssl profile –cipherName –FlushAllCiphers (From release 11.0 build 64.x, a new parameter FlushAllCiphers is added to unbind all the ciphers or cipher groups from a profile, because ALL is treated like a cipher group.)
SSLv3 is disabled in the profile	n/a	Yes. Note: SSLv3 is enabled in the virtual server settings by default to handle legacy deployments. You must explicitly disable SSLv3 in the virtual server if your deployment does not require the SSLv3 protocol.

Points to note

1. A profile can be bound to multiple virtual servers, but a virtual server can have only one profile bound to it.

2. You cannot delete a profile that is bound to a virtual server without first unbinding the profile.

3. A cipher or cipher group can be bound to multiple profiles at different priorities.

4. A profile can have multiple ciphers and cipher groups bound at different priorities.

5. Changes to a cipher group are immediately reflected in all the profiles and in all the virtual servers that one of the profiles is bound to.

6. If a cipher suite is part of a cipher group, you cannot remove the cipher suite from the profile without first editing the cipher group to remove the specific cipher suite.

7. If you do not assign a priority to a cipher suite or cipher group that you attach to a profile, it is assigned the lowest priority within the profile.

8. You can create a custom cipher group (also called a user-defined cipher group) from existing cipher groups and cipher suites. If you create cipher group A and add existing cipher groups X and Y to it, in that order, cipher group Y is assigned at a lower priority than cipher group X. That is, the group that is added first has a higher priority.

9. If a cipher suite is already part of a cipher group that is attached to a profile, and the same cipher suite is part of another cipher group that is also attached to the same profile, the cipher suite is not added again as part of the second cipher group. The cipher suite at the higher priority is in effect when traffic is processed.

10. Cipher groups are not expanded in the profile. As a result, the number of lines in the configuration file (ns.conf) is greatly reduced. For example, if there are a thousand SSL virtual servers to which two cipher groups are bound, and each cipher group contains 15 ciphers, expansion would result in 30*1000 entries related to ciphers in the configuration file. With the new profile, it would have only two entries: one for each cipher group that is bound to a profile.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

11. Creating a user defined cipher group from existing ciphers and cipher groups is a copy-paste operation. Any changes in the original group are not reflected in the new group.

12. A user-defined cipher group lists all the profiles that it is a part of.

13. A profile lists all the SSL virtual server, services, and service groups that it is bound to.

14. If the default SSL profile feature is enabled, you must use the profile to set or change any of the attributes of a virtual server, service, service group, or an internal service.

Upgrading to release 11 build 64.x or later

1. Run a script to capture the SSL-specific changes. In addition to other migration activities, the script analyzes the old ns.conf file and moves any special settings (settings other than the default) from an SSL end point configuration to a custom profile. You must enable the default profile after the upgrade for the configuration changes to apply.

> **Note**
>
> When running the migration script, you can choose to automatically generate the profile name, or you can prompt the user for the profile name interactively. The migration script checks the following and creates profiles accordingly.

- End points with the default settings and similar ciphers and cipher group settings: The script creates one profile.
- End points with the default settings and with different cipher groups or different priorities for the ciphers/cipher groups: In each case, the script creates a user-defined cipher group and binds it to a profile. Each profile is bound to the different end points.
- End points with the default settings and default ciphers: The default profile is bound to the end point.

Download the binary and run the following command from the folder in which you store the binary.

To run the script, at the command prompt, type:

Code COPY

```
./default_profile_script /nsconfig/ns.conf -b > <output file name>
```

2. Upgrade the software.
3. Enable SSL profiles by using the NetScaler command line or configuration utility.

- At the command line, type: set ssl parameter -defaultProfile ENABLED
- If you prefer to use the configuration utility, navigate to **Traffic Management** > **SSL** > **Change advanced SSL settings**, scroll down, and select **Enable Default Profile**.

If a profile was not bound to an end point before the upgrade, the default profile is bound to the SSL end point. If a profile was bound to an end point before the upgrade, the same profile is bound after the upgrade, and default ciphers are added to the profile.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Apply the commands in the text file (output of running the migration script) to the configuration. After you apply the commands in the text file, custom profile(s) are created for end points for which default parameters and ciphers have been changed, and automatically bound to the end points.

5. Verify the configuration.

Sample Migration of the SSL Configuration after Upgrade

Sample settings on an SSL virtual server, service, and service group are shown below. On the virtual server, client authentication is ENABLED (default is DISABLED), and the AES cipher group is bound to the virtual server. On the service, server authentication is ENABLED (default is DISABLED), and the AES cipher group is bound to the service. The service group has the default settings.

Example COPY

```
> sh ssl vserver v1


    Advanced SSL configuration for VServer v1:

    DH: DISABLED

    Ephemeral RSA: ENABLED        Refresh Count: 0

    Session Reuse: ENABLED        Timeout: 120 seconds

    Cipher Redirect: DISABLED

    SSLv2 Redirect: DISABLED

    ClearText Port: 0

    Client Auth: ENABLED Client Cert Required: Mandatory

    SSL Redirect: DISABLED

    Non FIPS Ciphers: DISABLED

    SNI: DISABLED

    SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Push Encryption Trigger: Always

    Send Close-Notify: YES



    ECC Curve: P_256, P_384, P_224, P_521



1)  CertKey Name: mycertkey    Server Certificate





1)  Cipher Name: AES

    Description: Predefined Cipher Alias

 Done



> sh ssl service svc1



    Advanced SSL configuration for Back-end SSL Service svc1:

    DH: DISABLED

    Ephemeral RSA: DISABLED

    Session Reuse: ENABLED        Timeout: 300 seconds

    Cipher Redirect: DISABLED

    SSLv2 Redirect: DISABLED
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    ClearText Port: 0

    Server Auth: ENABLED

    SSL Redirect: DISABLED

    Non FIPS Ciphers: DISABLED

    SNI: DISABLED

    SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED

    Send Close-Notify: YES





1)   Cipher Name: AES

    Description: Predefined Cipher Alias

 Done



> sh ssl serviceGroup

1) Service Group Name: sg1

    Session Reuse: ENABLED        Timeout: 300 seconds

    Server Auth: DISABLED

    Non FIPS Ciphers: DISABLED

    SSLv3: ENABLED  TLSv1.0: ENABLED

    Send Close-Notify: YES
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Send Close Notify: YES

Done
```

The following procedure migrates the above configuration.

1. Save your configuration.

2. Run the migration script. You can redirect the output to a text file if you use the default names for the profiles. Type:

Code COPY

```
./default_profile_script /nsconfig/ns.conf -b > ssl_config.txt
```

Use an editor, such as vi, to view the changes.The output cannot be redirected if you provide the profile names interactively. The output is displayed on the console and you must copy-paste it into a text file to apply it to your configuration after the upgrade.

3. After the upgrade, enable the profile.

- At the command line, type: set ssl parameter -defaultProfile ENABLED
- In the configuration utility, navigate to **Traffic Management** > **SSL** > **Change advanced SSL settings**, scroll down and select **Enable Default Profile**.

The interim output for the three new profiles that are created for the virtual server, service, and service group respectively is shown below. The default profiles are bound to the end point until you apply the changes in the text file that was created after running the migration script.

Code COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> sh ssl vserver v1

Advanced SSL configuration for VServer v1:

Profile Name :ns_default_ssl_profile_frontend

1) CertKey Name: mycertkey Server Certificate

 Done

>

> sh ssl service svc1



Advanced SSL configuration for Back-end SSL Service svc1:

Profile Name :ns_default_ssl_profile_backend

 Done

>

> sh ssl serviceGroup sg1



Advanced SSL configuration for Back-end SSL Service Group sg1:

Profile Name :ns_default_ssl_profile_backend

 Done
```

4. You must now apply the configuration in ssl_config.txt to the current configuration so that your non-default settings are applied after the upgrade.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Code COPY

```
batch -f /<path to the batch file>/ssl_config.txt
```

5. After applying the configuration, the output changes as follows:

Code COPY

```
> show ssl vserver v1

    Advanced SSL configuration for VServer v1:

    Profile Name :profile-002

1)   CertKey Name: mycertkey    Server Certificate

 Done

> show ssl service svc1

    Advanced SSL configuration for Back-end SSL Service svc1:

    Profile Name :profile-001

 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> show ssl serviceGroup sg1


    Advanced SSL configuration for Back-end SSL Service Group sg1:

    Profile Name :profile-003

 Done


> show ssl profile profile-002

1)   Name: profile-002    (Front-End)

    SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED

    Client Auth: ENABLED Client Cert Required: Mandatory

    Use only bound CA certificates: DISABLED

    Strict CA checks:         NO

    Session Reuse: ENABLED         Timeout: 120 seconds

    DH: DISABLED

    Ephemeral RSA: ENABLED         Refresh Count: 0

    Deny SSL Renegotiation         ALL

    Non FIPS Ciphers: DISABLED

    Cipher Redirect: DISABLED

    SSL Redirect: DISABLED
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Send Close-Notify: YES

    Push Encryption Trigger: Always

    PUSH encryption trigger timeout:     1 ms

    SNI: DISABLED

    Strict Host Header check for SNI enabled SSL sessions:         NO

    Push flag: 0x0 (Auto)

    SSL quantum size:         8 kB

    Encryption trigger timeout 100 mS

    Encryption trigger packet count:     45

    Subject/Issuer Name Insertion Format: Unicode





    ECC Curve: P_256, P_384, P_224, P_521



1)   Cipher Name: AES     Priority :1

    Description: Predefined Cipher Alias



1)   Vserver Name: v1

 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> show ssl profile profile-001

1)   Name: profile-001    (Back-End)

     SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED

     Server Auth: ENABLED

     Use only bound CA certificates: DISABLED

     Strict CA checks:         NO

     Session Reuse: ENABLED         Timeout: 120 seconds

     Deny SSL Renegotiation         ALL

     Non FIPS Ciphers: DISABLED

     Send Close-Notify: YES

     Push Encryption Trigger: Always

     PUSH encryption trigger timeout:     1 ms

     Push flag: 0x0 (Auto)

     SSL quantum size:         8 kB

     Encryption trigger timeout 100 mS

     Encryption trigger packet count:     45
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    ECC Curve: P_256, P_384, P_224, P_521


1)  Cipher Name: AES    Priority :1

    Description: Predefined Cipher Alias


1)  Service Name: svc1

 Done


> show ssl profile profile-003

1)  Name: profile-003   (Back-End)

    SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED

    Server Auth: DISABLED

    Use only bound CA certificates: DISABLED

    Strict CA checks:       NO

    Session Reuse: ENABLED       Timeout: 120 seconds

    Deny SSL Renegotiation       ALL

    Non FIPS Ciphers: DISABLED

    Send Close-Notify: YES

    Push Encryption Trigger: Always

    PUSH encryption trigger timeout:    1 ms
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Push flag: 0x0 (Auto)

    SSL quantum size:         8 kB

    Encryption trigger timeout 100 mS

    Encryption trigger packet count:     45





    ECC Curve: P_256, P_384, P_224, P_521



1)   Cipher Name: ALL     Priority :1

    Description: Predefined Cipher Alias



1)   Service Name: sg1

 Done
```

Default Front End and Backend SSL Profile Settings

A default front end profile has the following settings:

Code COPY

```
> sh ssl profile ns_default_ssl_profile_frontend

1)Name: ns_default_ssl_profile_frontend

    Configuration for Front-End SSL profile
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
DH: DISABLED

Ephemeral RSA: ENABLED        Refresh Count: 0

Session Reuse: ENABLED        Timeout: 120 seconds

Non FIPS Ciphers: DISABLED

Cipher Redirect: ENABLED   Redirect URL: http://10.102.28.212/redirect.html

Client Auth: DISABLED

SSL Redirect: DISABLED

SNI: DISABLED

SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2: ENABLED

Push Encryption Trigger: Always

PUSH encryption trigger timeout:     1 ms

Send Close-Notify: YES

Push flag: 0x0 (Auto)

Deny SSL Renegotiation        NO

SSL quantum size:        8 kB

Strict CA checks:        NO

Encryption trigger timeout 100 mS

Encryption trigger packet count:     45

Use only bound CA certificates: DISABLED

Subject/Issuer Name Insertion Format: Unicode
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Strict Host Header check for SNI enabled SSL sessions:        NO



    ECC Curve: P_256, P_384, P_521



1)  Cipher Name: AES     Priority :2

    Description: Predefined Cipher Alias



1)  Vserver Name: v1  >>>>>>>>>>>

2)  Vserver Name: nshttps-::1l-443 >>>>>>>>>>>

3)  Vserver Name: nsrpcs-::1l-3008

4)  Vserver Name: nskrpcs-127.0.0.1-3009

5)  Vserver Name: nshttps-127.0.0.1-443

6)  Vserver Name: nsrpcs-127.0.0.1-3008

Done
```

A default backend profile has the following settings:

Code COPY

```
> sh ssl profile ns_default_ssl_profile_backend

1)Name: ns_default_ssl_profile_backend

    Configuration for Back-End SSL profile
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Session Reuse: ENABLED        Timeout: 300 seconds

    Non FIPS Ciphers: DISABLED

    Server Auth: DISABLED

    SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED

    Push Encryption Trigger: Always

    PUSH encryption trigger timeout:     1 ms

    Send Close-Notify: YES

    Push flag: 0x0 (Auto)

    Deny SSL Renegotiation        ALL

    SSL quantum size:        8 kB

    Strict CA checks:        NO

    Encryption trigger timeout 100 mS

    Encryption trigger packet count:     45

    Use only bound CA certificates: DISABLED



    ECC Curve: P_256, P_224, P_521



1)  Cipher Name: AES     Priority :1

    Description: Predefined Cipher Alias
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
2)   Cipher Name: RC4     Priority :2

     Description: Predefined Cipher Alias



1)   Service Name: s2 >>>>>>>>>>>>>

2)   Service Name: s1 >>>>>>>>>>>>>

Done
```

Limitations

SSL profiles are not supported in a cluster setup, or with Admin Partitions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring an SSL Virtual Server for Secure Hosting of Multiple Sites

May 13, 2015

Virtual hosting is used by Web servers to host more than one domain name with the same IP address. The NetScaler supports hosting of multiple secure domains by offloading SSL processing from the Web servers using transparent SSL services or virtual server-based SSL offloading. However, when multiple Web sites are hosted on the same virtual server, the SSL handshake is completed before the expected host name is sent to the virtual server. As a result, the NetScaler cannot determine which certificate to present to the client after a connection is established. This problem is resolved by enabling Server Name Indication (SNI) on the virtual server. SNI is a Transport Layer Security (TLS) extension used by the client to provide the host name during handshake initiation. The NetScaler appliance compares this host name to the common name and, if it does not match, compares it to the subject alternative name (SAN). If the name matches, the appliance presents the corresponding certificate to the client.

A wildcard SSL Certificate helps enable SSL encryption on multiple subdomains if the domains are controlled by the same organization and share the same second-level domain name. For example, a wildcard certificate issued to a sports network using the common name "*.sports.net" can be used to secure domains, such as "login.sports.net" and "help.sports.net" but not "login.ftp.sports.net."

Note: On a NetScaler appliance, only domain name, URL, and email ID DNS entries in the SAN field are compared.
You can bind multiple server certificates to a single SSL virtual server or transparent service using the -SNICert option. These certificates are issued by the virtual server or service if SNI is enabled on the virtual server or service. You can enable SNI at any time.

To bind multiple server certificates to a single SSL virtual server by using the command line interface

At the command prompt, type the following commands to configure SNI and verify the configuration:

- set ssl vserver <vServerName>@ [-SNIEnable (ENABLED | DISABLED)]
- bind ssl vserver <vServerName>@ -certkeyName <string> -SNICert
- show ssl vserver <vServerName>

To bind multiple server certificates to a transparent service by using the NetScaler command line, replace vserver with service and vservername with servicename in the above commands.

Note: The SSL service should be created with -clearTextPort 80 option.

Example

```
set ssl vserver v1 -snI ENABLED
bind ssl vserver v1 -certkeyName serverabc -SNICert
sh ssl vserver v1
Advanced SSL configuration for VServer v1:
…
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: ENABLED
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
1)CertKey Name: servercert Server Certificate
1)CertKey Name: abccert Server Certificate for SNI
2)CertKey Name: xyzcert Server Certificate for SNI
3)CertKey Name: startcert Server Certificate for SNI
1)Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
```

To bind multiple server certificates to a single SSL virtual server or transparent SSL service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open an SSL virtual server and, in Certificates, click Server Certificate.
3. Add a new certificate or select a certificate from the list, and select Server Certificate for SNI.
4. In Advanced Settings, click SSL Parameters.
5. Select SNI Enable.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a DTLS Virtual Server

Jun 17, 2015

The SSL and TLS protocols have traditionally been used to secure streaming traffic. Both of these protocols are based on TCP, which is very slow. In addition, TLS cannot handle lost or reordered packets.

UDP is the preferred protocol for audio and video applications, such as Lync, Skype, iTunes, YouTube, training videos, and flash. However, UDP is not secure or reliable. The DTLS protocol is designed to secure data over UDP and is used for applications such as media streaming, VOIP, and online gaming for communication. In DTLS, each handshake message is assigned a specific sequence number within that handshake. When a peer receives a handshake message, it can quickly determine whether that message is the next one expected. If it is, the peer processes the message. If not, the message is queued for handling after all the previous messages have been received.

You must create a DTLS virtual server and a service of type UDP. By default, a DTLS profile (nsdtls_default_profile) is bound to the virtual server. Optionally, you can create and bind a user-defined DTLS profile to the virtual server.

Note: RC4 ciphers are not supported on a DTLS virtual server.

To create a DTLS configuration by using the command line

At the command prompt, type:

```
add lb vserver <vserver_name> DTLS <IPAddress>  <port>
add service  <service_name>  <IPAddress> UDP 443
bind lb vserver  <vserver_name>  <udp_service_name>
```

The following steps are optional:

```
add dtlsProfile dtls1 -maxretryTime <positive_integer>
set ssl vserver <vserver_name> -dtlsProfileName <dtls_profile_name>
```

To create a DTLS configuration by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Create a virtual server of type DTLS, and bind a UDP service to the virtual server.
3. A default DTLS profile is bound to the DTLS virtual server. To bind a different profile, in SSL Parameters, select a different DTLS profile. To create a new profile, click the plus (+) next to DTLS Profile.

Example

The following example is for an end-to-end DTLS configuration:

```
> enable ns feature SSL LB
> add server s1 10.102.59.190
> add service svc1 s1 UDP 32000
> add lb vserver lb1 DTLS 10.102.59.244 443
> add ssl certKey servercert -cert server_cert.pem -key server_key.pem
> bind ssl vserver lb1 -certkeyname servercert
> bind lb vserver lb1 svc1

> sh lb vserver lb1
    lb1 (10.102.59.244:443) - DTLS  Type: ADDRESS
    State: UP
    Last state change was at Tue May 20 16:41:27 2014
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        Time since last state change: 0 days, 00:01:39.120
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Appflow logging: ENABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: IP
        Persistence: NONE
        L2Conn: OFF
        Skip Persistency: None
        IcmpResponse: PASSIVE
        RHIstate: PASSIVE
        New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0
        TD: 0
        Mac mode Retain Vlan: DISABLED
        DBS_LB: DISABLED
        Process Local: DISABLED

        1 bound service:
1) svc1 (10.102.59.190: 32000) - UDP State: UP  Weight: 1
 Done
>
> sh ssl vserver lb1

        Advanced SSL configuration for VServer lb1:
        DH: DISABLED
        Ephemeral RSA: ENABLED         Refresh Count: 0
        Session Reuse: ENABLED         Timeout: 1800 seconds
        Cipher Redirect: DISABLED

        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        DTLSv1: ENABLED
        Send Close-Notify: YES

        DTLS profile name: nsdtls_default_profile

        1 bound certificate:

1)      CertKey Name: servercert       Server Certificate

        1 configured cipher:
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
1)     Cipher Name: DEFAULT
       Description: Predefined Cipher Alias
 Done

> sh dtlsProfile nsdtls_default_profile
1) Name: nsdtls_default_profile
 PMTU Discovery: DISABLED
 Max Record Size: 1460 bytes
 Max Retry Time: 3 sec
 Hello Verify Request: DISABLED
 Terminate Session: DISABLED
 Max Packet Count: 120 bytes
 Done
```

Features not supported by a DTLS virtual server

The following options cannot be enabled on a DTLS virtual server:

- SSLv2
- SSLv3
- TLSv1
- TLSv1.1
- TLSv1.2
- Push encrypt trigger
- SSLv2Redirect
- SSLv2URL
- SNI
- Secure renegotiation

Parameters not used by a DTLS virtual server

The following SSL parameters, even if set, are ignored by a DTLS virtual server:

- Encryption trigger packet count
- PUSH encryption trigger timeout
- SSL quantum size
- Encryption trigger timeout
- Subject/Issuer Name Insertion Format

© 1999-2017 Citrix Systems, Inc. All rights reserved.

DTLS Profile

Feb 03, 2015

A DTLS profile with the default settings is automatically bound to a DTLS virtual server. However, you can create a new DTLS profile with specific settings to suit your requirement.

To create a DTLS profile by using the command line

- add ssl dtlsProfile <name>
- show ssl dtlsProfile<name>

Example

```
> add dtlsProfile dtls1 -helloVerifyRequest ENABLED -maxretryTime 4
 Done
> show dtlsProfile dtls1
1)     Name: dtls1
       PMTU Discovery: DISABLED
       Max Record Size: 1460 bytes
       Max Retry Time: 4 sec
       Hello Verify Request: ENABLED
       Terminate Session: DISABLED
       Max Packet Count: 120 bytes

 Done
```

To create a DTLS profile by using the configuration utility

Navigate to System > Profiles > DTLS Profiles and configure a new profile.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Importing SSL Files from Remote Hosts

May 20, 2014

You can now import SSL resources, such as certificates, private keys , CRLs, and DH keys, from remote hosts even if FTP access to these hosts is not available. This is especially helpful in environments where shell access to the remote host is restricted. Default folders are created in /nsconfig/ssl as follows:

- For certificate files: /nsconfig/ssl/certfile
- For private keys: the /nsconfig/ssl/keyfile
- For CRLs: /var/netscaler/ssl/crlfile
- For DH keys: /nsconfig/ssl/dhfile

Imports from both HTTP and HTTPS servers are supported. However, the import fails if the file is on an HTTPS server that requires client certificate authentication for access.

Note: The import command is not stored in the configuration (ns.conf) file, because reimporting the file after a restart might cause an error.

To import a certificate file from a remote host by using the command line

At the command prompt, type:

```
import ssl certFile [<name>] [<src>]
```

Example

```
import ssl certfile my-certfile http://www.example.com/file_1
> show ssl certfile
   Name : my-certfile
   URL : http://www.example.com/file_1
```

To remove a certificate file, use the rm ssl certFile command, which accepts only the <name> argument.

To import a key file from a remote host by using the command line

At the command prompt, type:

```
import ssl keyFile [<name>] [<src>]
```

Example

```
import ssl keyfile my-keyfile http://www.example.com/key_file
> show ssl keyfile
   Name : my-keyfile
   URL : http://www.example.com/key_file
```

To remove a key file, use the rm ssl keyFile command, which accepts only the <name> argument.

To import a CRL file from a remote host by using the command line

At the command prompt, type:

```
import ssl crlFile [<name>] [<src>]
```

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
import ssl crlfile my-crlfile http://www.example.com/crl_file
> show ssl crlfile
    Name : my-crlfile
    URL : http://www.example.com/crl_file
```

To remove a CRL file, use the rm ssl crlFile command, which accepts only the <name> argument.

To import a DH file from a remote host by using the command line

At the command prompt, type:

```
import ssl dhFile [<name>] [<src>]
```

Example

```
import ssl dhfile my-dhfile http://www.example.com/dh_file
> show ssl dhfile
    Name : my-dhfile
    URL : http://www.example.com/dh_file
```

To remove a DH file, use the rm ssl dhFile command, which accepts only the <name> argument.

To import an SSL resource by using the configuration utility

Navigate to Traffic Management > SSL > Imports, and then select the appropriate tab.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL Profiles

May 20, 2015

You can use an SSL profile to specify how a NetScaler ADC processes SSL traffic. The profile is a collection of SSL parameter settings for SSL entities, such as virtual servers, services, and service groups, and offers ease of configuration and flexibility. You are not limited to configuring only one set of global parameters. You can create multiple sets (profiles) of global parameters and assign different sets to different SSL entities. SSL profiles are classified into two categories:

- Front end profiles, containing parameters applicable to the front-end entity. That is, they apply to the entity that receives requests from a client.
- Backend profiles, containing parameters applicable to the back-end entity. That is, they apply to the entity that sends client requests to a server.

Unlike a TCP or HTTP profile, an SSL profile is optional. Therefore, there is no default SSL profile. The same profile can be reused across multiples entities. If an entity does not have a profile attached, the values set at the global level apply. For dynamically learned services, current global values apply.

The following table lists the parameters that are part of each profile.

Front end profile	Backend profile
cipherRedirect, cipherURL	denySSLReneg
clearTextPort*	encryptTriggerPktCount
clientAuth, clientCert	nonFipsCiphers
denySSLReneg	pushEncTrigger
dh, dhFile, dhCount	pushEncTriggerTimeout
dropReqWithNoHostHeader	pushFlag
encryptTriggerPktCount	quantumSize
eRSA, eRSACount	serverAuth
insertionEncoding	commonName
nonFipsCiphers	sessReuse, sessTimeout
pushEncTrigger	SNIEnable

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Front end profile	Backend profile
pushEncTriggerTimeout	ssl3
pushFlag	sslTriggerTimeout
quantumSize	strictCAChecks
redirectPortRewrite	tls1
sendCloseNotify	
sessReuse, sessTimeout	
SNIEnable	
ssl3	
sslRedirect	
sslTriggerTimeout	
strictCAChecks	
tls1, tls11, tls12	

* The clearTextPort parameter applies only to an SSL virtual server.
An error message appears if you try to set a parameter that is not part of the profile (for example, if you try to set the clientAuth parameter in a backend profile).

Some SSL parameters, such as CRL memory size, OCSP cache size, UndefAction Control, and UndefAction Data, are not part of any of the above profiles, because these parameters are independent of entities.

An SSL profile supports the following operations:

- Add—Creates an SSL profile on the NetScaler ADC. Specify whether the profile is front end or backend. Front end is the default.
- Set—Modifies the settings of an existing profile.
- Unset—Sets the specified parameters to their default values. If you do not specify any parameters, an error message appears. If you unset a profile on an entity, the profile is unbound from the entity.
- Remove—Deletes a profile. A profile that is being used by any entity cannot be deleted. Clearing the configuration deletes all the entities. As a result, the profiles are also deleted.
- Show—Displays all the profiles that are available on the NetScaler ADC . If a profile name is specified, the details of that profile are displayed. If an entity is specified, the profiles associated with that entity are displayed.

To create an SSL profile by using the command line

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- To add an SSL profile, type: add ssl profile <name> [-sslProfileType (BackEnd | FrontEnd)]
- To modify an existing profile, type: set ssl profile <name>
- To unset an existing profile, type: unset ssl profile <name> [-dh] [-dhFile] [-dhCount] [-eRSA]…
- To unset an existing profile from an entity, type: unset ssl vserver <vServerName> –sslProfile
- To remove an existing profile, type: rm ssl profile <name>
- To display an existing profile, type: sh ssl profile <name>

Examples

1. Adding a front end (default) profile:
   > add sslprofile p1
   Done
2. Adding a backend profile:
   > add sslprofile p2 -sslprofileType backend -tls1 disabled
   Done
3. Enabling settings on a backend profile:
   > set sslprofile p2 -serverAuth eNABLED
   Done
4. Enabling settings on a frontend profile:
   > set sslprofile p1 -clientauth eNABLED -clientcert optional
   Done
   sh ssl profile p1
   1) Configuration for Front-End SSL profile
   Name: p1
   DH: DISABLED
   Ephemeral RSA: ENABLED Refresh Count: 0
   Session Reuse: ENABLED Timeout: 120 seconds
   Non FIPS Ciphers: DISABLED
   Cipher Redirect: DISABLED
   Client Auth: ENABLED Client Cert Required: Optional
   SSL Redirect: DISABLED
   SNI: DISABLED
   SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: DISABLED TLSv1.2: DISABLED
   Push Encryption Trigger: Always
   PUSH encryption trigger timeout: 1 ms
   Send Close-Notify: YES
   Push flag: 0x0 (Auto)
   Deny SSL Renegotiation NO
   SSL quantum size: 8 kB
   Strict CA checks: NO
   Encryption trigger timeout 100 mS
   Encryption trigger packet count: 45
   Subject/Issuer Name Insertion Format: Unicode
   Strict Host Header check for SNI enabled SSL sessions: NO
   Done
5. Settings parameters to their default values:
   > unset sslprofile p1 -clientauth -clientcert

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Done
> sh ssl profile p1
1)    Configuration for Front-End SSL profile
      Name: p1
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 0
      Session Reuse: ENABLED        Timeout: 120 seconds
      Non FIPS Ciphers: DISABLED
      Cipher Redirect: DISABLED
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      SNI: DISABLED
      SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED
      Push Encryption Trigger: Always
      PUSH encryption trigger timeout:      1 ms
      Send Close-Notify: YES
      Push flag:     0x0 (Auto)
      Deny SSL Renegotiation        NO
      SSL quantum size:             8 kB
      Strict CA checks:             NO
      Encryption trigger timeout     100 mS
      Encryption trigger packet count:      45
      Subject/Issuer Name Insertion Format:   Unicode
      Strict Host Header check for SNI enabled SSL sessions:        NO
Done
```

6. Deleting a profile:

```
> rm sslprofile p1
Done
```

7. Binding a profile to a virtual server:

```
> set ssl vserver v1 -sslprofile p3
Done
```

8. Unbinding a profile from a virtual server:

```
>unset ssl vserver v1 -sslprofile
  Done
```

To create an SSL profile by using the configuration utility

Navigate to System > Profiles, select the SSL Profiles tab, and create an SSL profile.

Enabling Stricter Control on Client Certificate Validation

The NetScaler appliance accepts valid Intermediate-CA certificates if they are issued by a single Root-CA. That is, if only the Root-CA certificate is bound to the virtual server, and any intermediate certificate sent with the client certificate is validated by that Root-CA, the appliance trusts the certificate chain and the handshake is successful.

However, if a client sends a chain of certificates in the handshake, none of the intermediate certificates can be validated by using a CRL or OCSP responder unless that certificate is bound to the SSL virtual server. Therefore, even if one of the intermediate certificates is revoked, the handshake is successful. As part of the handshake, the SSL virtual server sends the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

list of CA certificates that are bound to it. For stricter control, you can configure the SSL virtual server to accept only a certificate that is signed by one of the CA certificates bound to that virtual server. To do so, you must enable the ClientAuthUseBoundCAChain setting in the SSL profile bound to the virtual server. The handshake fails if the client certificate is not signed by one of the CA certificates bound to the virtual server.

For example, say two client certificates, clientcert1 and clientcert2, are signed by the intermediate certificates Int-CA-A and Int-CA-B, respectively. The intermediate certificates are signed by the root certificate Root-CA. Int-CA-A and Root-CA are bound to the SSL virtual server. In the default case (ClientAuthUseBoundCAChain disabled), both clientcert1 and clientcert2 are accepted. However, if ClientAuthUseBoundCAChain is enabled, only clientcert1 is accepted by the NetScaler appliance

To enable stricter control on client certificate validation by using the command line

At the NetScaler command prompt, type:set ssl profile <name> -ClientAuthUseBoundCAChain Enabled

To enable stricter control on client certificate validation by using the configuration utility

1. Navigate to System > Profiles, select the SSL Profiles tab, and create an SSL profile, or select an existing profile.
2. Select Enable Client Authentication using bound CA Chain.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Stricter Control on Client Certificate Validation

Jun 19, 2015

The NetScaler appliance accepts valid Intermediate-CA certificates if they are issued by a single Root-CA. That is, if only the Root-CA certificate is bound to the virtual server, and any intermediate certificate sent with the client certificate is validated by that Root-CA, the appliance trusts the certificate chain and the handshake is successful.

However, if a client sends a chain of certificates in the handshake, none of the intermediate certificates can be validated by using a CRL or OCSP responder unless that certificate is bound to the SSL virtual server. Therefore, even if one of the intermediate certificates is revoked, the handshake is successful. As part of the handshake, the SSL virtual server sends the list of CA certificates that are bound to it. For stricter control, you can configure the SSL virtual server to accept only a certificate that is signed by one of the CA certificates bound to that virtual server. To do so, you must enable the ClientAuthUseBoundCAChain setting in the SSL profile bound to the virtual server. The handshake fails if the client certificate is not signed by one of the CA certificates bound to the virtual server.

For example, say two client certificates, clientcert1 and clientcert2, are signed by the intermediate certificates Int-CA-A and Int-CA-B, respectively. The intermediate certificates are signed by the root certificate Root-CA. Int-CA-A and Root-CA are bound to the SSL virtual server. In the default case (ClientAuthUseBoundCAChain disabled), both clientcert1 and clientcert2 are accepted. However, if ClientAuthUseBoundCAChain is enabled, only clientcert1 is accepted by the NetScaler appliance

To enable stricter control on client certificate validation by using the command line

At the NetScaler command prompt, type:

- set ssl profile <name> -ClientAuthUseBoundCAChain Enabled
- set ssl vserver <vServerName> -sslProfile <string>

To enable stricter control on client certificate validation by using the configuration utility

1. Navigate to System > Profiles, select the SSL Profiles tab, and create an SSL profile, or select an existing profile.
2. Select Enable Client Authentication using bound CA Chain.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers, and select an SSL virtual server.
4. In Advanced Settings, select SSL Profiles, and select the profile on which you enabled Enable Client Authentication using bound CA Chain.
5. Click OK, and then click Done.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Graceful Cleanup of SSL Sessions

Dec 16, 2015

Some operations, such as updating a certificate to replace a potentially exposed certificate, using a stronger key (2048-bit instead of 1024-bit), adding or removing a certificate to or from a certificate chain, or changing any of the SSL parameters, should clean the SSL sessions gracefully instead of abruptly terminating the sessions.

From build 64.x, existing SSL connections do not break if you update the SSL certificate, cipher list, or SSL parameters. That is, all existing connections continue using the current settings until the sessions are closed, but all new connections use the new certificate or settings. To clear the sessions immediately after a configuration change, you must disable and reenable each entity.

Important: Connections that are in the middle of a handshake, or sessions that are renegotiating, are terminated. Session reuse is not allowed. Additionally, session multiplexing reuse at the back end is not allowed.

If you change a front-end parameter, such as on an SSL virtual server, only the front end connections are affected. If you change a back-end parameter, such as a parameter on an SSL service or service group, only the back-end connections are affected. Changes such as ciphers and certificates apply to both front-end and back-end connections.

The following configuration commands or changes trigger a graceful session cleanup on all affected SSL entities:

1. set ssl vserver command
2. set ssl service command
3. set ssl servicegroup command
4. set ssl profile command
5. set ssl cipher <cipherGroupName> command
6. Binding, unbinding, and reordering ciphers
7. Binding and unbinding ecccurves
8. Inserting, removing, linking and unlinking a certificate

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enhanced SSL Profiles Infrastructure Overview

Jun 30, 2016

Vulnerabilities in SSLv3 and RC4 implementation have emphasized the need to use the latest ciphers and protocols to negotiate the security settings for a network connection. Implementing any changes to the configuration, such as disabling SSLv3 across thousands of SSL end points, is a cumbersome process. Therefore, settings that were part of the SSL end points configuration have been moved to the SSL profiles, along with the default ciphers. To implement changes in the configuration, including cipher support, you need only modify the profile that is bound to the entities.

The default SSL profiles (default front-end and default back-end) contain all the default ciphers and ECC curves, in addition to the settings that were part of the old profiles. Sample outputs for the default profiles are provided in the appendix. The Enable Default Profile operation automatically binds the default front-end profile to all front-end entities, and the default back-end profile to all back-end entities. You can modify a default profile to suit your deployment. You can also create custom profiles and bind them to SSL entities.

> **Important**
>
> After the upgrade, if you enable the default profiles, you cannot undo the changes. That is, the profiles cannot be disabled. Save the configuration and create a copy of the configuration file (ns.conf) before enabling the profiles. However, if you do not wish to use the features in the default profile, you can continue to use the old SSL profiles. For more information about these profiles, see SSL Profiles.

By default, some SSL parameters, called global parameters, apply to all the SSL end points. However, if a profile is bound to an SSL end point, the global parameters do not apply. The settings specified in the profile apply instead.

Points to Note

1. A profile can be bound to multiple virtual servers, but a virtual server can have only one profile bound to it.
2. You cannot delete a profile that is bound to a virtual server without first unbinding the profile.
3. A cipher or cipher group can be bound to multiple profiles at different priorities.
4. A profile can have multiple ciphers and cipher groups bound at different priorities.
5. Changes to a cipher group are immediately reflected in all the profiles and in all the virtual servers that one of the profiles is bound to.
6. If a cipher suite is part of a cipher group, you cannot remove the cipher suite from the profile without first editing the cipher group to remove the specific cipher suite.
7. If you do not assign a priority to a cipher suite or cipher group that you attach to a profile, it is assigned the lowest priority within the profile.
8. You can create a custom cipher group (also called a user-defined cipher group) from existing cipher groups and cipher suites. If you create cipher group A and add existing cipher groups X and Y to it, in that order, cipher group Y is assigned at a lower priority than cipher group X. That is, the group that is added first has a higher priority.
9. If a cipher suite is already part of a cipher group that is attached to a profile, and the same cipher suite is part of another cipher group that is also attached to the same profile, the cipher suite is not added again as part of the second cipher group. The cipher suite at the higher priority is in effect when traffic is processed.
10. Cipher groups are not expanded in the profile. As a result, the number of lines in the configuration file (ns.conf) is greatly reduced. For example, if there are a thousand SSL virtual servers to which two cipher groups are bound, and each cipher

© 1999-2017 Citrix Systems, Inc. All rights reserved.

group contains 15 ciphers, expansion would result in 30*1000 entries related to ciphers in the configuration file. With the new profile, it would have only two entries: one for each cipher group that is bound to a profile.

11. Creating a user defined cipher group from existing ciphers and cipher groups is a copy-paste operation. Any changes in the original group are not reflected in the new group.
12. A user-defined cipher group lists all the profiles that it is a part of.
13. A profile lists all the SSL virtual server, services, and service groups that it is bound to.
14. If the default SSL profile feature is enabled, you must use the profile to set or change any of the attributes of a virtual server, service, service group, or an internal service.

Limitations

SSL profiles are not supported in a cluster setup, or with Admin Partitions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling the Default Profiles

Jun 30, 2016

> **Important**
>
> Save your configuration before you upgrade the software and enable the default profiles.

Upgrade the software to a build that supports the enhanced profile infrastructure, and then enable the default profiles. You can take one of two approaches depending on your specific deployment. If your deployment has a common SSL configuration across end points, see Use Case 1. If your deployment has a large SSL configuration and the SSL parameters and ciphers are not common among end points, see Use Case 2.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Differences between the Old and the New SSL Profile Infrastructure

Jun 30, 2016

	Old Profile	New Profile
Ciphers and ECC Curves included in the profile	No	Yes
Inserting a cipher or cipher group in the middle of an existing list	Unbind all the ciphers and bind again in the order of the required priority.	Add a cipher and assign it a priority. If a priority is not specified, the cipher is assigned the lowest priority in the list.
Unbinding all the ciphers	> unbind ssl vserver <name> ciphername –ALL	unbind ssl profile –cipherName –FlushAllCiphers (Release 11.0 build 64.x or later includes the FlushAllCiphers parameter for unbinding all the ciphers or cipher groups from a profile, because ALL is treated like a cipher group.)
State of SSLv3	n/a	Disabled on the default front-end profile (ns_default_ssl_profile_frontend). Note: Before you enable this profile, SSLv3 is enabled globally. After enabling the profile, SSLv3 is disabled on the front-end default profile.

Old Profile	New Profile	
Ciphers and ECC Curves included in the profile	No	Yes
Inserting a cipher or cipher group in the middle of an existing list	Unbind all the ciphers and bind again in the order of the required priority.	Add a cipher and assign it a priority. If a priority is not specified, the cipher is assigned the lowest priority in the list.
Unbinding all the ciphers	> unbind ssl vserver <name> ciphername –ALL	unbind ssl profile –cipherName –FlushAllCiphers (Release 11.0 build 64.x or later includes the FlushAllCiphers parameter for unbinding all the ciphers or cipher groups from a profile, because ALL is treated like a cipher group.)
State of SSLv3	n/a	Disabled on the default front-end profile (ns_default_ssl_profile_frontend). Note: Before you enable this profile, SSLv3 is enabled globally. After enabling the profile, SSLv3 is disabled on the front-end default profile.

Old Profile	New Profile	
Ciphers and ECC Curves included in the profile	No	Yes
Inserting a cipher or cipher group in the middle of an existing list	Unbind all the ciphers and bind again in the order of the required priority.	Add a cipher and assign it a priority. If a priority is not specified, the cipher is assigned the lowest priority in the list.
Unbinding all the ciphers	> unbind ssl vserver <name> ciphername –ALL	unbind ssl profile –cipherName –FlushAllCiphers (Release 11.0 build 64.x or later includes the FlushAllCiphers parameter for unbinding all the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

		ciphers or cipher groups from a profile, because ALL is treated like a cipher group.)
State of SSLv3	n/a	Disabled on the default front-end profile (ns_default_ssl_profile_frontend). Note: Before you enable this profile, SSLv3 is enabled globally. After enabling the profile, SSLv3 is disabled on the front-end default profile.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Appendix A: Sample Migration of the SSL Configuration after Upgrade

Jun 30, 2016

Sample settings on an SSL virtual server, service, and service group are shown below. On the virtual server, client authentication is ENABLED (default is DISABLED), and the AES cipher group is bound to the virtual server. On the service, server authentication is ENABLED (default is DISABLED), and the AES cipher group is bound to the service. The service group has the default settings.

```
> sh ssl vserver v1

  Advanced SSL configuration for VServer v1:

  DH: DISABLED

  Ephemeral RSA: ENABLED        Refresh Count: 0

  Session Reuse: ENABLED        Timeout: 120 seconds

  Cipher Redirect: DISABLED

  SSLv2 Redirect: DISABLED

  ClearText Port: 0

  Client Auth: ENABLED Client Cert Required: Mandatory

  SSL Redirect: DISABLED

  Non FIPS Ciphers: DISABLED

  SNI: DISABLED

  SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED

  Push Encryption Trigger: Always

  Send Close-Notify: YES


  ECC Curve: P_256, P_384, P_224, P_521


1)  CertKey Name: mycertkey    Server Certificate


1)  Cipher Name: AES

  Description: Predefined Cipher Alias
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Done

> **sh ssl service svc1**

Advanced SSL configuration for Back-end SSL Service svc1:

DH: DISABLED

Ephemeral RSA: DISABLED

Session Reuse: ENABLED Timeout: 300 seconds

Cipher Redirect: DISABLED

SSLv2 Redirect: DISABLED

ClearText Port: 0

Server Auth: ENABLED

SSL Redirect: DISABLED

Non FIPS Ciphers: DISABLED

SNI: DISABLED

SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: DISABLED TLSv1.2: DISABLED

Send Close-Notify: YES

1) Cipher Name: AES

Description: Predefined Cipher Alias

Done

> **sh ssl serviceGroup**

1) Service Group Name: sg1

Session Reuse: ENABLED Timeout: 300 seconds

Server Auth: DISABLED

Non FIPS Ciphers: DISABLED

SSLv3: ENABLED TLSv1.0: ENABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Send Close-Notify: YES

Done

The following procedure migrates the above configuration.

1. Save your configuration.

2. Run the migration script. You can redirect the output to a text file if you use the default names for the profiles. Type:

./default_profile_script /nsconfig/ns.conf -b > ssl_config.txt

Use an editor, such as vi, to view the changes. The output cannot be redirected if you provide the profile names interactively. The output is displayed on the console and you must copy and paste it into a text file to apply it to your configuration after the upgrade.

3. After the upgrade, enable the profile.

- At the command line, type: **set ssl parameter -defaultProfile ENABLED**
- In the GUI, navigate to **Traffic Management** > **SSL** > **Change advanced SSL settings**, scroll down and select **Enable Default Profile**.

The interim output for the three new profiles that are created for the virtual server, service, and service group, respectively, is shown below. The default profiles are bound to the end points until you apply the changes in the text file that was created after running the migration script.

> **sh ssl vserver v1**

Advanced SSL configuration for VServer v1:

Profile Name :ns_default_ssl_profile_frontend

1) CertKey Name: mycertkey Server Certificate

Done

>

> **sh ssl service svc1**

Advanced SSL configuration for Back-end SSL Service svc1:

Profile Name :ns_default_ssl_profile_backend

Done

>

> **sh ssl serviceGroup sg1**

Advanced SSL configuration for Back-end SSL Service Group sg1:

Profile Name :ns_default_ssl_profile_backend

Done

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

4. You must now apply the configuration in ssl_config.txt to the current configuration, so that your non-default settings are applied after the upgrade.

batch -f /<path to the batch file>/ssl_config.txt

5. After applying the configuration, the output changes as follows:

> **show ssl vserver v1**

Advanced SSL configuration for VServer v1:

Profile Name :profile-002

1) CertKey Name: mycertkey Server Certificate

Done

> **show ssl service svc1**

Advanced SSL configuration for Back-end SSL Service svc1:

Profile Name :profile-001

Done

> **show ssl serviceGroup sg1**

Advanced SSL configuration for Back-end SSL Service Group sg1:

Profile Name :profile-003

Done

> **show ssl profile profile-002**

1) Name: profile-002 (Front-End)

SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: DISABLED TLSv1.2: DISABLED

Client Auth: ENABLED Client Cert Required: Mandatory

Use only bound CA certificates: DISABLED

Strict CA checks: NO

Session Reuse: ENABLED Timeout: 120 seconds

DH: DISABLED

Ephemeral RSA: ENABLED Refresh Count: 0

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Deny SSL Renegotiation ALL

Non FIPS Ciphers: DISABLED

Cipher Redirect: DISABLED

SSL Redirect: DISABLED

Send Close-Notify: YES

Push Encryption Trigger: Always

PUSH encryption trigger timeout: 1 ms

SNI: DISABLED

Strict Host Header check for SNI enabled SSL sessions: NO

Push flag: 0x0 (Auto)

SSL quantum size: 8 kB

Encryption trigger timeout 100 mS

Encryption trigger packet count: 45

Subject/Issuer Name Insertion Format: Unicode

ECC Curve: P_256, P_384, P_224, P_521

1) Cipher Name: AES Priority :1

Description: Predefined Cipher Alias

1) Vserver Name: v1

Done

> **show ssl profile profile-001**

1) Name: profile-001 (Back-End)

SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: DISABLED TLSv1.2: DISABLED

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Server Auth: ENABLED

Use only bound CA certificates: DISABLED

Strict CA checks: NO

Session Reuse: ENABLED Timeout: 120 seconds

Deny SSL Renegotiation ALL

Non FIPS Ciphers: DISABLED

Send Close-Notify: YES

Push Encryption Trigger: Always

PUSH encryption trigger timeout: 1 ms

Push flag: 0x0 (Auto)

SSL quantum size: 8 kB

Encryption trigger timeout 100 mS

Encryption trigger packet count: 45

ECC Curve: P_256, P_384, P_224, P_521

1) Cipher Name: AES Priority :1

Description: Predefined Cipher Alias

1) Service Name: svc1

Done

> **show ssl profile profile-003**

1) Name: profile-003 (Back-End)

SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: DISABLED TLSv1.2: DISABLED

Server Auth: DISABLED

Use only bound CA certificates: DISABLED

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Strict CA checks: NO

Session Reuse: ENABLED Timeout: 120 seconds

Deny SSL Renegotiation ALL

Non FIPS Ciphers: DISABLED

Send Close-Notify: YES

Push Encryption Trigger: Always

PUSH encryption trigger timeout: 1 ms

Push flag: 0x0 (Auto)

SSL quantum size: 8 kB

Encryption trigger timeout 100 mS

Encryption trigger packet count: 45

ECC Curve: P_256, P_384, P_224, P_521

1) Cipher Name: ALL Priority :1

Description: Predefined Cipher Alias

1) Service Name: sg1

Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Appendix B: Default Front-End and Back-End SSL Profile Settings

Jun 30, 2016

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Certificates

Jan 01, 2018

An SSL certificate, which is an integral part of any SSL transaction, is a digital data form (X509) that identifies a company (domain) or an individual. The certificate has a public key component that is visible to any client that wants to initiate a secure transaction with the server. The corresponding private key, which resides securely on the NetScaler appliance, is used to complete asymmetric key (or public key) encryption and decryption.

You can obtain an SSL certificate and key in either of the following ways:

- From an authorized certificate authority (CA), such as VeriSign
- By generating a new SSL certificate and key on the NetScaler appliance

Alternately, you can use an existing SSL certificate on the appliance.

Caution: Citrix recommends that you use certificates obtained from authorized CAs, such as VeriSign, for all your SSL transactions. Certificates generated on the NetScaler appliance should be used for testing purposes only, not in any live deployment.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Obtaining a Certificate from a Certificate Authority

Mar 30, 2015

A certificate authority (CA) is an entity that issues digital certificates for use in public key cryptography. Certificates issued or signed by a CA are automatically trusted by applications, such as web browsers, that conduct SSL transactions. These applications maintain a list of the CAs that they trust. If the certificate being used for the secure transaction is signed by any of the trusted CAs, the application proceeds with the transaction.

To obtain an SSL certificate from an authorized CA, you must create a private key, use that key to create a certificate signing request (CSR), and submit the CSR to the CA. The only special characters allowed in the file names are underscore and dot.

Creating a Private Key

The private key is the most important part of a digital certificate. By definition, this key is not to be shared with anyone and should be kept securely on the NetScaler appliance. Any data encrypted with the public key can be decrypted only by using the private key.

The appliance supports two encryption algorithms, RSA and DSA, for creating private keys. You can submit either type of private key to the CA. The certificate that you receive from the CA is valid only with the private key that was used to create the CSR, and the key is required for adding the certificate to the NetScaler.

Caution: Be sure to limit access to your private key. Anyone who has access to your private key can decrypt your SSL data.
All SSL certificates and keys are stored in the /nsconfig/ssl folder on the appliance. For added security, you can use the Data Encryption Standard (DES) or triple DES (3DES) algorithm to encrypt the private key stored on the appliance.
Note: The length of the SSL key name allowed includes the length of the absolute path name if the path is included in the key name.

To create an RSA private key by using the command line interface

At the command prompt, type the following command:

create ssl rsakey <keyFile> <bits> [-exponent (3 | F4)] [-keyform (DER | PEM)]

Example

```
> create ssl rsakey Key-RSA-1 2048 -exponent F4 -keyform PEM
```

To create a DSA private key by using the command line interface

At the command prompt, type the following command:

create ssl dsakey <keyfile> <bits> [-keyform (DER | PEM)]

Example

```
> create ssl dsakey Key-DSA-1 2048 -keyform PEM
```

To create an RSA private key by using the configuration utility

Navigate to Traffic Management > SSL and, in the SSL Keys group, select Create RSA Key, and create an RSA key.

To create an DSA private key by using the configuration utility

Navigate to Traffic Management > SSL and, in the SSL Keys group, select Create DSA Key, and create a DSA key.

Creating a Certificate Signing Request

The certificate signing request (CSR) is a collection of information, including the domain name, other important company details, and the private key to be used to create the certificate. To avoid generating an invalid certificate, make sure that the details you provide are accurate.

To create a certificate signing request by using the command line interface

At the command prompt, type the following command:

create ssl certreq <reqFile> -keyFile <input_filename> | -fipsKeyName <string>) [-keyForm (DER | PEM) {-PEMPassPhrase }] -countryName <string> -stateName <string> -organizationName <string> [-organizationUnitName <string>] [-localityName <string>] [-commonName <string>] [-emailAddress <string>] {-challengePassword } [-companyName <string>]

Example

```
create ssl certreq csreq1 -keyfile ramp -keyform PEM -countryName US -stateName Florida -localityName FortLauderdale  -organizationName Citrix -organizationUnitName DNPG
```

To create a certificate signing request by using the configuration utility

Navigate to Traffic Management > SSL and, in the SSL Certificates group, select Create Certificate Signing Request (CSR), and create a CSR.

Submitting the CSR to the CA

Most CAs accept certificate submissions by email. The CA will return a valid certificate to the email address from which you submit the CSR.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Importing Existing Certificates and Keys

Jan 01, 2018

If you want to use certificates and keys that you already have on other secure servers or applications in your network, you can export them, and then import them to the NetScaler appliance. You might have to convert exported certificates and keys before you can import them to the NetScaler appliance.

For the details of how to export certificates from secure servers or applications in your network, see the documentation of the server or application from which you want to export.

Note: For installation on the NetScaler appliance, key and certificate names cannot contain spaces or special characters other than those supported by the UNIX file system. Follow the appropriate naming convention when you save the exported key and certificate.
A certificate and private key pair is commonly sent in the PKCS#12 format. The NetScaler supports PEM and DER formats for certificates and keys. To convert PKCS#12 to PEM or DER, or PEM or DER to PKCS#12, see Converting the Format of SSL Certificates for Import or Export.

The NetScaler appliance does not support PEM keys in PKCS#8 format. However, you can convert these keys to a supported format by using the OpenSSL interface, which you can access from the NetScaler command line or the configuration utility. Before you convert the key, you need to verify that the private key is in PKCS#8 format. Keys in PKCS#8 format typically start with the following text:

-----BEGIN ENCRYPTED PRIVATE KEY-----

leuSSZQZKgrgUQ==

-----END ENCRYPTED PRIVATE KEY-----

To open the OpenSSL interface from the command line interface

1. Open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on to the appliance by using the administrator credentials.
3. At the command prompt, type shell.
4. At the shell prompt type openssl.

To open the ssl interface from the configuration utility

Navigate to Traffic Management > SSL and, in the Tools group, select OpenSSL interface.

To convert a non-supported PKCS#8 key format to an encrypted supported key format by using the OpenSSL interface

At the OpenSSL prompt, type one of the following commands, depending on whether the non-supported key format is of type rsa or dsa:

- rsa- in <PKCS#8 Key Filename> -des3 -out <encrypted Key Filename>
- dsa -in <PKCS#8 Key Filename> -des3 -out <encrypted Key Filename>

To convert a non-supported PKCS#8 key format to an unencrypted key format by using the OpenSSL interface

At the OpenSSL prompt, type the following commands, depending on whether the non-supported key format is of type rsa

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

or dsa:

- rsa -in <PKCS#8 Key Filename> -out <unencrypted Key Filename>
- dsa -in <PKCS#8 Key Filename> -out <unencrypted Key Filename>

Parameters for converting an unsupported key format to a supported key format

<PKCS#8 Key Filename>

The input file name of the incompatible PKCS#8 private key.

<encrypted Key Filename>

The output file name of the compatible encrypted private key in PEM format.

<unencrypted Key Filename>

The output file name of the compatible unencrypted private key in PEM format.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Generating a Test Certificate

Dec 21, 2015

> **Note**
>
> To generate a server test certificate, see Generating a Server Test Certificate.

The NetScaler appliance has a built in CA tools suite that you can use to create self-signed certificates for testing purposes.

Caution: Because these certificates are signed by the NetScaler itself, not by an actual CA, you should not use them in a production environment. If you attempt to use a self-signed certificate in a production environment, users will receive a "certificate invalid" warning each time the virtual server is accessed.
The NetScaler supports creation of the following types of certificates

- Root-CA certificates
- Intermediate-CA certificates
- End-user certificates
  - server certificates
  - client certificates

Before generating a certificate, create a private key and use that to create a certificate signing request (CSR) on the appliance. Then, instead of sending the CSR out to a CA, use the NetScaler CA Tools to generate a certificate.

For details on how to create a private key and a CSR, see Obtaining a Certificate from a Certificate Authority.

To create a certificate by using a wizard

1. Navigate to Traffic Management > SSL.
2. In the details pane, under Getting Started, select the wizard for the type of certificate that you want to create.
3. Follow the instructions on the screen.

To create a Root-CA certificate by using the command line interface

At the command prompt, type the following command:

create ssl cert <certFile> <reqFile> <certType> [-keyFile <input_filename>] [-keyform (**DER** | **PEM**)] [-days <positive_integer>]

Example

In the following example, csreq1 is the CSR and rsa1 is the private key that was created earlier.

```
> create ssl cert cert1 csreq1 ROOT_CERT -keyFile rsa1 -keyForm PEM -days 365
  Done
```

To create an Intermediate-CA certificate certificate by using the command line interface

At the command prompt, type the following command:

create ssl cert <certFile> <reqFile> <certType> [-keyFile <input_filename>] [-keyform (**DER** | **PEM**)] [-days <positive_integer>] [-certForm (**DER** | **PEM**)] [-CAcert <input_filename>] [-CAcertForm (**DER** | **PEM**)] [-CAkey <input_filename>] [-CAkeyForm (**DER** | **PEM**)] [-CAserial <output_filename>]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

In the following example, csr1 is the CSR created earlier. Cert1 and rsakey1 are the certificate and corresponding key of the self-signed (root-CA) certificate, and pvtkey1 is the private key of the intermediate-CA certificate.

```
> create ssl cert certsy csr1 INTM_CERT -CAcert cert1 -CAkey rsakey1 -CAserial 23
Done
> create ssl rsakey pvtkey1 2048 -exponent F4 -keyform PEM
```

To create a Root-CA certificate by using the configuration utility

Navigate to Traffic Management > SSL and, in the Getting Started group, select Root-CA Certificate Wizard, and configure a root CA certificate.

To create an Intermediate-CA certificate certificate by using the configuration utility

Navigate to Traffic Management > SSL and, in the Getting Started group, select Intermediate-CA Certificate Wizard, and configure an intermediate CA certificate.

Creating an End-User Certificate

An end-user certificate can be a client certificate or a server certificate. To create a test end-user certificate, specify the Intermediate CA certificate or the self-signed root-CA certificate.

Note: To create an end-user certificate for production use, specify a trusted CA certificate and send the CSR to a certificate authority (CA).

To create a test end-user certificate by using the command line interface

create ssl cert <certFile> <reqFile> <certType> [-keyFile <input_filename>] [-keyform (DER | PEM)] [-days<positive_integer>] [-certForm (DER | PEM)] [-CAcert <input_filename>] [-CAcertForm (DER | PEM)] [-CAkey<input_filename>] [-CAkeyForm (DER | PEM)] [-CAserial <output_filename>]

Example

If there is no intermediate certificate, use the certificate (cert1) and private key (rsakey1) values of the root-CA certificate in CAcert and CAkey.

```
> create ssl cert cert12 csr1 SRVR_CERT -CAcert cert1 -CAkey rsakey1 -CAserial 23
```

Done

If there is an intermediate certificate, use the certificate (certsy) and private key (pvtkey1) values of the intermediate certificate in CAcert and CAkey.

```
> create ssl cert cert12 csr1 SRVR_CERT -CAcert certsy -CAkey pvtkey1 -CAserial 23
```

Done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Generating a Diffie-Hellman (DH) Key

Dec 21, 2015

The Diffie-Hellman (DH) key exchange is a way for two parties involved in an SSL transaction that have no prior knowledge of each other to agree upon a shared secret over an insecure channel. This secret can then be converted into cryptographic keying material for mainly symmetric key cipher algorithms that require such a key exchange.

This feature is disabled by default and should be specifically configured to support ciphers that use DH as the key exchange algorithm.

> **Note**
>
> Generating a 2048-bit DH key may take a long time (up to 30 minutes).

To generate a DH key by using the command line interface

At the command prompt, type the following command:

create ssl dhparam <dhFile> [<bits>] [-gen (2 | 5)]

Example COPY

```
create ssl dhparam Key-DH-1 512 -gen 2
```

To generate a DH key by using the configuration utility

Navigate to **Traffic Management** > **SSL** and, in the **Tools** group, select **Create Diffie-Hellman (DH) key**, and generate a DH key.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding a Group of SSL Certificates

Feb 17, 2017

If the server certificate is issued by an intermediate CA that is not recognized by standard web browsers as a trusted CA, the CA certificate(s) must be sent to the client with the server's own certificate. Otherwise, the browser terminates the SSL session because it fails to authenticate the server certificate.

There are two ways to add the server and intermediate certificates:

- Create a certificate set that contains the chain of certificates.
- Create a chain of certificates manually by adding and linking the certificates individually.

Adding and Linking a Certificate Set

Note: This feature is not supported on the NetScaler FIPS platform and in a cluster setup.
Instead of adding and linking individual certificates, you can now group a server certificate and up to nine intermediate certificates in a single file, and then specify the file's name when adding a certificate-key pair. Before you do so, make sure that the following prerequisites are met.

- The certificates in the file are in the following order:
  - Server certificate (should be the first certificate in the file)
  - Optionally, a server key
  - Intermediate certificate 1 (ic1)
  - Intermediate certificate 2 (ic2)
  - Intermediate certificate 3 (ic3), and so on
    Note: Intermediate certificate files are created for each intermediate certificate with the name "<certificatebundlename>.pem_ic<*n*>" where n is between 1 and 9. For example, bundle.pem_ic1, where bundle is the name of the certificate set and ic1 is the first intermediate certificate in the set.
- Bundle option is selected.
- No more than nine intermediate certificates are present in the file.

The file is parsed and the server certificate, intermediate certificates, and server key (if present) are identified. First, the server certificate and key are added. Then, the intermediate certificates are added, in the order in which they were added to the file, and linked accordingly.

An error is reported if any of the following conditions exist:

- A certificate file for one of the intermediate certificates already exists on the appliance.
- The key is placed before the server certificate in the file.
- An intermediate certificate is placed before the server certificate.
- Intermediate certificates are not in placed in the file in the same order as they are created.
- No certificates are present in the file.
- A certificate is not in the proper PEM format.
- The number of intermediate certificates in the file exceeds nine.

To add a certificate set by using the command line interface

At the command prompt, type the following commands to create a certificate set and verify the configuration:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. add ssl certKey <certkeyName> -cert <string> -key <string> -bundle (YES | NO)
2. show ssl certKey
3. show ssl certlink

Example

In the following example, the certificate set (bundle.pem) contains the following files:

- server certificate (bundle) linked to bundle_ic1
- First intermediate certificate (bundle_ic1) linked to bundle_ic2
- Second intermediate certificate (bundle_ic2) linked to bundle_ic3
- Third intermediate certificate (bundle_ic3)

```
> add ssl certKey bundle -cert bundle.pem -key bundle.pem -bundle yes

> show ssl certkey

1) Name: bundle
Cert Path: /nsconfig/ssl/bundle.pem
Format: PEM
Status: Valid, Days to expiration:10415
Certificate Expiry Monitor: DISABLED

2) Name: bundle_ic1
Cert Path: /nsconfig/ssl/bundle.pem_ic1
Format: PEM
Status: Valid, Days to expiration:10415
Certificate Expiry Monitor: DISABLED

3) Name: bundle_ic2
Cert Path: /nsconfig/ssl/bundle.pem_ic2
Format: PEM
Status: Valid, Days to expiration:10415
Certificate Expiry Monitor: DISABLED

4) Name: bundle_ic3
Cert Path: /nsconfig/ssl/bundle.pem_ic3
Format: PEM
Status: Valid, Days to expiration:10415
Certificate Expiry Monitor: DISABLED
Done

> show ssl certlink

1) Cert Name: bundle CA Cert Name: bundle_ic1
2) Cert Name: bundle_ic1 CA Cert Name: bundle_ic2
3) Cert Name: bundle_ic2 CA Cert Name: bundle_ic3
Done
```

To add a certificate set by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Traffic Management > SSL > Certificates.
2. In the SSL Certificates pane, click Install.
3. In the Install Certificate dialog box, type the details, such as the certificate and key file name, and then select Certificate Bundle.
4. Click Install, and then click Close.

Creating a Chain of Certificates

Updated: 2013-08-20

Instead of using a set of certificates (a single file), you can create a chain of certificates. The chain links the server certificate to its issuer (the intermediate CA). For this approach to work, the intermediate CA certificate file must already be installed on the NetScaler appliance, and one of the certificates in the chain must be trusted by the client application. For example, link Cert-Intermediate-A to Cert-Intermediate-B, where Cert-Intermediate-B is linked to Cert-Intermediate-C, which is a certificate trusted by the client application.

Note: The NetScaler supports sending a maximum of 10 certificates in the chain of certificates sent to the client (one server certificate and nine CA certificates).

To create a certificate chain by using the command line interface

At the command prompt, type the following commands to create a certificate chain and verify the configuration. (Repeat the first command for each new link in the chain.)

- link ssl certkey <certKeyName> <linkCertKeyName>
- show ssl certlink

Example

```
> link ssl certkey siteAcertkey CAcertkey
 Done

> show ssl certlink

linked certificate:
     1) Cert Name: siteAcertkey CA Cert Name: CAcertkey
 Done
```

To create a certificate chain by using the configuration utility

1. Navigate to Traffic Management > SSL > Certificates.
2. Select a server certificate, and in the Action list, select Link, and specify a CA certificate name.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Displaying a Certificate Chain

Jun 18, 2014

A certificate contains the name of the issuing authority and the subject to whom the certificate is issued. To validate a certificate, you must look at the issuer of that certificate and confirm if you trust the issuer. If you do not trust the issuer, you must see who issued the issuer certificate. Go up the chain till you reach the root CA certificate or an issuer that you trust.

As part of the SSL handshake, when a client requests a certificate, the NetScaler appliance presents a certificate and the chain of issuer certificates that are present on the appliance. An administrator can view the certificate chain for the certificates present on the appliance and install any missing certificates.

To view the certificate chain for the certificates present on the appliance by using the command line

At the command prompt, type:

show ssl certchain <cert_name>

Examples

There are 3 certificates: c1, c2, and c3. Certificate c1 is signed by c2, c2 is signed by c3, and c3 is the root CA certificate. The following examples illustrate the output of the **show ssl certchain c1** command in different scenarios.

1. Scenario 1:
   - Certificate c2 is linked to c1, and c3 is linked to c2.
   - Certificate c3 is a root CA certificate.

   If you run the following command, the certificate links up to the root CA certificate are displayed.

   ```
   > show ssl certchain c1
   Certificate chain details of certificate name c1 are:
   1) Certificate name: c2          linked; not a root certificate
   2) Certificate name: c3          linked; root certificate
   Done
   ```
2. Scenario 2:
   - Certificate c2 is linked to c1.
   - Certificate c2 is not a root CA certificate.

   If you run the following command, information that certificate c3 is a root CA certificate but is not linked to c2 is displayed.

   ```
   > show ssl certchain c1
   Certificate chain  details of certificate name c1 are:
   1) Certificate Name: c2          linked; not a root certificate
   2) Certificate Name: c3          not linked; root certificate
   Done
   ```
3. Scenario 3:
   - Certificate c1, c2, and c3 are not linked but are present on the appliance.

   If you run the following command, information about all the certificates starting with the issuer of certificate c1 is displayed and it is specified that the certificates are not linked.

   ```
   > show ssl certchain c1
   Certificate chain details of certificate name c1 are:
   1) Certificate Name: c2          not linked; not a root certificate
   ```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

2) Certificate Name: c3 not linked; root certificate

Done

4. Scenario 4:
   - Certificate c2 is linked to c1.
   - Certificate c3 is not present on the appliance.

   If you run the following command, information about the certificate linked to c1 is displayed and you are prompted to add a certificate with the subject name specified in c2. In this case, the user is asked to add the root CA certificate c3.

   > show ssl certchain c1

   Certificate chain details of certificate name c1 are:

   1) Certificate Name: c2 linked; not a root certificate

   2) Certificate Name: /C=IN/ST=ka/O=netscaler/CN=test

   Action: Add a certificate with this subject name.

   Done

5. Scenario 5:
   - A certificate is not linked to certificate c1 and the issuer certificate of c1 is not present on the appliance.

   If you run the following command, you are prompted to add a certificate with the subject name in certificate c1.

   > sh ssl certchain c1

   Certificate chain details of certificate name c1 are:

   1) Certificate Name: /ST=KA/C=IN

   Action: Add a certificate with this subject name.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Generating a Server Test Certificate

Aug 20, 2013

The NetScaler appliance allows you to create a test certificate for server authentication by using a GUI wizard in the configuration utility. A server certificate is used to authenticate and identify a server in an SSL handshake. A server certificate is generally issued by a trusted CA and is sent out by the server to a client who uses it to authenticate the server.

For issuing a server test certificate, the appliance operates as a CA. This certificate can be bound to an SSL virtual server for authentication in an SSL handshake with a client. This certificate is for testing purposes only. It should not be used in a production environment.

You can install the server test certificate on any virtual server that uses the SSL or the SSL_TCP protocol.

To generate a server test certificate by using the configuration utility

Navigate to Traffic Management > SSL and, in the SSL Certificates group, select Create and Install a Server Test Certificate.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Achieving Perfect Forward Secrecy With DHE

Jun 24, 2015

Generating the DH key is a CPU-intensive operation. In earlier releases, key generation, on a VPX appliance, took a long time because it was done in the software. In earlier releases, key generation is optimized (as defined by NIST in http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf) by setting the dhKeyExpSizeLimit parameter. You can set this parameter for an SSL virtual server or an SSL profile and then bind the profile to a virtual server.

Additionally, to maintain perfect forward secrecy, DH count must ideally be zero. With this enhancement, you can generate a DH key for each transaction (minimum DHcount is 0) without a significant drop in performance, because the operation is optimized. Earlier, the minimum DH count allowed was 500. That is, you could not regenerate the key for up to 500 transactions.

To optimize DH key generation on a VPX appliance by using the command line

At the command prompt, type commands 1 and 2, or type command 3:

1. add ssl profile <name> [-sslProfileType (BackEnd | FrontEnd)] [-dhCount <positive_integer>] [-dh (ENABLED | DISABLED) -dhFile <string>] [-dhKeyExpSizeLimit (ENABLED | DISABLED)]
2. set ssl vserver <vServerName> [-sslProfile <string>]
3. set ssl vserver <vServerName> [-dh (ENABLED | DISABLED) -dhFile <string>] [-dhCount <positive_integer>] [-dhKeyExpSizeLimit (ENABLED | DISABLED)]

To optimize DH key generation on a VPX appliance by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select Enable DH Key Expire Size Limit.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Modifying and Monitoring Certificates and Keys

Apr 28, 2016

To avoid downtime when replacing a certificate-key pair, you can update an existing certificate. If you want to replace a certificate with a certificate that was issued to a different domain, you must disable domain checks before updating the certificate.

To receive notifications about certificates due to expire, you can enable the expiry monitor.

Updating an Existing Server Certificate

When you remove or unbind a certificate from a configured SSL virtual server, or an SSL service, the virtual server or service becomes inactive until a new valid certificate is bound to it. To avoid downtime, you can use the update feature to replace a certificate-key pair that is bound to an SSL virtual server or an SSL service, without first unbinding the existing certificate.

To update an existing certificate-key pair by using the command line interface

At the command prompt, type the following commands to update an existing certificate-key pair and verify the configuration:

- update ssl certkey <certkeyName> -cert <string> -key <string>
- show ssl certKey <certkeyName>

Example

```
> update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem
    -key /nsconfig/ssl/pkey.pem
 Done

> show ssl certkey siteAcertkey
Name: siteAcertkey      Status: Valid
    Version: 3
    Serial Number: 02
    Signature Algorithm: md5WithRSAEncryption
    Issuer: /C=US/ST=CA/L=Santa Clara/O=siteA/OU=Tech
    Validity
        Not Before: Nov 11 14:58:18 2001 GMT
        Not After: Aug 7 14:58:18 2004 GMT
    Subject: /C=US/ST-CA/L=San Jose/O=CA/OU=Security
    Public Key Algorithm: rsaEncryption
    Public Key size: 2048
 Done
```

To update an existing certificate-key pair by using the configuration utility

Navigate to Traffic Management > SSL > Certificates, select a certificate, and click Update.

Disabling Domain Checks

© 1999-2017 Citrix Systems, Inc. All rights reserved.

When an SSL certificate is replaced on the NetScaler, the domain name mentioned on the new certificate should match the domain name of the certificate being replaced. For example, if you have a certificate issued to abc.com, and you are updating it with a certificate issued to def.com, the certificate update fails.

However, if you want the server that has been hosting a particular domain to now host a new domain, you can disable the domain check before updating its certificate.

To disable the domain check for a certificate by using the command line interface

At the command prompt, type the following commands to disable the domain check and verify the configuration:

- update ssl certKey <certkeyName> -noDomainCheck
- show ssl certKey <certkeyName>

Example

```
> update ssl certKey sv -noDomainCheck
 Done
> show ssl certkey sv
 Name: sv
 Cert Path: /nsconfig/ssl/complete/server/server_rsa_512.pem
 Key Path: /nsconfig/ssl/complete/server/server_rsa_512.ky
 Format: PEM
 Status: Valid,   Days to expiration:9349
 Certificate Expiry Monitor: DISABLED
 Done
```

To disable the domain check for a certificate by using the configuration utility

1. Navigate to Traffic Management > SSL > Certificates, select a certificate, and click Update.
2. Select No Domain Check.

Enabling the Expiry Monitor

An SSL certificate is valid for a specific period of time. A typical deployment includes multiple virtual servers that process SSL transactions, and the certificates bound to them can expire at different times. An expiry monitor configured on the NetScaler appliance creates entries in the appliance's syslog and nsaudit logs when a certificate configured on the appliance is due to expire.

If you want to create SNMP alerts for certificate expiration, you must configure them separately.

To enable an expiry monitor for a certificate by using the command line interface

At the command prompt, type the following commands to enable an expiry monitor for a certificate and verify the configuration:

- set ssl certKey <certkeyName> [-expiryMonitor (ENABLED | DISABLED) [-notificationPeriod <positive_integer>]]
- show ssl certKey <certkeyName>

Example

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> set ssl certKey sv -expiryMonitor ENABLED –notificationPeriod 60
  Done

> show ssl certkey sv
Name: sv
 Cert Path: /nsconfig/ssl/complete/server/server_rsa_512.pem
 Key Path: /nsconfig/ssl/complete/server/server_rsa_512.ky
 Format: PEM
 Status: Valid,   Days to expiration:9349
 Certificate Expiry Monitor: ENABLED
 Expiry Notification period: 60 days
  Done
```

To enable an expiry monitor for a certificate by using the configuration utility

1. Navigate to Traffic Management > SSL > Certificates, select a certificate, and click Update.
2. Select Notify When Expires, and optionally specify a notification period.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using Global Site Certificates

Jan 01, 2018

A global site certificate is a special-purpose server certificate whose key length is greater than 128 bits. A global site certificate consists of a server certificate and an accompanying intermediate-CA certificate. You must import the global site certificate and its key from the server to the NetScaler appliance.

How Global Site Certificates Work

Export versions of browsers use 40-bit encryption to initiate connections to SSL Web-servers. The server responds to connection requests by sending its certificate. The client and server then decide on an encryption strength based on the server certificate type:

- If the server certificate is a normal certificate and not a global site certificate, the export client and server complete the SSL handshake and uses 40-bit encryption for data transfer.
- If the server certificate is a global site certificate (and if the export client feature is supported by the browser), the export client automatically upgrades to 128-bit encryption for data transfer.

If the server certificate is a global site certificate, the server sends its certificate, along with the accompanying intermediate-CA certificate. The browser first validates the intermediate-CA certificate by using one of the Root-CA certificates that are normally included in web browsers. Upon successful validation of the intermediate-CA certificate, the browser uses the intermediate-CA certificate to validate the server certificate. Once the server is successfully validated, the browser renegotiates (upgrades) the SSL connection to 128-bit encryption.

With Microsoft's Server Gated Cryptography (SGC), if the Microsoft IIS server is configured with an SGC certificate, export clients that receive the certificate renegotiate to use128-bit encryption.

Importing a Global Site Certificate

To import a global site certificate, first export the certificate and server key from the Web server. Global site certificates are generally exported in some binary format, therefore, before importing the global site certificate, convert the certificate and key to the PEM format.

To import a global site certificate

1. Using a text editor, copy the server certificate and the accompanying intermediate-CA certificate into two separate files.

   The individual PEM encoded certificate will begin with the header ----- BEGIN CERTIFICATE----- and end with the trailer -----END CERTIFICATE-----.

2. Use an SFTP client to transfer the server certificate, intermediate-CA certificate, and server-key to the NetScaler.
3. Use the following OpenSSL command to identify the server certificate and intermediate-CA certificate from the two separate files.
   Note: You can launch the OpenSSL interface from the configuration utility. In the navigation pane, click SSL. In the details pane, under Tools, click Open SSL interface.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> openssl x509 -in >path of the CA cert file< -text X509v3 Basic Constraints: **CA:TRUE**

X509v3 Key Usage:

Certificate Sign, CRL Sign

Netscape Cert Type:

SSL CA, S/MIME CA

> openssl x509 -in >path of the server certificate file< -text

X509v3 Basic Constraints:

CA:FALSE

Netscape Cert Type:

SSL Server

4. At the FreeBSD shell prompt, enter the following command:

   openssl x509 -in cert.pem -text | more

   Where **cert.pem** is one of the two certificate files.

   Read the **Subject** field in the command output. For example,

   Subject: C=US, ST=Oregon, L=Portland, O=mycompany, Inc., OU=IT, CN=www.mycompany.com
   If the CN field in the Subject matches the domain-name of your Web site, then this is the server certificate and the other certificate is the accompanying intermediate-CA certificate.

5. Use the server certificate and its private key) to create a certificate key pair on the NetScaler. For details on creating a certificate-key pair on the NetScaler, see Adding or Updating a Certificate-Key Pair.
   .
6. Add the intermediate-CA certificate on the NetScaler. Use the server certificate you created in step 4 to sign this intermediate certificate. For details on creating an Intermediate-CA certificate on the NetScaler, see Generating a Test Certificate.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Converting the Format of SSL Certificates for Import or Export

Aug 20, 2013

A NetScaler appliance supports the PEM and DER formats for SSL certificates. Other applications, such as client browsers and some external secure servers, require various public key cryptography standard (PKCS) formats. The NetScaler can convert the PKCS#12 format (the personal information exchange syntax standard) to PEM or DER format for importing a certificate to the appliance, and can convert PEM or DER to PKCS#12 for exporting a certificate. For additional security, conversion of a file for import can include encryption of the private key with the DES or DES3 algorithm.

Note: If you use the configuration utility to import a PKCS#12 certificate, and the password contains a dollar sign ($), backquote (`), or escape (\) character, the import may fail. If it does, the ERROR: Invalid password message appears. If you must use a special character in the password, be sure to prefix it with an escape character (\) unless all imports are performed by using the NetScaler command line.

To convert the format of a certificate by using the command line interface

At the command prompt, type the following command:

Convert ssl pkcs12 <outfile> [-import [-pkcs12File <inputFilename>] [-des | -des3] [-export [-certFile <inputFilename>] [-keyFile <inputFilename>]] During the operation, you are prompted to enter an import password or an export password. For an encrypted file, you are also prompted to enter a passphrase.

Example

```
convert ssl pkcs12 Cert-Import-1.pem -import -pkcs12File Cert-Import-1.pfx -des

convert ssl pkcs12 Cert-Client-1.pfx -export -certFile Cert-Client-1 -keyFile Key-Client-1
```

To convert the format of a certificate by using the configuration utility

Navigate to Traffic Management > SSL and, in the Tools group, select Import PKCS#12 or Export PKCS#12.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Stricter Control on Client Certificate Validation

May 02, 2017

The NetScaler appliance accepts valid Intermediate-CA certificates if they are issued by a single Root-CA. That is, if only the Root-CA certificate is bound to the virtual server, and any intermediate certificate sent with the client certificate is validated by that Root-CA, the appliance trusts the certificate chain and the handshake is successful.

However, if a client sends a chain of certificates in the handshake, none of the intermediate certificates can be validated by using a CRL or OCSP responder unless that certificate is bound to the SSL virtual server. Therefore, even if one of the intermediate certificates is revoked, the handshake is successful. As part of the handshake, the SSL virtual server sends the list of CA certificates that are bound to it. For stricter control, you can configure the SSL virtual server to accept only a certificate that is signed by one of the CA certificates bound to that virtual server. To do so, you must enable the ClientAuthUseBoundCAChain setting in the SSL profile bound to the virtual server. The handshake fails if the client certificate is not signed by one of the CA certificates bound to the virtual server.

For example, say two client certificates, clientcert1 and clientcert2, are signed by the intermediate certificates Int-CA-A and Int-CA-B, respectively. The intermediate certificates are signed by the root certificate Root-CA. Int-CA-A and Root-CA are bound to the SSL virtual server. In the default case (ClientAuthUseBoundCAChain disabled), both clientcert1 and clientcert2 are accepted. However, if ClientAuthUseBoundCAChain is enabled, only clientcert1 is accepted by the NetScaler appliance

To enable stricter control on client certificate validation by using the command line

At the NetScaler command prompt, type:

set ssl profile <name> -ClientAuthUseBoundCAChain Enabled

To enable stricter control on client certificate validation by using the configuration utility

1. Navigate to **System** > **Profiles**, select the **SSL Profiles** tab, and create an SSL profile, or select an existing profile.
2. Select **Enable Client Authentication using bound CA Chain**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Certificate Revocation Lists

Jan 01, 2018

A certificate issued by a CA typically remains valid until its expiration date. However, in some circumstances, the CA may revoke the issued certificate before the expiration date (for example, when an owner's private key is compromised, a company's or individual's name changes, or the association between the subject and the CA changes).

A Certificate Revocation List (CRL) identifies invalid certificates by serial number and issuer.

Certificate authorities issue CRLs on a regular basis. You can configure the NetScaler appliance to use a CRL to block client requests that present invalid certificates.

If you already have a CRL file from a CA, add that to the NetScaler. You can configure refresh options. You can also configure the NetScaler to sync the CRL file automatically at a specified interval, from either a web location or an LDAP location. The NetScaler supports CRLs in either the PEM or the DER file format. Be sure to specify the file format of the CRL file being added to the NetScaler.

If you have used the NetScaler as a CA to create certificates that are used in SSL deployments, you can also create a CRL to revoke a particular certificate. This feature can be used, for example, to ensure that self-signed certificates that are created on the NetScaler are not used either in a production environment or beyond a particular date.

Note:
By default, CRLs are stored in the /var/netscaler/ssl directory on the NetScaler appliance.

To manage certificate revocation lists, see the following sections in this topic:

- Creating a CRL on the NetScaler
- Adding an Existing CRL to the NetScaler
- Configuring CRL Refresh Parameters
- Synchronizing CRLs
- Performing Client Authentication by using a Certificate Revocation List

Creating a CRL on the NetScaler

Since you can use the NetScaler appliance to act as a certificate authority and create self-signed certificates, you can also revoke certificates that you have created and certificates whose CA certificate you own.

The appliance must revoke invalid certificates before creating a CRL for those certificates. The appliance stores the serial numbers of revoked certificates in an index file and updates the file each time it revokes a certificate. The index file is automatically created the first time a certificate is revoked.

To revoke a certificate or create a CRL by using the command line interface

At the command prompt, type the following command:

create ssl crl <CAcertFile> <CAkeyFile> <indexFile> (-revoke <input_filename> | -genCRL <output_filename>)

Example

```
create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -revoke Invalid-1

create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -genCRL CRL-1
```

To revoke a certificate or create a CRL by using the configuration utility

1. Navigate to Traffic Management > SSL and, in the Getting Started group, select CRL Management.
2. Enter the certificate details and, in the Choose Operation list, select Revoke Certificate or Generate CRL.

Adding an Existing CRL to the NetScaler

Before you configure the CRL on the NetScaler appliance, make sure that the CRL file is stored locally on the NetScaler. In the case of an HA setup, the CRL file must be present on both NetScaler appliances, and the directory path to the file must be the same on both appliances.

To add a CRL on the NetScaler by using the command line

At the command prompt, type the following commands to add a CRL on the NetScaler and verify the configuration:

- add ssl crl <crlName> <crlPath> [-inform (DER | PEM)]
- show ssl crl [<crlName>]

Example

```
> add ssl crl crl-one /var/netscaler/ssl/CRL-one -inform PEM
 Done
> show ssl crl crl-one
      Name: crl-one   Status: Valid,  Days to expiration: 29
      CRL Path: /var/netscaler/ssl/CRL-one
      Format: PEM     CAcert: samplecertkey
      Refresh: DISABLED
      Version: 1
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Signature Algorithm: sha1WithRSAEncryption
    Issuer:  C=US,ST=California,L=Santa Clara,O=NetScaler Inc.,OU=SSL Acceleration,CN=www.ns.com/emailAddress=support@netscaler.com
    Last_update:Jun 15 10:53:53 2010 GMT
    Next_update:Jul 15 10:53:53 2010 GMT

1)    Serial Number: 00
      Revocation Date:Jun 15 10:51:16 2010 GMT
 Done
```

To add a CRL on the NetScaler by using the configuration utility

Navigate to Traffic Management > SSL > CRL, and add a CRL.

Configuring CRL Refresh Parameters

A CRL is generated and published by a Certificate Authority periodically or, in some cases, immediately after a particular certificate is revoked. Citrix recommends that you update CRLs on the NetScaler appliance regularly, for protection against clients trying to connect with certificates that are not valid.

The NetScaler can refresh CRLs from a web location or an LDAP directory. When you specify refresh parameters and a web location or an LDAP server, the CRL does not have to be present on the local hard disk drive at the time you execute the command. The first refresh stores a copy on the local hard disk drive, in the path specified by the CRL File parameter. The default path for storing the CRL is /var/netscaler/ssl.
Note: In release 10.0 and later, the method for refreshing a CRL is not included by default. You must explicitly specify an HTTP or LDAP method. If you are upgrading from an earlier release to release 10.0 or later, you must add a method and run the command again.

To configure CRL autorefresh by using the command line

At the command prompt, type the following commands to configure CRL auto refresh and verify the configuration the following commands to configure CRL auto refresh and verify the configuration:

- set ssl crl <crlName> [-refresh (ENABLED | DISABLED)] [-CAcert <string>] [-url <URL | -server <ip_addr|ipv6_addr>] [-method HTTP | (LDAP [-baseDN <string>] [-bindDN <string>] [-scope (Base | One)] [-password <string>] [-binary (YES | NO)])] [-port <port>] [-interval <interval>]
- show ssl crl [<crlName>]

Example

```
Set CRL crl1  -refresh enabled -method ldap -inform DER -CAcert ca1 -server 10.102.192.192 -port 389 -scope base -baseDN "cn=clnt_rsa4_multicert_der,ou=eng,o=ns,c=in" -time 00:01

set ssl crl crl1 -refresh enabled -method http -cacert ca1 -port 80 -time 00:10 -url http://10.102.192.192/crl/ca1.crl

> sh crl

1)      Name: crl1       Status: Valid,     Days to expiration: 355
        CRL Path: /var/netscaler/ssl/crl1
        Format: PEM      CAcert: ca1
        Refresh: ENABLED         Method: HTTP
        URL: http://10.102.192.192/crl/ca1.crl              Port:80
        Refresh Time: 00:10
        Last Update: Successful, Date:Tue Jul  6 14:38:13 2010
Done
```

To configure CRL autorefresh using LDAP or HTTP by using the configuration utility

1. Navigate to Traffic Management > SSL > CRL.
2. Open a CRL, and select Enable CRL Auto Refresh.
   Note: If the new CRL has been refreshed in the external repository before its actual update time as specified by the Last Update time field of the CRL, you should immediately refresh the CRL on the NetScaler.
   To view the last update time, select the CRL, and click Details.

Synchronizing CRLs

The NetScaler appliance uses the most recently distributed CRL to prevent clients with revoked certificates from accessing secure resources.

If CRLs are updated often, the NetScaler needs an automated mechanism to fetch the latest CRLs from the repository. You can configure the NetScaler to update CRLs automatically at a specified refresh interval.

The NetScaler maintains an internal list of CRLs that need to be updated at regular intervals. At these specified intervals, the appliance scans the list for CRLs that need to be updated, connects to the remote LDAP server or HTTP server, retrieves the latest CRLs, and then updates the local CRL list with the new CRLs.

Note: If CRL check is set to mandatory when the CA certificate is bound to the virtual server, and the initial CRL refresh fails, all client-authentication connections with the same issuer as the CRL are rejected as REVOKED until the CRL is successfully refreshed.
You can specify the interval at which the CRL refresh should be carried out. You can also specify the exact time.

To synchronize CRL autorefresh by using the command line interface

At the command prompt, type the following command:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

set ssl crl <crlName> [-interval <interval>] [-day <integer>] [-time <HH:MM>]
Example

```
set ssl crl CRL-1 -refresh ENABLE -interval
MONTHLY -days 10 -time 12:00
```

To synchronize CRL refresh by using the configuration utility

1. Navigate to Traffic Management > SSL > CRL.
2. Open a CRL, select enable CRL Auto Refresh, and specify the interval.

Performing Client Authentication by using a Certificate Revocation List

If a certificate revocation list (CRL) is present on a NetScaler appliance, a CRL check is performed regardless of whether performing the CRL check is set to mandatory or optional. The success or failure of a handshake depends on a combination of the following factors:

- Rule for CRL check
- Rule for client certificate check
- State of the CRL configured for the CA certificate

The following table lists the results of the possible combinations for a handshake involving a revoked certificate.

Table 1. Result of a Handshake with a Client Using a Revoked Certificate

Rule for CRL Check	Rule for Client Certificate Check	State of the CRL Configured for the CA certificate	Result of a Handshake with a Revoked Certificate
Optional	Optional	Missing	Success
Optional	Mandatory	Missing	Success
Optional	Mandatory	Present	Failure
Mandatory	Optional	Missing	Success
Mandatory	Mandatory	Missing	Failure
Mandatory	Optional	Present	Success
Mandatory	Mandatory	Present	Failure
Optional/Mandatory	Optional	Expired	Success
Optional/Mandatory	Mandatory	Expired	Failure

Note:

- The CRL check is optional by default. To change from optional to mandatory or vice-versa, you must first unbind the certificate from the SSL virtual server, and then bind it again after changing the option.
- In the output of the sh ssl vserver command, OCSP check: optional implies that a CRL check is also optional. The CRL check settings are displayed in the output of the sh ssl vserver command only if CRL check is set to mandatory. If CRL check is set to optional, the CRL check details do not appear.

To configure CRL check by using the command line interface

At the command prompt, type the following command:
bind ssl vserver <vServerName> -certkeyName <string> [(-CA -crlCheck (Mandatory | Optional))]
Example

```
bind ssl vs v1 -certkeyName ca -CA -crlCheck mandatory
sh ssl vserver
> sh ssl vs v1

Advanced SSL configuration for VServer v1:

DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Non FIPS Ciphers: DISABLED
SNI: DISABLED
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
Push Encryption Trigger: Always
Send Close-Notify: YES

1) CertKey Name: ca CA Certificate CRLCheck: Mandatory CA_Name Sent

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
```

To configure CRL check by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open an SSL virtual server.
2. Click in the Certificates section.
3. Select a certificate and, in the OCSP and CRL Check list, select CRL Mandatory.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Monitoring Certificate Status with OCSP

Jul 07, 2016

Online Certificate Status Protocol (OCSP) is an Internet protocol that is used to determine the status of a client SSL certificate. NetScaler appliances support OCSP as defined in RFC 2560. OCSP offers significant advantages over certificate revocation lists (CRLs) in terms of timely information. Up-to-date revocation status of a client certificate is especially useful in transactions involving large sums of money and high-value stock trades. It also uses fewer system and network resources. NetScaler implementation of OCSP includes request batching and response caching.

To monitor certificate status with OCSP, see the following sections:

- NetScaler Implementation of OCSP
- OCSP Request Batching
- OCSP Response Caching
- Configuring an OCSP Responder

NetScaler Implementation of OCSP

OCSP validation on a NetScaler appliance begins when the appliance receives a client certificate during an SSL handshake. To validate the certificate, the NetScaler creates an OCSP request and forwards it to the OCSP responder. To do so, the NetScaler uses a locally configured URL. The transaction is in a suspended state until the NetScaler evaluates the response from the server and determines whether to allow the transaction or reject it. If the response from the server is delayed beyond the configured time and no other responders are configured, the NetScaler will allow the transaction or display an error, depending on whether the OCSP check was set to optional or mandatory, respectively.

The NetScaler supports batching of OCSP requests and caching of OCSP responses to reduce the load on the OCSP responder and provide faster responses.

OCSP Request Batching

Each time the NetScaler receives a client certificate, it sends a request to the OCSP responder. To help avoid overloading the OCSP responder, the NetScaler can query the status of more than one client certificate in the same request. For this to work efficiently, a timeout needs to be defined so that processing of a single certificate is not inordinately delayed while waiting to form a batch.

OCSP Response Caching

Caching of responses received from the OCSP responder enables faster responses to the clients and reduces the load on the OCSP responder. Upon receiving the revocation status of a client certificate from the OCSP responder, the NetScaler caches the response locally for a predefined length of time. When a client certificate is received during an SSL handshake, the NetScaler first checks its local cache for an entry for this certificate. If an entry is found that is still valid (within the cache timeout limit), it is evaluated and the client certificate is accepted or rejected. If a certificate is not found, the NetScaler sends a request to the OCSP responder and stores the response in its local cache for a configured length of time.

Configuring an OCSP Responder

Configuring OCSP involves adding an OCSP responder, binding the OCSP responder to a certification authority (CA) certificate, and binding the certificate to an SSL virtual server. If you need to bind a different certificate to an OCSP responder that has already been configured, you need to first unbind the responder and then bind the responder to a different certificate.

To add an OCSP responder by using the command line interface

At the command prompt, type the following commands to configure OCSP and verify the configuration:

- add ssl ocspResponder <name> -url <URL> [-cache (ENABLED | DISABLED)[-cacheTimeout <positive_integer>]] [-batchingDepth <positive_integer>][-batchingDelay <positive_integer>] [-resptimeout <positive_integer>] [-responderCert <string> | -trustResponder] [-producedAtTimeSkew <positive_integer>][-signingCert <string>][-useNonce (YES | NO)][-insertClientCert(YES | NO)]
- bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <positive_integer>]
- bind ssl vserver <vServerName>@ (-certkeyName <string> (CA [-ocspCheck (Mandatory | Optional)]))
- show ssl ocspResponder [<name>]

Example

```
add ssl ocspResponder ocsp_responder1 -url "http:// www.myCA.org:80/ocsp/" -cache ENABLED -cacheTimeout 30 -batchingDepth 8 -batchingDelay 100 -resptimeout 100 -responderCert responder_ce
bind ssl certKey ca_cert -ocspResponder ocsp_responder1 -priority 1
bind ssl vserver vs1 -certkeyName ca_cert -CA -ocspCheck Mandatory

sh ocspResponder ocsp_responder1
1)Name: ocsp_responder1
URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
Caching: Enabled        Timeout: 30 minutes
Batching: 8 Timeout: 100 mS
HTTP Request Timeout: 100mS
Request Signing Certificate: sign_cert
Response Verification: Full, Certificate: responder_cert
ProducedAt Time Skew: 300 s
Nonce Extension: Enabled
 Client Cert Insertion: Enabled
Done

show certkey ca_cert
Name: ca_cert     Status: Valid,   Days to expiration:8907
Version: 3
…
1)  VServer name: vs1      CA Certificate
1)  OCSP Responder name: ocsp_responder1     Priority: 1
Done

sh ssl vs vs1
Advanced SSL configuration for VServer vs1:
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
DH: DISABLED
...
1) CertKey Name: ca_cert CA Certificate OCSPCheck: Mandatory
1) Cipher Name: DEFAULT
 Description: Predefined Cipher Alias
Done
```

To modify an OCSP responder by using the command line interface

You cannot modify the responder name. All other parameters can be changed using the set ssl ocspResponder command.

At the command prompt, type the following commands to set the parameters and verify the configuration:

- set ssl ocspResponder <name> [-url <URL>] [-cache (ENABLED | DISABLED)] [-cacheTimeout <positive_integer>] [-batchingDepth <positive_integer>] [-batchingDelay <positive_integer>] [-resptimeout <positive_integer>] [-responderCert <string> | -trustResponder][-producedAtTimeSkew <positive_integer>][-signingCert <string>] [-useNonce (YES | NO)]
- unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]
- bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <positive_integer>]
- show ssl ocspResponder [<name>]

To configure an OCSP responder by using the configuration utility

1. Navigate to Traffic Management > SSL > OCSP Responder, and configure an OCSP responder.
2. Navigate to Traffic Management > SSL > Certificates, select a certificate, and in the Action list, select OCSP Bindings. Bind an OCSP responder.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers, open a virtual server, and click in the Certificates section to bind a CA certificate.
4. Optionally, select select OCSP Mandatory.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Client Authentication

Jan 01, 2018

In a typical SSL transaction, the client that is connecting to a server over a secure connection checks the validity of the server by checking the server's certificate before initiating the SSL transaction. In some cases, however, you might want to configure the server to authenticate the client that is connecting to it.

With client authentication enabled on an SSL virtual server, the NetScaler appliance asks for the client certificate during the SSL handshake. The appliance checks the certificate presented by the client for normal constraints, such as the issuer signature and expiration date.

Note: For the NetScaler to verify issuer signatures, the certificate of the CA that issued the client certificate must be installed on the NetScaler and bound to the virtual server that the client is transacting with.
If the certificate is valid, the NetScaler allows the client to access all secure resources. But if the certificate is invalid, the NetScaler drops the client request during the SSL handshake.

The NetScaler verifies the client certificate by first forming a chain of certificates, starting with the client certificate and ending with the root CA certificate for the client (for example, VeriSign). The root CA certificate may contain one or more intermediate CA certificates (if the client certificate is not directly issued by the root CA).

Before you enable client authentication on the NetScaler, make sure that a valid client certificate is installed on the client. Then, enable client authentication for the virtual server that will handle the transactions. Finally, bind the certificate of the CA that issued the client certificate to the virtual server on the NetScaler.

Note: A NetScaler MPX appliance supports a certificate-key pair size from 512 to 4096 bits. The certificate must be signed by using one of the following hash algorithms:

- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

On an SDX appliance, if an SSL chip is assigned to a VPX instance, the certificate-key pair size support of an MPX appliance applies. Otherwise, the normal certificate-key pair size support of a VPX instance applies.

A NetScaler virtual appliance (VPX instance) supports certificates of at least 512 bits, up to the following sizes:

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service
- 4096-bit CA certificate
- 2048-bit certificate on the physical server
- 2048-bit client certificate (if client authentication is enabled on the virtual server)

To configure client authentication, see the following topics in this section:

- Providing the Client Certificate
- Enabling Client-Certificate-Based Authentication
- Binding CA Certificates to the Virtual Server

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Providing the Client Certificate

Before you configure client authentication, a valid client certificate must installed on the client. A client certificate includes details about the specific client system that will create secure sessions with the NetScaler appliance. Each client certificate is unique and should be used by only one client system.

Whether you obtain the client certificate from a CA, use an existing client certificate, or generate a client certificate on the NetScaler appliance, you must convert the certificate to the correct format. On the NetScaler, certificates are stored in either the PEM or DER format and must be converted to PKCS#12 format before they are installed on the client system. After converting the certificate and transferring it to the client, system, make sure that it is installed on that system and configured for the client application that will be part of the SSL transactions (for example, the web browser).

For instructions on how to convert a certificate from PEM or DER format to PKCS#12 format, see Converting SSL Certificates for Import or Export.

For instructions on how to generate a client certificate, see Generating a Test Certificate.

.

Enabling Client-Certificate-Based Authentication

By default, client authentication is disabled on the NetScaler appliance, and all SSL transactions proceed without authenticating the client. You can configure client authentication to be either optional or mandatory as part of the SSL handshake.

If client authentication is optional, the NetScaler requests the client certificate but proceeds with the SSL transaction even if the client presents an invalid certificate. If client authentication is mandatory, the NetScaler terminates the SSL handshake if the SSL client does not provide a valid certificate.

Caution: Citrix recommends that you define proper access control policies before changing client-certificate-based authentication check to optional.
Note: Client authentication is configured for individual SSL virtual servers, not globally.

To enable client-certificate-based authentication by using the command line interface

At the command prompt, type the following commands to enable the client-certificate-based authentication and verify the configuration:

- set ssl vserver <vServerName> [-clientAuth (ENABLED | DISABLED)] [-clientCert (MANDATORY | OPTIONAL)]
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vssl -clientAuth ENABLED -clientCert Mandatory
 Done
> show ssl vserver vssl

    Advanced SSL configuration for VServer vssl:
    DH: DISABLED
    Ephemeral RSA: ENABLED        Refresh Count: 0
    Session Reuse: ENABLED        Timeout: 120 seconds
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Cipher Redirect: DISABLED
    SSLv2 Redirect: DISABLED
    ClearText Port: 0
    Client Auth: ENABLED    Client Cert Required: Mandatory
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)    CertKey Name: sslckey   Server Certificate

1)    Policy Name: client_cert_policy  Priority: 0

1)    Cipher Name: DEFAULT
    Description: Predefined Cipher Alias
Done
```

To enable client-certificate-based authentication by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select Client Authentication, and in the Client Certificate list, select Mandatory.

Binding CA Certificates to the Virtual Server

A CA whose certificate is present on the NetScaler appliance must issue the client certificate used for client authentication. You must bind this certificate to the NetScaler virtual server that will carry out client authentication.

You must bind the CA certificate to the SSL virtual server in such a way that the NetScaler can form a complete certificate chain when it verifies the client certificate. Otherwise, certificate chain formation fails and the client is denied access even if its certificate is valid.

You can bind CA certificates to the SSL virtual server in any order. The NetScaler forms the proper order during client certificate verification.

For example, if the client presents a certificate issued by **CA_A**, where **CA_A** is an intermediate CA whose certificate is issued by **CA_B**, whose certificate is in turn issued by a trusted root CA, **Root_CA**, a chain of certificates that contain all three of these certificates must be bound to the virtual server on the NetScaler.

For instructions on binding one or more certificates to the virtual server, see Binding the Certificate-key Pair to the SSL Based Virtual Server.

For instructions on creating a chain of certificates, see the "Creating a Chain of Certificates" section in Adding a Group of SSL Certificates.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Customizing the SSL Configuration

Aug 20, 2015

Once your basic SSL configuration is operational, you can customize some of the parameters that are specific to the certificates being used in SSL transactions. You can also enable and disable session reuse and client authentication, and you can configure redirect responses for cipher and SSLv2 protocol mismatches.

You can also customize SSL settings for two NetScaler appliances in a High Availability configuration, and you can synchronize settings, certificates and keys across those appliances.

These settings will depend on your network deployment and the type of clients you expect will connect to your servers.

To customize the SSL configuration, see the following sections:

- Configuring Diffie-Hellman (DH) Parameters
- Configuring Ephemeral RSA
- Configuring Session Reuse
- Configuring Cipher Redirection
- Configuring SSLv2 Redirection
- Configuring SSL Protocol Settings
- Configuring Close-Notify
- Configuring ECDHE Ciphers
- Configuring a Common Name on an SSL Service or Service Group for Server Certificate Authentication
- Configuring Advanced SSL Settings
- Synchronizing Configuration Files in a High Availability Setup
- Managing Server Authentication
- Configuring User-Defined Cipher Groups on the NetScaler Appliance

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Diffie-Hellman (DH) Parameters

Dec 21, 2015

If you are using ciphers on the NetScaler that require a DH key exchange to set up the SSL transaction, enable DH key exchange on the NetScaler and configure other settings based on your network.

To list the ciphers for which DH parameters must be set by using the NetScaler command line, type: sh cipher DH.

To list the ciphers for which DH parameters must be set by using the configuration utility, navigate to Traffic Management > SSL > Cipher Groups, and double-click DH.

For details on how to enable DH key exchange, see Generating a Diffie-Hellman (DH) Key.

To configure DH Parameters by using the command line interface

At the command prompt, type the following commands to configure DH parameters and verify the configuration:

- set ssl vserver <vserverName> -dh <Option> -dhCount <RefreshCountValue> -filepath <string>
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-server -dh ENABLED  -dhFile /nsconfig/ssl/ns-server.cert -dhCount 1000
  Done
> show ssl vserver vs-server

      Advanced SSL configuration for VServer vs-server:
      DH: ENABLED
      Ephemeral RSA: ENABLED        Refresh Count: 1000
      Session Reuse: ENABLED        Timeout: 120 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: DISABLED
      ClearText Port: 0
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)     Cipher Name: DEFAULT
       Description: Predefined Cipher Alias
 Done
```

To configure DH Parameters by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select Enable DH Param, and specify a refresh count and file path.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Ephemeral RSA

Aug 20, 2013

Ephemeral RSA allows export clients to communicate with the secure server even if the server certificate does not support export clients (1024-bit certificate). If you want to prevent export clients from accessing the secure web object and/or resource, you need to disable ephemeral RSA key exchange.

By default, this feature is enabled on the NetScaler appliance, with the refresh count set to zero (infinite use).

Note:
The ephemeral RSA key is automatically generated when you bind an export cipher to an SSL or TCP-based SSL virtual server or service. When you remove the export cipher, the eRSA key is not deleted but reused at a later date when another export cipher is bound to an SSL or TCP-based SSL virtual server or service. The eRSA key is deleted when the system restarts.

To configure Ephemeral RSA by using the command line interface

At the command prompt, type the following commands to configure ephemeral RSA and verify the configuration:

- set ssl vserver <vServerName> -eRSA (enabled | disabled) -eRSACount <positive_integer>
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-server -eRSA ENABLED -eRSACount 1000
 Done
> show ssl vserver vs-server

      Advanced SSL configuration for VServer vs-server:
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 1000
      Session Reuse: ENABLED        Timeout: 120 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: DISABLED
      ClearText Port: 0
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)     Cipher Name: DEFAULT
       Description: Predefined Cipher Alias
 Done
```

To configure Ephemeral RSA by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select Enable Ephemereal RSA, and specify a refresh count.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Session Reuse

Aug 20, 2013

For SSL transactions, establishing the initial SSL handshake requires CPU-intensive public key encryption operations. Most handshake operations are associated with the exchange of the SSL session key (client key exchange message). When a client session is idle for some time and is then resumed, the SSL handshake is typically conducted all over again. With session reuse enabled, session key exchange is avoided for session resumption requests received from the client.

Session reuse is enabled on the NetScaler appliance by default. Enabling this feature reduces server load, improves response time, and increases the number of SSL transactions per second (TPS) that can be supported by the server.

To configure session reuse by using the command line interface

At the command prompt, type the following commands to configure session reuse and verify the configuration:

- set ssl vserver <vServerName> -sessReuse (ENABLED | DISABLED) -sessTimeout <positive_integer>
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-ssl -sessreuse enabled -sesstimeout 600
 Done

> show ssl vserver vs-ssl

      Advanced SSL configuration for VServer vs-ssl:
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 1000
      Session Reuse: ENABLED        Timeout: 600 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: DISABLED
      ClearText Port: 0
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)    CertKey Name: Auth-Cert-1      Server Certificate

1)    Cipher Name: DEFAULT
      Description: Predefined Cipher Alias
 Done
```

To configure session reuse by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select Enable Session Reuse, and specify a time for which to keep the session active.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Cipher Redirection

Aug 20, 2013

During the SSL handshake, the SSL client (usually a web browser) announces the suite of ciphers that it supports, in the configured order of cipher preference. From that list, the SSL server then selects a cipher that matches its own list of configured ciphers.

If the ciphers announced by the client do not match those configured on the SSL server, the SSL handshake fails, and the failure is announced by a cryptic error message displayed in the browser. These messages rarely mention the exact cause of the error.

With cipher redirection, you can configure an SSL virtual server to deliver accurate, meaningful error messages when an SSL handshake fails. When SSL handshake fails, the NetScaler appliance redirects the user to a previously configured URL or, if no URL is configured, displays an internally generated error page.

To configure cipher redirection by using the command line interface

At the command prompt, type the following commands to configure cipher redirection and verify the configuration:

- set ssl vserver <vServerName> -cipherRedirect < ENABLED | DISABLED> -cipherURL < URL>
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-ssl -cipherRedirect ENABLED -cipherURL http://redirectURI
 Done
> show ssl vserver vs-ssl

      Advanced SSL configuration for VServer vs-ssl:
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 1000
      Session Reuse: ENABLED        Timeout: 600 seconds
      Cipher Redirect: ENABLED      Redirect URL: http://redirectURI
      SSLv2 Redirect: DISABLED
      ClearText Port: 0
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)    CertKey Name: Auth-Cert-1      Server Certificate

1)    Cipher Name: DEFAULT
      Description: Predefined Cipher Alias
 Done
```

To configure cipher redirection by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. In the SSL Parameters section, select Enable Cipher Redirect, and specify a redirect URL.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SSLv2 Redirection

Aug 20, 2013

For an SSL transaction to be initiated, and for successful completion of the SSL handshake, the server and the client should agree on an SSL protocol that both of them support. If the SSL protocol version supported by the client is not acceptable to the server, the server does not go ahead with the transaction, and an error message is displayed.

You can configure the server to display a precise error message (user-configured or internally generated) advising the client on the next action to be taken. Configuring the server to display this message requires that you set up SSLv2 redirection.

To configure SSLv2 redirection by using the command line interface

At the command prompt, type the following commands to configure SSLv2 redirection and verify the configuration:

- set ssl vserver <vServerName> [-sslv2Redirect (ENABLED | DISABLED) [-sslv2URL <URL>]]
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-ssl -sslv2Redirect ENABLED -sslv2URL http://sslv2URL
 Done
> show ssl vserver vs-ssl

      Advanced SSL configuration for VServer vs-ssl:
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 1000
      Session Reuse: ENABLED        Timeout: 600 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: ENABLED Redirect URL: http://sslv2URL
      ClearText Port: 0
      Client Auth: DISABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)    CertKey Name: Auth-Cert-1      Server Certificate

1)    Cipher Name: DEFAULT
      Description: Predefined Cipher Alias
 Done
```

To configure SSLv2 redirection by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select SSLv2 Redirect, and specify a URL.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SSL Protocol Settings

May 20, 2015

The NetScaler appliance supports the SSLv2, SSLv3, TLSv1, TLSv1.1, and TLSv1.2 protocols. Each of these can be set on the appliance as required by your deployment and the type of clients that will connect to the appliance.

TLS protocol versions 1.0, 1.1, and 1.2 are more secure than older versions of the TLS/SSL protocol. However, to support legacy systems, many TLS implementations maintain backward compatibility with the SSLv3 protocol. In an SSL handshake, the highest protocol version common to the client and the SSL virtual server configured on the NetScaler appliance is used.

In the first handshake attempt, a TLS client offers the highest protocol version that it supports. If the handshake fails, the client offers a lower protocol version. For example, if a handshake with TLS version 1.1 is not successful, the client attempts to renegotiate by offering the TLSv1.0 protocol. If that attempt is unsuccessful, the client reattempts with the SSLv3 protocol. A “man in the middle” (MITM) attacker can break the initial handshake and trigger renegotiation with the SSLv3 protocol, and then exploit a vulnerability in SSLv3. To mitigate such attacks, you can disable SSLv3 or not allow renegotiation using a downgraded protocol. However, this might not be practical if your deployment includes legacy systems. An alternative is to recognize a signaling cipher suite value (TLS_FALLBACK_SCSV) in the client request.

A TLS_FALLBACK_SCSV value in a client hello message indicates to the virtual server that the client has previously attempted to connect with a higher protocol version and that the current request is a fallback. If the virtual server detects this value, and it supports a version higher than the one indicated by the client, it rejects the connection with a fatal alert. If a TLS_FALLBACK_SCSV is not included in the client hello message, or if the protocol version in the client hello is the highest protocol version supported by the virtual server, the handshake succeeds.

To configure SSL protocol support by using the command line interface

At the command prompt, type the following commands to configure SSL protocol support and verify the configuration:

- set ssl vserver <vServerName> -ssl2 (ENABLED | DISABLED) -ssl3 (ENABLED | DISABLED) -tls1 (ENABLED | DISABLED) -tls11 (ENABLED | DISABLED) -tls12 (ENABLED | DISABLED)
- show ssl vserver <vServerName>

Example

```
> set ssl vserver vs-ssl -tls11 ENABLED -tls12 ENABLED
 Done
> sh ssl vs vs-ssl

    Advanced SSL configuration for VServer vs-ssl:
    DH: DISABLED
    Ephemeral RSA: ENABLED          Refresh Count: 0
    Session Reuse: ENABLED          Timeout: 120 seconds
    Cipher Redirect: DISABLED
    SSLv2 Redirect: DISABLED
    ClearText Port: 0
    Client Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: DISABLED
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    SSLv2: DISABLED       SSLv3: ENABLED    TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2: ENABLED
    Push Encryption Trigger: Always
    Send Close-Notify: YES

    1 bound certificate:
1)   CertKey Name: mycert  Server Certificate

    1 configured cipher:
1)   Cipher Name: DEFAULT
    Description: Predefined Cipher Alias
Done
```

To configure SSL protocol support by using the configuration Utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select a protocol to enable.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Close-Notify

Jan 01, 2018

A close-notify is a secure message that indicates the end of SSL data transmission. A close-notify setting is required at the global level. This setting applies to all virtual servers, services, and service groups. For information about the global setting, see Configuring Advanced SSL Settings.

In addition to the global setting, you can set the close-notify parameter at the virtual server, service, or service group level. You therefore have the flexibility of setting the parameter for one entity and unsetting it for another entity. However, make sure that you set this parameter at the global level. Otherwise, the setting at the entity level does not apply.

To configure close-notify at the entity level by using the command line interface

At the command prompt, type any of the following commands to configure close-notify and verify the configuration:

1. To configure close-notify at the virtual server level, type:
   - set ssl vserver <vServerName> -sendCloseNotify (**YES** | **NO**)
   - show ssl vserver <vServerName>
2. To configure close-notify at the service level, type:
   - set ssl service <serviceName> -sendCloseNotify (**YES** | **NO**)
   - show ssl service <serviceName>
3. To configure close-notify at the service group level, type:
   - set ssl serviceGroup <serviceGroupName> -sendCloseNotify (**YES** | **NO**)
   - show ssl serviceGroup <serviceGroupName>

Example

```
> set ssl vserver sslvsvr -sendCloseNotify YES
 Done
```

To configure close-notify at the entity level by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select Send Close-Notify.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring ECDHE Ciphers

Jan 12, 2017

The Citrix NetScaler VPX (frontend only), MPX, SDX, and MPX 9700 series FIPS appliances support the ECDHE cipher group. On an SDX appliance, if an SSL chip is assigned to a VPX instance, the cipher support of an MPX appliance applies. Otherwise, the normal cipher support of a VPX instance applies. For a complete list of ciphers supported by the NetScaler appliance, see Ciphers Supported by the NetScaler Appliance.

The following table lists the ciphers supported on VPX instances and MPX appliances.

Cipher Suite	VPX	MPX
TLS1-ECDHE-RSA-RC4-SHA	YES	YES
TLS1-ECDHE-RSA-DES-CBC3-SHA	YES	YES
TLS1-ECDHE-RSA-AES128-SHA	YES	YES
TLS1-ECDHE-RSA-AES256-SHA	YES	YES
TLS1.2-ECDHE-RSA-AES128-GCM-SHA256	NO	YES
TLS1.2-ECDHE-RSA-AES256-GCM-SHA384	NO	YES
TLS1.2-ECDHE-RSA-AES-128-SHA256	NO	YES
TLS1.2-ECDHE-RSA-AES-256-SHA384	NO	YES

ECDHE cipher suites use elliptical curve cryptography (ECC). Because of its smaller key size, ECC is especially useful in a mobile (wireless) environment. It is also very useful in an interactive voice response environment, where every millisecond is important. Smaller key sizes save power, memory, bandwidth, and computational cost.

A NetScaler appliance supports the following ECC curves:

- P_256
- P_384
- P_224
- P_521

Note:

- If you upgrade from a build earlier than release 10.1 build 121.10, you must explicitly bind ECC curves to your existing SSL virtual servers or front end services. The curves are bound by default to any virtual servers or front end services that you create after the upgrade.

You can bind an ECC curve to SSL front-end entities only. By default all four curves are bound, in the following order: P_256,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

P_384,P_224,P_521. To change the order, you must first unbind all the curves, and then bind them in the desired order.

To unbind ECC curves and bind an ECC curve to an SSL virtual server by using the command line

At the command prompt, type:

- unbind ssl vserver <vServerName> -eccCurveName ALL
- bind ssl vserver <vServerName> -eccCurveName <eccCurveName>

Example

```
unbind ssl vs v1 –eccCurvename ALL
bind ssl vserver  v1 -eccCurveName P_224
> sh ssl vserver v1

 Advanced SSL configuration for VServer v1:
 DH: DISABLED
 Ephemeral RSA: ENABLED  Refresh Count: 0
 Session Reuse: ENABLED  Timeout: 120 seconds
 Cipher Redirect: DISABLED
 SSLv2 Redirect: DISABLED
 ClearText Port: 0
 Client Auth: DISABLED
 SSL Redirect: DISABLED
 Non FIPS Ciphers: DISABLED
 SNI: DISABLED
 SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED
 Push Encryption Trigger: Always
 Send Close-Notify: YES

 ECC Curve: P_224

1) Cipher Name: DEFAULT
 Description: Predefined Cipher Alias
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Common Name on an SSL Service or Service Group for Server Certificate Authentication

May 13, 2015

In end-to-end encryption with server authentication enabled, you can include a common name in the configuration of an SSL service or service group. The name that you specify is compared to the common name in the server certificate during an SSL handshake. If the two names match, the handshake is successful. If the common names do not match, the common name specified for the service or service group is compared to values in the subject alternative name (SAN) field in the certificate. If it matches one of those values, the handshake is successful. This configuration is especially useful if there are, for example, two servers behind a firewall and one of the servers spoofs the identity of the other. If the common name is not checked, a certificate presented by either server is accepted if the IP address matches.

Note: Only domain name, URL, and email ID DNS entries in the SAN field are compared.

To configure common-name verification for an SSL service or service group by using the command line interface

At the command prompt, type the following commands to specify server authentication with common-name verification and verify the configuration:

1. To configure common name in a service, type:
   - set ssl service <serviceName> -commonName <string> -serverAuth ENABLED
   - show ssl service <serviceName>
2. To configure common name in a service group, type:
   - set ssl serviceGroup <serviceGroupName> -commonName <string> -serverAuth ENABLED
   - show ssl serviceGroup <serviceGroupName>

Example

```
> set ssl service svc1 -commonName xyz.com -serverAuth ENABLED
 Done
> show ssl service svc1
 Advanced SSL configuration for Back-end SSL Service svc1:
 DH: DISABLED
 Ephemeral RSA: DISABLED
 Session Reuse: ENABLED  Timeout: 300 seconds
 Cipher Redirect: DISABLED
 SSLv2 Redirect: DISABLED
 Server Auth: ENABLED Common Name: www.xyz.com
 SSL Redirect: DISABLED
 Non FIPS Ciphers: DISABLED
 SNI: DISABLED
 SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1) CertKey Name: cacert CA Certificate  OCSPCheck: Optional

1) Cipher Name: ALL
 Description: Predefined Cipher Alias
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Done

To configure common-name verification for an SSL service or service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services or Navigate to Traffic Management > Load Balancing > Service Groups, and open a service or service group.
2. In the SSL Parameters section, select Enable Server Authentication, and specify a common name.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Advanced SSL Settings

Sep 20, 2013

Advanced customization of your SSL configuration addresses specific issues. You can use the set ssl parameter command or the configuration utility to specify the following:

- Quantum size to be used for SSL transactions.
- CRL memory size.
- OCSP cache size.
- Deny SSL renegotiation.
- Set the PUSH flag for decrypted, encrypted, or all records.
- Drop requests if the client initiates the handshake for one domain and sends an HTTP request for another domain.
- Set the time after which encryption is triggered.
  Note: The time that you specify applies only if you use the set ssl vserver command or the configuration utility to set timer-based encryption.

To configure advanced SSL settings by using the command line interface

At the command prompt, type the following commands to configure advanced SSL settings and verify the configuration:

- set ssl parameter [-quantumSize <quantumSize>] [-crlMemorySizeMB <positive_integer>] [-strictCAChecks (YES | NO)] [-sslTriggerTimeout <positive_integer>] [-sendCloseNotify (YES | NO)] [-encryptTriggerPktCount <positive_integer>] [-denySSLReneg <denySSLReneg>] [-insertionEncoding (Unicode|UTF-8)] [-ocspCacheSize <positive_integer>][- pushFlag <positive_integer>] [- dropReqWithNoHostHeader (YES | NO)] [-pushEncTriggerTimeout <positive_integer>]
- show ssl parameter

Example

```
> set ssl parameter -quantumSize 8 -crlMemorySizeMB 256 -strictCAChecks no -sslt
iggerTimeout 100 -sendClosenotify no -encryptTriggerPktCount 45 -denySSLReneg no
-insertionEncoding unicode -ocspCacheSize 10 -pushFlag 3 -dropReqWithNoHostHeader YES  -pushEncTriggerTimeout 100 ms
 Done

> show ssl parameter
Advanced SSL Parameters
-----------------------
      SSL quantum size:            8 kB
      Max CRL memory size:         256 MB
      Strict CA checks:            NO
      Encryption trigger timeout   100 mS
      Send Close-Notify            NO
      Encryption trigger packet count:     45
      Deny SSL Renegotiation       NO
      Subject/Issuer Name Insertion Format:  Unicode
      OCSP cache size:     10 MB
      Push flag:    0x3 (On every decrypted and encrypted record)
      Strict Host Header check for SNI enabled SSL sessions:       YES
       PUSH encryption trigger timeout 100 ms
 Done
```

To configure advanced SSL settings by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Navigate to Traffic Management > SSL and, in the Settings group, select Change advanced SSL settings.

PUSH Flag-Based Encryption Trigger Mechanism

The encryption trigger mechanism that is based on the PSH TCP flag now enables you to do the following:

- Merge consecutive packets in which the PSH flag is set into a single SSL record, or ignore the PSH flag.
- Perform timer-based encryption, in which the time-out value is set globally by using the set ssl parameter -pushEncTriggerTimeout <positive_integer> command.

To configure PUSH flag-based encryption by using the command line interface

At the command prompt, type the following commands to configure PUSH flag-based encryption and verify the configuration:

- set ssl vserver <vServerName> [-pushEncTrigger <pushEncTrigger>]
- show ssl vserver

Example

```
Advanced SSL configuration for VServer v1:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 0
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        SSLv2: DISABLED     SSLv3: ENABLED     TLSv1: ENABLED
        Push Encryption Trigger: Always
```

To configure PUSH flag-based encryption by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual servers and open an SSL virtual server.
2. In the SSL Parameters section, from the PUSH Encryption Trigger list, select a value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Synchronizing Configuration Files in a High Availability Setup

Oct 28, 2013

In a high availability (HA) set up, the primary NetScaler appliance in the HA pair automatically synchronizes with the secondary appliance in the pair. In the synchronization process, the secondary copies the primary's /nsconfig/ssl/ directory, which is the default location for storing the certificates and keys for SSL transactions. Synchronization occurs at one-minute intervals and every time a new file is added to the directory.

To synchronize files in a high availability setup by using the command line interface

At the command prompt, type the following command:

sync HA files [<Mode>]

Example

```
sync HA files SSL
```

To synchronize files in a high availability setup by using the configuration utility

Navigate to Traffic Management > SSL and, in the Tools group, select Start SSL certificate, key file synchronization for HA.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Server Authentication

Aug 20, 2013

Since the NetScaler appliance performs SSL offload and acceleration on behalf of a web server, the appliance does not usually authenticate the Web server's certificate. However, you can authenticate the server in deployments that require end-to-end SSL encryption.

In such a situation, the NetScaler becomes the SSL client, carries out a secure transaction with the SSL server, verifies that a CA whose certificate is bound to the SSL service has signed the server certificate, and checks the validity of the server certificate.

To authenticate the server, you must first enable server authentication and then bind the certificate of the CA that signed the server's certificate to the SSL service on the NetScaler. When binding the certificate, you must specify the bind as CA option.

To enable (or disable) server certificate authentication by using the command line interface

At the command prompt, type the following commands to enable server certificate authentication and verify the configuration:

- set ssl service <serviceName> -serverAuth (ENABLED | DISABLED)
- show ssl service <serviceName>

Example

```
> set ssl service ssl-service-1 -serverAuth ENABLED
 Done
> show ssl service ssl-service-1

      Advanced SSL configuration for Back-end SSL Service ssl-service-1:
      DH: DISABLED
      Ephemeral RSA: DISABLED
      Session Reuse: ENABLED        Timeout: 300 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: DISABLED
      Server Auth: ENABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)    Cipher Name: ALL
      Description: Predefined Cipher Alias
 Done
```

To enable (or disable) server certificate authentication by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open an SSL service.
2. In the SSL Parameters section, select Enable Server Authentication, and specify a Common Name.
3. In Advanced Settings, select Certificates, and bind a CA certificate to the service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To bind the CA certificate to the service by using the command line interface

At the command prompt, type the following commands to bind the CA certificate to the service and verify the configuration:

- bind ssl service <serviceName> -certkeyName <string> -CA
- show ssl service <serviceName>

Example

```
> bind ssl service ssl-service-1 -certkeyName samplecertkey -CA
 Done
> show ssl service ssl-service-1

      Advanced SSL configuration for Back-end SSL Service ssl-service-1:
      DH: DISABLED
      Ephemeral RSA: DISABLED
      Session Reuse: ENABLED        Timeout: 300 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: DISABLED
      Server Auth: ENABLED
      SSL Redirect: DISABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)    CertKey Name: samplecertkey    CA Certificate        CRLCheck: Optional

1)    Cipher Name: ALL
      Description: Predefined Cipher Alias
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring User-Defined Cipher Groups on the NetScaler Appliance

Oct 02, 2013

A cipher group is a set of cipher suites that you bind to an SSL virtual server, service, or service group on the NetScaler appliance. A cipher suite comprises a protocol, a key exchange (Kx) algorithm, an authentication (Au) algorithm, an encryption (Enc) algorithm, and a message authentication code (Mac) algorithm. Your appliance ships with a predefined set of cipher groups. When you create a SSL service or SSL service group, the ALL cipher group is automatically bound to it. However, when you create an SSL virtual server or a transparent SSL service, the DEFAULT cipher group is automatically bound to it. In addition, you can create a user-defined cipher group and bind it to an SSL virtual server, service, or service group.

Note: If your MPX appliance does not have any licenses, then only the EXPORT cipher is bound to your SSL virtual server, service, or service group.

To create a user-defined cipher group, first you create a cipher group and then you bind ciphers or cipher groups to this group. If you specify a cipher alias or a cipher group, all the ciphers in the cipher alias or group are added to the user-defined cipher group. You can also add individual ciphers (cipher suites) to a user-defined group. However, you cannot modify a predefined cipher group. Before removing a cipher group, unbind all the cipher suites in the group.

If you bind a cipher group to an SSL virtual server, service, or service group, the ciphers are appended to the existing ciphers that are bound to the entity. To bind a specific cipher group to the entity, you must first unbind the ciphers or cipher group that is bound to the entity and then bind the specific cipher group. For example, to bind only the AES cipher group to an SSL service, you perform the following steps:

1. Unbind the default cipher group ALL that is bound by default to the service when the service is created.
   unbind ssl service <service name> -cipherName ALL

2. Bind the AES cipher group to the service
   bind ssl service <Service name> -cipherName AE

If you want to bind the cipher group DES in addition to AES, at the command prompt, type:

- bind ssl service <service name> -cipherName DES

Note: The free NetScaler virtual appliance supports only the DH cipher group.

To configure a user-defined cipher group by using the command line interface

At the command prompt, type the following commands to add a cipher group, or to add ciphers to a previously created group, and verify the settings:

- add ssl cipher <cipherGroupName>
- bind ssl cipher <cipherGroupName> -cipherName <string>
- show ssl cipher <cipherGroupName>

Example

```
 > add ssl cipher test
 Done
> bind ssl cipher test -cipherName SSLv2
 Done
> show ssl cipher test
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
1)     Cipher Name: SSL2-RC2-CBC-MD5
Description: SSLv2 Kx=RSA      Au=RSA  Enc=RC2(128)  Mac=MD5
2)     Cipher Name: SSL2-RC4-MD5
Description: SSLv2 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=MD5
3)     Cipher Name: SSL2-DES-CBC3-MD5
Description: SSLv2 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=MD5
4)     Cipher Name: SSL2-DES-CBC-MD5
Description: SSLv2 Kx=RSA      Au=RSA  Enc=DES(56)   Mac=MD5
5)     Cipher Name: SSL2-RC4-64-MD5
Description: SSLv2 Kx=RSA      Au=RSA  Enc=RC4(64)   Mac=MD5
6)     Cipher Name: SSL2-EXP-RC4-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA  Enc=RC4(40)   Mac=MD5  Export
7)     Cipher Name: SSL2-EXP-RC2-CBC-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA  Enc=RC2(40)   Mac=MD5  Export
 Done
```

To unbind ciphers from a cipher group by using the command line interface

At the command prompt, type the following commands to unbind ciphers from a user-defined cipher group, and verify the settings:

- show ssl cipher <cipherGroupName>
- unbind ssl cipher <cipherGroupName> -cipherName <string>
- show ssl cipher <cipherGroupName>

Example

```
> show ssl cipher test
1) Cipher Name: SSL2-RC2-CBC-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC2(128) Mac=MD5
2) Cipher Name: SSL2-RC4-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5
3) Cipher Name: SSL2-DES-CBC3-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=3DES(168) Mac=MD5
4) Cipher Name: SSL2-DES-CBC-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=DES(56) Mac=MD5
5) Cipher Name: SSL2-RC4-64-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(64) Mac=MD5
6) Cipher Name: SSL2-EXP-RC4-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 Export
7) Cipher Name: SSL2-EXP-RC2-CBC-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC2(40) Mac=MD5 Export
Done

> unbind ssl cipher test -cipherName SSL2-RC2-CBC-MD5

> show ssl cipher test
1) Cipher Name: SSL2-RC4-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5
2) Cipher Name: SSL2-DES-CBC3-MD5
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Description: SSLv2 Kx=RSA Au=RSA Enc=3DES(168) Mac=MD5
3) Cipher Name: SSL2-DES-CBC-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=DES(56) Mac=MD5
4) Cipher Name: SSL2-RC4-64-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(64) Mac=MD5
5) Cipher Name: SSL2-EXP-RC4-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 Export
6) Cipher Name: SSL2-EXP-RC2-CBC-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC2(40) Mac=MD5 Export
Done
```

To remove a cipher group by using the command line interface

Note: You cannot remove a built-in cipher group. Before removing a user-defined cipher group, make sure that the cipher group is empty.
At the command prompt, type the following commands to remove a user-defined cipher group, and verify the configuration:

- rm ssl cipher <userDefCipherGroupName> [<cipherName> ...]
- show ssl cipher <cipherGroupName>

Example

```
> rm ssl cipher test
Done

> sh ssl cipher test
ERROR: No such resource [cipherGroupName, test]
```

To configure a user-defined cipher group by using the configuration utility

Navigate to Traffic Management > SSL > Cipher Groups, and configure a cipher group.

To bind a cipher group to an SSL virtual server, service, or service group by using the command line interface

At the command prompt, type one of the following:

- bind ssl vserver <vServerName> -cipherName <string>
- bind ssl service <serviceName> -cipherName <string>
- bind ssl serviceGroup <serviceGroupName> -cipherName <string>

Examples

```
> bind ssl vserver ssl_vserver_test -cipherName test
Done
bind ssl service  nshttps -cipherName test
Done
> bind ssl servicegroup  ssl_svc  -cipherName test
Done
```

To bind a cipher group to an SSL virtual server, service, or service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers or navigate to Traffic Management > Load Balancing > Services or navigate to Traffic Management > Load Balancing > Service Groups, and open the virtual server, service, or service group.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

2. In Advanced Settings, select SSL Ciphers, and bind a cipher group to the virtual server, service, or service group.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SSL Actions and Policies

Jul 13, 2017

An SSL policy evaluates incoming traffic and applies a predefined action to requests that match a rule (expression). You have to configure the actions before creating the policies, so that you can specify an action when you create a policy. To put a policy into effect, you must either bind it to a virtual server on the appliance, so that it applies only to traffic flowing through that virtual server, or bind it globally, so that it applies to all traffic flowing through the appliance.

SSL actions define SSL settings that you can apply to the selected requests. You associate an action with one or more policies. Data in client connection requests or responses is compared to a rule specified in the policy, and the action is applied to connections that match the rule (expression).

You can configure classic policies with classic expressions and default syntax policies with default syntax expressions for SSL.
Note: Users who are not experienced in configuring policies at the NetScaler command line usually find using the configuration utility to be considerably easier.
You can associate a user-defined action or a built-in action to a default syntax policy. Classic policies allow only user-defined actions. In default syntax policy, you can also group policies under a policy label, in which case they are applied only when invoked from another policy.

Common uses of SSL actions and policies include per-directory client authentication, support for Outlook web access, and SSL-based header insertions. SSL-based header insertions contain SSL settings required by a server whose SSL processing has been offloaded to the NetScaler appliance.

To configure SSL actions and policies, see the following sections:

- Configuring User-Defined Actions for SSL Policies
- Configuring SSL Policies
- Configuring an SSL Default Syntax Policy
- Configuring Built-in Actions for SSL Default Syntax Policies
- Configuring SSL Policy Labels
- Configuring Per-Directory Client Authentication
- Configuring Support for Outlook Web Access
- Configuring SSL-Based Header Insertion
- Binding SSL Policies to a Virtual Server
- Binding SSL Policies Globally

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring User-Defined Actions for SSL Policies

Jan 01, 2018

SSL policies require that you create an action before creating a policy, so that you can specify the actions when you create the policies. In SSL default syntax policies, you can also use the built-in actions. For more information about built-in actions, see Configuring Built-in Actions for SSL Default Syntax Policies.

To configure an SSL action by using the command line interface

At the command prompt, type the following commands to configure an action and verify the configuration:

- add SSL action <name> -clientAuth(DOCLIENTAUTH | NOCLIENTAUTH) -clientCert (ENABLED | DISABLED) certHeader <string> -clientHeader <string> -clientCertSerialNumber (ENABLED | DISABLED) -certSerialHeader <string> **-clientCertSubject** (ENABLED | DISABLED) -certSubjectHeader <string> -clientCertHash (ENABLED | DISABLED) -certHashHeader <string> -clientCertIssuer (ENABLED | DISABLED) -certIssuerHeader <string> -sessionID (ENABLED | DISABLED) -sessionIDheader <string> -cipher (ENABLED | DISABLED) -cipherHeader <string> -clientCertNotBefore (ENABLED | DISABLED) **-certNotBeforeHeader** <string> -clientCertNotAfter (ENABLED | DISABLED) -certNotAfterHeader <string> -OWASupport (ENABLED | DISABLED)
- show ssl action [<name>]

Example

```
> add ssl action Action-SSL-ClientCert -clientCert ENABLED -certHeader "X-Client-Cert"
 Done
> show ssl action Action-SSL-ClientCert
1)     Name: Action-SSL-ClientCert
       Data Insertion Action:
      Cert Header: ENABLED          Cert Tag: X-Client-Cert
 Done
```

To configure an SSL action by using the configuration utility

Navigate to Traffic Management > SSL > Policies and, on the Actions tab, click Add.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SSL Policies

Sep 26, 2013

Policies on the NetScaler help identify specific connections that you want to process. The processing is based on the actions that are configured for that particular policy. Once you create the policy and configure an action for it, you must either bind it to a virtual server on the NetScaler, so that it applies only to traffic flowing through that virtual server, or bind it globally, so that it applies to all traffic flowing through any virtual server configured on the NetScaler.

The NetScaler SSL feature supports both classic policies and default syntax policies . For a complete description of classic and default syntax expressions, how they work, and how to configure them manually, see Policies and Expresssions.

Note: Users who are not experienced in configuring policies at the NetScaler command line will usually find using the configuration utility considerably easier.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring an SSL Default Syntax Policy

Sep 25, 2014

An SSL default syntax policy defines a control or a data action to be performed on requests. SSL policies can therefore be categorized as control policies and data policies:

- **Control policy**. A control policy uses a control action, such as forcing client authentication.
  Note: In release 10.5 or later, deny SSL renegotiation (denySSLReneg) is set, by default, to ALL. However, control policies, such as CLIENTAUTH, trigger a renegotiation handshake. If you use such policies, you must set denySSLReneg to NO.
- **Data policy**. A data policy uses a data action, such as inserting some data into the request.

The essential components of a policy are an expression and an action. The expression identifies the requests on which the action is to be performed. SSL policies use the default expression syntax or the classic expression syntax. For information about expressions and how to configure them, see .

You can configure a default syntax policy with a built-in action or a user-defined action. You can configure a policy with a built-in action without creating a separate action. However, to configure a policy with a user-defined action, first configure the action and then configure the policy.

You can specify an additional action, called an UNDEF action, to be performed in the event that applying the expression to a request has an undefined result.

To configure an SSL default syntax policy by using the command line interface

At the command prompt, type:

```
add ssl policy <name> -rule <expression> -Action <string> [-undefAction <string>] [-comment <string>]
```

To configure an SSL default syntax policy by using the configuration utility

Navigate to Traffic Management > SSL > Policies and, on the Polices tab, click Add.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Built-in Actions for SSL Default Syntax Policies

Jan 01, 2018

Unless you need only the built-in actions in your policies, you have to create the actions before creating the policies, so that you can specify the actions when you create the policies. The built-in actions are of two types, control actions and data actions. You use control actions in control policies, and data actions in data policies.

The built-in control actions are:

- CLIENTAUTH—Perform client certificate authentication.
- NOCLIENTAUTH—Do not perform client certificate authentication.

The built-in data actions are:

- RESET—Close the connection by sending a RST packet to the client.
- DROP—Drop all packets from the client. The connection remains open until the client closes it.
- NOOP—Forward the packet without performing any operation on it.

You can create user-defined data actions. For example, if you enable client authentication, you can create an SSL action to insert client-certificate data into the request header before forwarding the request to the web server. For more information about user-defined actions, see Configuring User-Defined SSL Actions.

If a policy evaluation results in an undefined state, an UNDEF action is performed. For either a data policy or a control policy, you can specify RESET, DROP, or NOOP as the UNDEF action. For a control policy, you also have the option of specifying CLIENTAUTH or NOCLIENTAUTH.

Examples of built-in actions in a policy

In the following example, if the client sends a cipher other than an EXPORT category cipher, the NetScaler appliance requests client authentication. The client has to provide a valid certificate for a successful transaction.

```
add ssl policy pol1 -rule CLIENT.SSL.CIPHER_EXPORTABLE.NOT -reqAction CLIENTAUTH
```

The following examples assume that client authentication is enabled.

If the version in the certificate provided by the user matches the version in the policy, no action is taken and the packet is forwarded:

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -reqAction NOOP
```

If the version in the certificate provided by the user matches the version in the policy, the connection is dropped:

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -reqAction DROP
```

If the version in the certificate provided by the user matches the version in the policy, the connection is reset:

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -reqAction RESET
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SSL Policy Labels

Aug 20, 2013

Policy labels are holders for policies. A policy label helps in managing a group of policies, called a policy bank, which can be invoked from another policy. SSL policy labels can be control labels or data labels, depending on the type of policies that are included in the policy label. You can add only data policies in a data policy label and only control policies in a control policy label. To create the policy bank, you bind policies to the label and specify the order of evaluation of each policy relative to others in the bank of policies for the policy label. At the NetScaler command line, you enter two commands to create a policy label and bind policies to the policy label. In the configuration utility, you select options from a dialog box.

To create an SSL policy label and bind policies to the label by using the command line interface

At the command prompt, type:

- add ssl policylabel <labelName> -type (CONTROL | DATA)
- bind ssl policylabel <labelName> <policyName> <priority> [<gotoPriorityExpression>] [-invoke (<labelType> <labelName>)]

Example

```
add ssl policylabel cpl1 -type CONTROL
add ssl policylabel dpl1 -type DATA
bind ssl policylabel cpl1 -policyName ctrlpol -priority 1
bind ssl policylabel dpl1 -policyName datapol -priority 1
```

To configure an SSL policy label and bind policies to the label by using the configuration utility

Navigate to Traffic Management > SSL > Policy Labels, and configure an SSL policy label.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Per-Directory Client Authentication

Aug 20, 2013

If you create an action specifying client-side authentication on a per-directory basis, a client identified by a policy associated with the action is not authenticated as part of the initial SSL handshake. Instead, authentication is carried out every time the client wants to access a specific directory on the web server.

For example, if you have multiple divisions in the company, where each division has a folder in which all its files are stored, and you want to know the identity of each client that tries to access files from a particular directory, such as the finance directory, you can enable per-directory client authentication for that directory.

To enable per-directory client authentication, first configure client authentication as an SSL action, and then create a policy that identifies the directory that you want to monitor. When you create the policy, specify your client-authentication action as the action associated with the policy. Then, bind the policy to the SSL virtual server that will receive the SSL traffic.

To create an SSL action and a policy to enable client authentication by using the command line interface

At the command prompt, type the following commands to create an SSL action to enable to client authentication and verify the configuration:

- add ssl action <name> [-clientAuth (DOCLIENTAUTH | NOCLIENTAUTH)]
- show ssl action [<name>]
- add ssl policy <name> -rule <expression> [-action <string>] [-undefAction <string>] [-comment <string>]
- show ssl policy [<name>]

Example

```
> add ssl action ssl-action-1 -clientAuth DOCLIENTAUTH
 Done
> show ssl action ssl-action-1
1)     Name: ssl-action-1
       Client Authentication Action: DOCLIENTAUTH
       Hits: 0
       Undef Hits: 0
       Action Reference Count: 1
 Done
> add ssl policy ssl-pol-1 -rule 'REQ.HTTP.METHOD==GET' -reqaction ssl-action-1
> sh ssl policy ssl-pol-1
       Name: ssl-pol-1
       Rule: REQ.HTTP.METHOD == GET
       Action: ssl-action-1
       UndefAction: Use Global
       Hits: 0
       Undef Hits: 0
 Done
```

To create an SSL action to enable client authentication by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Traffic Management > SSL > Policies and, on the Actions tab, click Add.
2. In the Client Authentication list, select Enabled.

To create and bind an SSL policy to enable client authentication by using the configuration utility

1. Navigate to Traffic Management > SSL and, on the Polices tab, click Add.
2. Navigate to Traffic Management > Load Balancing > Virtual Servers and open an SSL virtual server.
   In Advanced Settings, select SSL Policy, and bind the policy to the virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Support for Outlook Web Access

Aug 20, 2013

If your SSL configuration is offloading SSL transactions from an Outlook Web Access (OWA) server, you must insert a special header field, FRONT-END-HTTPS: ON, in all HTTP requests directed to the OWA servers. This is required for the OWA servers to generate URL links as https:// instead of http://.

When you enable support for OWA on the NetScaler, the header is automatically inserted into the specified HTTP traffic, and you do not need to configure a specific header insertion. Use SSL policies to identify all traffic directed to the OWA server.

Note: You can enable Outlook Web Access support for HTTP-based SSL virtual servers and services only. You cannot apply it to TCP-based SSL virtual servers and services.
To enable OWA support, first configure OWA support as an SSL action, and then create a policy that identifies the virtual servers or services for which you want to enable OWA support. When you create the policy, specify your OWA support action as the action associated with the policy. Then, bind the policy to the SSL virtual server that will receive the SSL traffic.

To create an SSL action and a policy to enable OWA support by using the command line interface

At the command prompt, type the following commands to create an SSL action to enable OWA support and verify the configuration:

- add ssl action <name> -OWASupport (ENABLED | DISABLED)
- show ssl action [<name>]
- add ssl policy <name> -rule <expression> [-action <string>] [-undefAction <string>] [-comment <string>]
- show ssl policy [<name>]

Example

```
> add ssl action ssl-action-2 -OWASupport ENABLED
 Done
> show ssl action ssl-action-2
1)     Name: ssl-action-2
       Type: Data Insertion
       OWA Support: ENABLED
       Hits: 0
       Undef Hits: 0
       Action Reference Count: 1
 Done
> add ssl policy ssl-pol -rule 'REQ.HTTP.METHOD == GET' -reqaction ssl-action-2
Done
> sh ssl policy ssl-pol
      Name: ssl-pol
      Rule: REQ.HTTP.METHOD == GET
      Action: ssl-action-2
      UndefAction: Use Global
      Hits: 0
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Undef Hits: 0

Done

To create an SSL action to enable OWA support by using the configuration utility

1. Navigate to Traffic Management > SSL > Policies and, on the Actions tab, click Add.
2. In the Outlook Web Access list, select Enabled.
   Note: Outlook Web Access support is applicable only for SSL virtual server based configurations and transparent SSL service based configurations and not for SSL configurations with back-end encryption.

To create and bind an SSL policy to enable OWA support by using the configuration utility

Navigate to Traffic Management > SSL > Policies, and add a policy.
In the Action list, select the action that you created earlier. Specify an undefined action, and an expression.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SSL-Based Header Insertion

Aug 20, 2013

Because the NetScaler appliance offloads all SSL-related processing from the servers, the servers receive only HTTP traffic. In some circumstances, the server needs certain SSL information. For example, security audits of recent SSL transactions require the client subject name (contained in an X509 certificate) to be logged on the server.

Such data can be sent to the server by inserting it into the HTTP header as a name-value pair. You can insert the entire client certificate, if required, or only the specific fields from the certificate, such as the subject, serial number, issuer, certificate hash, SSL session ID, cipher suite, or the not-before or not-after date used to determine certificate validity.

You can enable SSL-based insertion for HTTP-based SSL virtual servers and services only. You cannot apply it to TCP-based SSL virtual servers and services. Also, client authentication must be enabled on the SSL virtual server, because the inserted values are taken from the client certificate that is presented to the virtual server for authentication.

To configure SSL-based header insertion, first create an SSL action for each specific set of information to be inserted, and then create policies that identify the connections for which you want to insert the information. As you create each policy, specify the action that you want associated with the policy. Then, bind the policies to the SSL virtual servers that will receive the SSL traffic.

The following example uses default syntax policies. In the following example, a control policy (ctrlpol) is created to perform client authentication if a request is received for the URL /testsite/file5.html. A data policy (datapol) is created to perform an action (act1) if client authentication is successful, and an SSL action (act1) is added to insert the certificate details and issuer's name in the request before forwarding the request. For other URLs, client authentication is disabled. The policies are then bound to an SSL virtual server (ssl_vserver) that receives the SSL traffic.

Command-line example of configuring SSL-based header insertion

Example

```
> add ssl action act1 -clientCert ENABLED -certHeader mycert -clientcertissuer ENABLED -certIssuerHeader myissuer
> add ssl policy datapol -rule HTTP.REQ.URL.EQ(\"/testsite/file5.html\") -action act1
> add ssl policy ctrlpol -rule HTTP.REQ.URL.EQ(\"/testsite/file5.html\") -action CLIENTAUTH
> bind ssl vserver ssl_vserver -policyName ctrlpol -priority 1
> bind ssl vserver ssl_vserver -policyName datapol -priority 1
 Done
```

To configure SSL-based header insertion by using the configuration utility

1. Navigate to Traffic Management > SSL > Policies.
2. In the details pane, on the Actions tab, click Add.
3. In the Create SSL Action dialog box, set the following parameters:
   - Name*
   - Client Certificate
   - Certificate Tag
   - Client Certificate Issuer
   - Issuer Tag

   * A required parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Click Create, and then click Close.
5. On the tab, click Add to create a control policy.
6. In the Create SSL Policy dialog box, set the following parameters:
   - Name*
   - Expression
   - Request Action

   * A required parameter

7. Click Create, and then click Close.
8. Create a data policy by repeating steps 5 through 7.
9. In the navigation pane, expand SSL Offload, and then click Virtual Servers.
10. In the details pane, from the list of virtual servers, select the virtual server to which you want to bind the SSL policies, and then click Open.
11. In the Configure Virtual Server (SSL Offload) dialog box, click SSL Settings, and then click SSL Policies.
12. In the Bind/Unbind SSL Policies dialog box, click Insert Policy. Under Policy Name, select the policy that you created in steps 5 through 7.
13. Click OK, and then click Close. A message appears in the status bar, stating that the policy has been bound successfully.
14. Repeat steps 12 and 13 and select the policy that you created in step 8.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding SSL Policies to a Virtual Server

Aug 20, 2013

The SSL policies that are configured on the NetScaler appliance need to be bound to a virtual server that intercepts traffic directed to the virtual server. If the incoming data matches any of the rules configured in the SSL policy, the policy is triggered and the action associated with it is carried out.

You can also bind SSL policies globally or to custom bind points on the NetScaler appliance. For more information about binding policies on the appliance, see .

To bind an SSL policy to a virtual server by using the command line interface

At the command prompt, type the following command to bind an SSL policy to a virtual server and verify the configuration:

- bind ssl vserver <vServerName> -policyName <string> [-priority <positive_integer>]
- show ssl vserver <vServerName>

Example

```
> bind ssl vserver vs-server -policyName ssl-policy-1 -priority 10
 Done
> show ssl vserver vs-server

      Advanced SSL configuration for VServer vs-server:
      DH: DISABLED
      Ephemeral RSA: ENABLED        Refresh Count: 1000
      Session Reuse: ENABLED        Timeout: 120 seconds
      Cipher Redirect: DISABLED
      SSLv2 Redirect: DISABLED
      ClearText Port: 80
      Client Auth: DISABLED
      SSL Redirect: ENABLED
      SSL-REDIRECT Port Rewrite: ENABLED
      Non FIPS Ciphers: DISABLED
      SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)    Policy Name: ssl-policy-1       Priority: 10

1)    Cipher Name: DEFAULT
      Description: Predefined Cipher Alias
 Done
```

To bind an SSL policy to a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open an SSL virtual server.
2. In Advanced Settings, select SSL Policy, Click in the SSL Policy section to bind to the virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding SSL Policies Globally

Aug 20, 2013

Globally bound policies are evaluated after all policies bound to services, virtual servers, or other NetScaler bind points are evaluated.

To globally bind an SSL policy by using the command line interface

At the command prompt, type the following command to bind a global SSL policy and verify the configuration:

- bind ssl global - policyName <string> [- priority <positive_integer>]
- show ssl global

Example

```
> bind ssl global -policyName Policy-SSL-2 -priority 90
Done
> sh ssl global
1) Name: Policy-SSL-2 Priority: 90
2) Name: Policy-SSL-1 Priority: 100
Done
```

To bind a global SSL policy by using the configuration utility

1. Navigate to Traffic Management > SSL > Policies.
2. In the details pane, click Global Bindings.
3. In the Bind/Unbind SSL Policies to Global dialog box, click Insert Policy.
4. In the Policy Name drop-down list, select a policy.
5. Optionally, drag the entry to a new position in the policy bank to automatically update the priority level.
6. Click OK. A message appears in the status bar, stating that the policy has been bound successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 1: Configuring SSL Offloading with End-to-End Encryption

Aug 09, 2017

A simple SSL offloading setup terminates SSL traffic (HTTPS), decrypts the SSL records, and forwards the clear text (HTTP) traffic to the back-end web servers. However, the clear text traffic is vulnerable to being spoofed, read, stolen, or compromised by individuals who succeed in gaining access to the back-end network devices or web servers.

You can, therefore, configure SSL offloading with end-to-end security by re-encrypting the clear text data and using secure SSL sessions to communicate with the back-end Web servers.

Additionally, you can configure the back-end SSL transactions so that the NetScaler appliance uses SSL session multiplexing to reuse existing SSL sessions with the back-end web servers, thus avoiding CPU-intensive key exchange (full handshake) operations. This reduces the overall number of SSL sessions on the server, and therefore accelerates the SSL transaction while maintaining end-to-end security.

To configure SSL Offloading with end-to-end encryption, add SSL based services that represent secure servers with which the NetScaler appliance will carry out end-to-end encryption. Then create an SSL based virtual server, and create and bind a valid certificate-key pair to the virtual server. Bind the SSL services to the virtual server to complete the configuration.

For details on adding SSL based services, see Configuring Services.

For details on adding an SSL virtual server, see Configuring an SSL Based Virtual Server.

For details on creating a certificate-key pair, see Adding a Certificate-Key Pair.

For details on binding a certificate-key pair to a virtual server, see Binding the Certificate Key Pair to the SSL Based Virtual Server.

For details on binding services to a virtual server, see Binding Services to the SSL Based Virtual Server.

To configure an end-to-end encryption deployment, perform the following steps:

- Create SSL services
- Create an SSL virtual server
- Add a certificate-key pair
- Bind the certificate-key pair to the SSL virtual server
- Bind the services to the SSL virtual server

Sample values used in the configuration are listed in the table.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Entity	Name	IP Address	Port
SSL service	service-ssl-1	198.51.100.5	443
SSL service	service-ssl-2	198.51.100.10	443
SSL virtual server	vserver-ssl	203.0.113.5	443
SSL certificate-key pair	certkey-1	-	-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 2: Configuring Transparent SSL Acceleration

Jan 02, 2018

Note: You need to enable L2 mode on the NetScaler appliance for transparent SSL acceleration to work.

Transparent SSL acceleration is useful for running multiple applications on a secure server with the same public IP, and also for SSL acceleration without using an additional public IP.

In a transparent SSL acceleration setup, the NetScaler appliance is transparent to the client, because the IP address at which the appliance receives requests is the same as the Web server's IP address.

The NetScaler offloads SSL traffic processing from the Web server and sends either clear text or encrypted traffic (depending on the configuration) to the web server. All other traffic is transparent to the NetScaler and is bridged to the Web server. Therefore, other applications running on the server are unaffected.

There are three modes of transparent SSL acceleration available on the NetScaler:

- Service-based transparent access, where the service type can be SSL or SSL_TCP.
- Virtual server-based transparent access with a wildcard IP address (*:443).
- SSL VIP-based transparent access with end-to-end encryption.

Note: An SSL_TCP service is used for non-HTTPS services (for example SMTPS and IMAPS).

Service-based Transparent SSL Acceleration

To enable transparent SSL acceleration using the SSL service mode, configure an SSL or an SSL_TCP service with the IP address of the actual back-end Web server. Instead of a virtual server intercepting SSL traffic and passing it on to the service, the traffic is now directly passed on to the service, which decrypts the SSL traffic and sends clear text data to the back-end server.

The service-based mode allows you to configure individual services with a different certificate, or with a different clear text port. Also, you can also select individual services for SSL acceleration.

You can apply service-based transparent SSL acceleration to data that uses different protocols, by setting the clear text port of the SSL service to the port on which the data transfer between the SSL service and the back-end server occurs.

To configure service-based transparent SSL acceleration, first enable both the SSL and the load balancing features. Then create an SSL based service and configure its clear text port. After the service is created, create and bind a certificate-key pair to this service.

For details on configuring the clear text port for an SSL based service, see Configuring Advanced SSL Settings.

For details on creating a certificate-key pair and binding a certificate-key pair to a service, see Adding or Updating a Certificate-Key Pair.

.

Example

Enable SSL offloading and load balancing.

Create an SSL based service, Service-SSL-1 with the IP address 10.102.20.30 using port 443 and configure its clear text port.

Next, create a certificate-key pair, CertKey-1 and bind it to the SSL service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Table 1. Entities in the Service-based Transparent SSL Acceleration

Entity	Name	Value
SSL Service	Service-SSL-1	102.20.30
Certificate - Key Pair	Certkey-1	

Virtual Server-based Acceleration with a Wildcard IP Address (*:443)

You can use an SSL virtual server in the wildcard IP address mode if when you want to enable SSL acceleration for multiple servers that host the secure content of a Web site. In this mode, a single-digital certificate is enough for the entire secure Web site, instead of one certificate per virtual server. This results in significant cost savings on SSL certificates and renewals. The wildcard IP address mode also enables centralized certificate management.

To configure global transparent SSL acceleration on the NetScaler appliance, create a *:443 virtual server, which is a virtual server that accepts any IP address associated with port 443. Then, bind a valid certificate to this virtual server, and also bind all services to which the virtual server is to transfer. Such a virtual server can use the SSL protocol for HTTP-based data or the SSL_TCP protocol for non-HTTP-based data.

To configure virtual server-based acceleration with a wildcard IP address

1. Enable SSL, as described in Enabling SSL Processing.
2. Enable load balancing, as described in Load Balancing.
3. Add an SSL based virtual server (see Configuring an SSL-Based Virtual Server for the basic settings), and set the clearTextPort parameter (described in see Configuring Advanced SSL Settings).
4. Add a certificate-key pair, as described in Adding or Updating a Certificate-Key Pair.
   Note: The wildcard server will automatically learn the servers configured on the NetScaler, so you do not need to configure services for a wildcard virtual server.

Example

After enabling SSL offloading and load balancing, create an SSL based wildcard virtual server with IP address set to * and port number 443, and configure its clear text port (optional).

If you specify the clear text port, decrypted data will be sent to the backend server on that particular port. Otherwise, encrypted data will be sent to port 443.

Next, create an SSL certificate key pair, CertKey-1 and bind it to the SSL virtual server.

Table 2. Entities in the Virtual Server-based Acceleration with a Wildcard IP Address Example

Entity	Name	IP Address	Port
SSL Based Virtual Server	Vserver-SSL-Wildcard	*	443
Certificate - Key Pair	Certkey-1		

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL VIP-based Transparent Access with End-To-End Encryption

You can use an SSL virtual server for transparent access with end-to-end encryption if you have no clear text port specified. In such a configuration, the NetScaler terminates and offloads all SSL processing, initiates a secure SSL session, and sends the encrypted data, instead of clear text data, to the web servers on the port that is configured on the wildcard virtual server.

Note: In this case, the SSL acceleration feature runs at the back-end, using the default configuration, with all 34 ciphers available.
To configure SSL VIP based transparent access with end-to-end encryption, Follow instructions for Configuring a Virtual Server-based Acceleration with a Wildcard IP Address (*:443), but do not configure a clear text port on the virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 3: Configuring SSL Acceleration with HTTP on the Front End and SSL on the Back End

May 26, 2015

In certain deployments, you might be concerned about network vulnerabilities between the NetScaler appliance and the backend servers, or you might need complete end-to-end security and interaction with certain devices that can communicate only in clear text (for example, caching devices).

In such cases, you can set up an HTTP virtual server that receives data from clients that connect to it at the front end and hands the data off to a secure service, which securely transfers the data to the web server.

To implement this type of configuration, you configure an HTTP virtual server on the NetScaler and bind SSL based services to the virtual server. The NetScaler receives HTTP requests from the client on the configured HTTP virtual server, encrypts the data, and sends the encrypted data to the web servers in a secure SSL session.

To configure SSL acceleration with HTTP on the front-end and SSL on the back-end, first enable the load balancing and SSL features on the NetScaler. Then, add SSL based services that represent secure servers to which the NetScaler appliance will send encrypted data. Finally, add an HTTP based virtual server and bind the SSL services to this virtual server.

Example

Enable load balancing and SSL acceleration on the NetScaler.

After enabling load balancing and SSL acceleration, create two SSL based services, Service-SSL-1 and Service-SSL-2, with IP addresses 10.102.20.30 and 10.102.20.31, and both using port 443.

Then create an HTTP based virtual server, Vserver-HTTP-1, with an IP address of 10.102.10.20.

Bind the SSL services to the virtual server to complete the configuration.

Table 1. Entities in the SSL Acceleration with HTTP on the Front End and SSL on the Back End Example

Entity	Name	Value
SSL Service	Service-SSL-1	10.102.20.30
	Service-SSL-2	10.102.20.31
HTTP Based Virtual Server	Vserver-HTTP-1	10.102.10.20

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 4: SSL Offloading with Other TCP Protocols

Jan 01, 2018

In addition to the secure HTTP (HTTPS) protocol, NetScaler appliances support SSL acceleration for other TCP-based secure protocols. However, only simple requests and response-based TCP application protocols are supported. Applications such as FTPS, that insert the server's IP address and port information in their payloads, are not currently supported.

Note: The STARTTLS feature for SMTP is currently not supported.
The NetScaler supports SSL acceleration for Other TCP protocols with and without end-to-end encryption.

To configure SSL offloading with Other TCP protocols, create a virtual server of type SSL_TCP, bind a certificate-key pair and TCP based services to the virtual server, and configure SSL actions and policies based on the type of traffic expected and the acceleration to be provided.

Follow the instructions in Configuring SSL Offloading, but create an SSL_TCP virtual server instead of an SSL virtual server, and configure TCP services instead of HTTP services.

SSL_TCP Based Offloading with End-to-End Encryption

To configure SSL_TCP-based offloading with end-to-end encryption, both the virtual server that intercepts secure traffic and the services that it forwards the traffic to must be of type SSL_TCP.

Configure SSL_TCP-based offloading as described in Use Case 1: Configuring SSL Offloading with End-to-End Encryption, but create an SSL_TCP virtual server instead of an SSL virtual server.

Backend Encryption for TCP Based Data

Some deployments might require the NetScaler appliance to encrypt TCP data received as clear text and send the data securely to the back end servers.

To provide SSL acceleration with back-end encryption for clear text TCP traffic arriving from the client, create a TCP based virtual server and bind it to SSL_TCP based services.

To configure end-to-end encryption for TCP-based data, follow the procedure described in Use Case 3: Configuring SSL Acceleration with HTTP on the Front End and SSL on the Back End, but create a TCP virtual server instead of an HTTP virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 5: Configuring SSL Bridging

May 26, 2015

An SSL bridge configured on the NetScaler appliance enables the appliance to bridge all secure traffic between the SSL client and the SSL server. The appliance does not offload or accelerate the bridged traffic, nor does it perform encryption or decryption. Only load balancing is done by the appliance. The SSL server must handle all SSL-related processing. Features such as content switching, SureConnect, and cache redirection do not work, because the traffic passing through the appliance is encrypted.

Because the appliance does not carry out any SSL processing in an SSL bridging setup, there is no need for SSL certificates.

Citrix recommends that you use this configuration only if an acceleration unit (for example, a PCI-based SSL accelerator card) is installed in the web server to handle the SSL processing overhead.

Before you configure SSL bridging, first enable SSL and load balancing on the appliance. Then, create SSL_Bridge services and bind them to an SSL_Bridge virtual server. Configure the load balancing feature to maintain server persistency for secure requests.

Example

After enabling SSL and load balancing, create two servers, s1 and s2. Create two SSL_Bridge services, sc1 and src2. Create an SSL_Bridge virtual server and bind the SSL_Bridge services to the virtual server to complete the configuration. At the command line, type:

```
enable ns feature SSL LB
add server s1 10.102.1.101
add server s2 10.102.1.102
add service src1 s1 SSL_BRIDGE 443
add service src2 s2 SSL_BRIDGE 443
add lb vserver ssl_bridge_vip SSL_BRIDGE 10.102.1.200 443
bind lb vserver ssl_bridge_vip src1
bind lb vserver ssl_bridge_vip src2
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 6: Configuring SSL Monitoring when Client Authentication is Enabled on the Backend Service

May 26, 2015

Consider a scenario in which you need to load balance servers that require SSL client certificates to validate clients. For this deployment, you need to create an SSL service on the NetScaler appliance, add an HTTPS monitor, add a certificate-key pair, bind this certificate-key pair to the SSL service, and then bind the https monitor to this service. You can use this https monitor to perform health checks on the backend services.

To configure SSL monitoring with client certificate

1. Open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on the appliance by using the administrator credentials.
3. Add an SSL service. At the command prompt, type:
   add service <name> <serverName> <serviceType> <port>
4. Add an https monitor. At the command prompt, type:
   add lb monitor <name> <type>
5. Add the certificate-key pair that is going to be used as the client cert for that SSL service. At the command prompt, type:
   add ssl certKey <certkeyName> -cert <string> -key <string>
6. Bind this certkey to the SSL service. At the command prompt, type:
   bind ssl service <serviceName> -certkeyName <string>
7. Bind the https monitor to the SSL service. At the command prompt, type:
   bind lb monitor <monitorName> <serviceName>

Now, when the appliance tries to probe the backend service on which client authentication is enabled, the backend service will request a certificate as part of the SSL handshake. When the appliance returns the certificate-key bound in step 6 above, the monitor probe will succeed.

Example

```
add service svc_k 10.102.145.30 SSL 443
add lb monitor sslmon HTTP -respCode 200 -httpRequest "GET /testsite/file5.html" -secure YES
add ssl certKey ctest -cert client_rsa_2048.pem -key client_rsa_2048.ky
bind ssl service svc_k -certkeyName ctest
bind lb monitor sslmon svc_k
> show service svc_k
    svc_k (10.102.145.30:443) - SSL
    State: UP
    Last state change was at Tue Jan 10 13:12:24 2012
    Time since last state change: 0 days, 00:09:37.890
    Server Name: 10.102.145.30
    Server ID : 0 Monitor Threshold : 0
    Max Conn: 0 Max Req: 0 Max Bandwidth: 0 kbits
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
    TCP Buffering(TCPB): NO
    HTTP Compression(CMP): NO
    Idle timeout: Client: 180 sec Server: 360 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: OFF
    Down state flush: ENABLED
    Appflow logging: ENABLED

1) Monitor Name: sslmon
    State: UP Weight: 1
    Probes: 1318 Failed [Total: 738 Current: 0]
    Last response: Success - HTTP response code 200 received.
    Response Time: 0.799 millisec
  Done
>
> show ssl service svc_k
    Advanced SSL configuration for Back-end SSL Service svc_k:
    DH: DISABLED
    Ephemeral RSA: DISABLED
    Session Reuse: ENABLED Timeout: 300 seconds
    Cipher Redirect: DISABLED
    SSLv2 Redirect: DISABLED
    Server Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: DISABLED
    SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
1) CertKey Name: ctest Client Certificate

1) Cipher Name: ALL
  Description: Predefined Cipher Alias
Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use Case 7: Configuring a Secure Content Switching Server

May 26, 2015

An SSL-based content switching virtual server first decrypts the secure data and then redirects the data to appropriately configured servers as determined by the type of content and the configured content switching policies. The packets sent to the server have a mapped IP address as the source IP address.

The following example shows the steps to configure two address-based virtual servers to perform load balancing on the HTTP services. One virtual server, Vserver-LB-HTML, load balances the dynamic content (cgi, asp), and the other, Vserver-LB-Image, load balances the static content (gif, jpeg). The load-balancing method used is the default, LEASTCONNECTION. A content switching SSL virtual server, Vserver-CS-SSL, is then configured to perform SSL acceleration and switching of HTTPS requests on the basis of configured content switching policies.

Example

```
> enable ns feature lb cs ssl
> add lb vserver Vserver-LB-HTML http 10.1.1.2 80
> add lb vserver Vserver-LB-Image http 10.1.1.3 80
> add service s1 10.1.1.4 http 80
> add service s2 10.1.1.5 http 80
> add service s3 10.1.1.6 http 80
> add service s4 10.1.1.7 http 80
> bind lb vserver Vserver-LB-HTML s1
> bind lb vserver Vserver-LB-HTML s2
> bind lb vserver Vserver-LB-Image s3
> bind lb vserver Vserver-LB-Image s4
> add cs vserver Vserver-CS-SSL ssl 10.1.1.1 443
> add cs policy pol1 -url "*.cgi"
> add cs policy pol2 -url "*.asp"
> add cs policy pol3 -url "*.gif"
> add cs policy pol4 -url "*.jpeg"
> bind cs vserver Vserver-CS-SSL -policyName pol1 Vserver-LB-HTML
> bind cs vserver Vserver-CS-SSL -policyName pol2 Vserver-LB-HTML
> bind cs vserver Vserver-CS-SSL -policyName pol3 Vserver-LB-Image
> bind cs vserver Vserver-CS-SSL -policyName pol4 Vserver-LB-Image
> add certkey mykey -cert /nsconfig/ssl/ns-root.cert -key /nsconfig/ssl/ns-root.key
> bind certkey Vserver-CS-SSL mykey
>
> show cs vserver Vserver-CS-SSL
    Vserver-CS-SSL (10.1.1.1:443) - SSL Type: CONTENT
    State: UP
    Last state change was at Tue Jul 13 02:11:37 2010
    Time since last state change: 0 days, 00:02:12.440
    Client Idle Timeout: 180 sec
    Down state flush: ENABLED
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Disable Primary Vserver On Down : DISABLED
State Update: DISABLED
Default:       Content Precedence: RULE
Vserver IP and Port insertion: OFF
Case Sensitivity: ON
Push: DISABLED  Push VServer:
Push Label Rule: none
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Ciphers Supported by the NetScaler Appliance

May 06, 2016

Your NetScaler appliance ships with a predefined set of cipher groups. To use ciphers that are not part of the DEFAULT cipher group, you have to explicitly bind them to an SSL virtual server. You can also create a user-defined cipher group to bind to the SSL virtual server. For more information about creating a user-defined cipher group, see Configuring User-Defined Cipher Groups on the NetScaler Appliance.

Table 1 lists the NetScaler appliances that support ciphers for the TLS 1.1 and 1.2 protocols.

Table 2 lists the ciphers that are part of the DEFAULT cipher group and are therefore bound by default to an SSL virtual server.

Table 3 lists the other ciphers currently supported by NetScaler appliances.

Table 1. Support for TLS1.1/1.2 on NetScaler Appliances

	Virtual Server	Front End Service	Internal Service	Backend Service
MPX appliances/SDX instances with SSL card/ MPX-FIPS appliances running firmware version 2.2	yes	yes	yes	yes
MPX-FIPS appliances running firmware version other than version 2.2	no	no	no	no
VPX appliances/ SDX instances without SSL card	yes	yes	yes	yes

Table 2. Ciphers That the NetScaler Appliance Supports by Default on the Front End

Cipher Suite	Protocol	Key Exchange Algorithm	Authentication Algorithm	Encryption Algorithm (Key Size)	Message Authentication Code (MAC) Algorithm	Hex Code	Wireshark Ciphersuite Name
TLS1-AES-256-CBC-SHA	SSLv3 TLSv1 TLSv1.1 TLSv1.2	RSA	RSA	AES(256)	SHA1	0x0035	TLS_RSA_WITH_AES_256_CBC_SHA
TLS1-AES-128-CBC-SHA	SSLv3 TLSv1 TLSv1.1 TLSv1.2	RSA	RSA	AES(128)	SHA1	0x002f	TLS_RSA_WITH_AES_128_CBC_SHA
TLS1.2-AES-256-SHA256	TLSv1.2	RSA	RSA	AES(256)	SHA-256	0x003d	TLS_RSA_WITH_AES_256_CBC_SHA256
TLS1.2-AES-128-SHA256	TLSv1.2	RSA	RSA	AES(128)	SHA-256	0x003c	TLS_RSA_WITH_AES_128_CBC_SHA256

© 1999-2017 Citrix Systems, Inc. All rights reserved.

TLS1.2-AES256-GCM-SHA384	TLSv1.2	RSA	RSA	AES-GCM(256)	SHA-384	0x009d	TLS_RSA_WITH_AES_256_GCM_SHA384
TLS1.2-AES128-GCM-SHA256	TLSv1.2	RSA	RSA	AES-GCM(128)	SHA-256	0x009c	TLS_RSA_WITH_AES_128_GCM_SHA256
TLS1-ECDHE-RSA-AES256-SHA	SSLv3 TLSv1 TLSv1.1 TLSv1.2	ECC-DHE	RSA	AES(256)	SHA1	0xc014	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS1-ECDHE-RSA-AES128-SHA	SSLv3 TLSv1 TLSv1.1 TLSv1.2	ECC-DHE	RSA	AES(128)	SHA1	0xc013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS1.2-ECDHE-RSA-AES-256-SHA384	TLSv1.2	ECC-DHE	RSA	AES(256)	SHA-384	0xc028	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS1.2-ECDHE-RSA-AES-128-SHA256	TLSv1.2	ECC-DHE	RSA	AES(128)	SHA-256	0xc027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS1.2-ECDHE-RSA-AES256-GCM-SHA384	TLSv1.2	ECC-DHE	RSA	AES-GCM(256)	SHA-384	0xc030	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS1.2-ECDHE-RSA-AES128-GCM-SHA256	TLSv1.2	ECC-DHE	RSA	AES-GCM(128)	SHA-256	0xc02f	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS1.2-DHE-RSA-AES-256-SHA256	TLSv1.2	DH	RSA	AES(256)	SHA-256	0x006b	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS1.2-	TLSv1.2	DH	RSA	AES(128)	SHA-256	0x0067	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

© 1999-2017 Citrix Systems, Inc. All rights reserved.

DHE-RSA-AES-128-SHA256							
TLS1.2-DHE-RSA-AES256-GCM-SHA384	TLSv1.2	DH	RSA	AES-GCM(256)	SHA-384	0x009f	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS1.2-DHE-RSA-AES128-GCM-SHA256	TLSv1.2	DH	RSA	AES-GCM(128)	SHA-256	0x009e	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS1-DHE-RSA-AES-256-CBC-SHA	SSLv3 TLSv1 TLSv1.1 TLSv1.2	DH	RSA	AES(256)	SHA1	0x0039	TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS1-DHE-RSA-AES-128-CBC-SHA	SSLv3 TLSv1 TLSv1.1 TLSv1.2	DH	RSA	AES(128)	SHA1	0x0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS1-DHE-DSS-AES-256-CBC-SHA	SSLv3 TLSv1	DH	DSS	AES(256)	SHA1	0x0038	TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS1-DHE-DSS-AES-128-CBC-SHA	SSLv3 TLSv1	DH	DSS	AES(128)	SHA1	0x0032	TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS1-ECDHE-RSA-DES-CBC3-SHA	SSLv3 TLSv1 TLSv1.1 TLSv1.2	ECC-DHE	RSA	3DES(168)	SHA1	0xc012	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
SSL3-EDH-RSA-DES-CBC3-	SSLv3 TLSv1 TLSv1.1	DH	RSA	3DES(168)	SHA1	0x0016	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SHA	TLSv1.2						
SSL3-EDH-DSS-DES-CBC3-SHA	SSLv3 TLSv1	DH	DSS	3DES(168)	SHA1	0x0013	TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
TLS1-ECDHE-RSA-RC4-SHA	SSLv3 TLSv1 TLSv1.1 TSv1.2	ECC-DHE	RSA	RC4(128)	SHA1	0xc011	TLS_ECDHE_RSA_WITH_RC4_128_SHA
TLS1-DHE-DSS-RC4-SHA	SSLv3 TLSv1	DH	DSS	RC4(128)	SHA1	0x0066	TLS_DHE_DSS_WITH_RC4_128_SHA
SSL3-DES-CBC3-SHA	SSLv3 TLSv1 TLSv1.1 TLSv1.2	RSA	RSA	3DES(168)	SHA1	0x000a	TLS_RSA_WITH_3DES_EDE_CBC_SHA
SSL3-RC4-SHA	SSLv3 TLSv1 TLSv1.1 TLSv1.2	RSA	RSA	RC4(128)	SHA1	0x0005	TLS_RSA_WITH_RC4_128_SHA
SSL3-RC4-MD5	SSLv3 TLSv1 TLSv1.1 TLSv1.2	RSA	RSA	RC4(128)	MD5	0x0004	TLS_RSA_WITH_RC4_128_MD5

Table 3. Additional Ciphers Supported by the NetScaler Appliance

Cipher Suite	Protocol	Key Exchange Algorithm	Authentication Algorithm	Encryption Algorithm (Key Size)	Message Authentication Code (MAC) Algorithm	Hex Code	Wireshark Ciphersuite Name
SSL3-DES-CBC-SHA	SSLv3 TLSv1 TLv1.1	RSA	RSA	DES(56)	SHA1	0x0009	TLS_RSA_WITH_DES_CBC_SHA
TLS1-	TLSv1	RSA(1024)	RSA	RC4(56)	SHA1 Export	0x0064	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA

© 1999-2017 Citrix Systems, Inc. All rights reserved.

EXP1024-RC4-SHA							
SSL3-EXP-RC4-MD5	SSLv3 TLSv1	RSA(512)	RSA	RC4(40)	MD5 Export	0x0003	TLS_RSA_EXPORT_WITH_RC4_40_MD5
SSL3-EXP-DES-CBC-SHA	SSLv3 TLSv1	RSA(512)	RSA	DES(40)	SHA1 Export	0x0008	TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL3-EXP-RC2-CBC-MD5	SSLv3 TLSv1	RSA(512)	RSA	RC2(40)	MD5 Export	0x0006	TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
SSL2-RC4-MD5	SSLv2	RSA	RSA	RC4(128)	MD5	0x0080	SSL2_RC4_128_WITH_MD5
SSL2-DES-CBC3-MD5	SSLv2	RSA	RSA	3DES(168)	MD5	0x00c0	SSL2_DES_192_EDE3_CBC_WITH_MD5
SSL2-RC2-CBC-MD5	SSLv2	RSA	RSA	RC2(128)	MD5	0x0080	SSL2_RC4_128_EXPORT40_WITH_MD5
SSL2-DES-CBC-MD5	SSLv2	RSA	RSA	DES(56)	MD5	0x0040	SSL2_DES_64_CBC_WITH_MD5
SSL2-RC4-64-MD5	SSLv2	RSA	RSA	RC4(64)	MD5	0x0080	SSL2_RC4_64_WITH_MD5
SSL2-EXP-RC4-MD5	SSLv2	RSA(512)	RSA	RC4(40)	MD5 Export	0x0080	SSL2_RC4_128_EXPORT40_WITH_MD5
SSL3-EDH-DSS-DES-CBC-SHA	SSLv3 TLSv1	DH	DSS	DES(56)	SHA1	0x0012	TLS_DHE_DSS_WITH_DES_CBC_SHA
TLS1-EXP1024-DHE-DSS-DES-CBC- SHA	TLSv1	DH(1024)	DSS	DES(56)	SHA1 Export	0x0063	TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA

© 1999-2017 Citrix Systems, Inc. All rights reserved.

TLS1-EXP1024-DHE-DSS-RC4- SHA	TLSv1	DH(1024)	DSS	RC4(56)	SHA1 Export	0x0065	TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA
SSL3-EXP-EDH-DSS-DES-CBC-SHA	SSLv3 TLSv1	DH(512)	DSS	DES(40)	SHA1 Export	0x0011	TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
SSL3-EDH-RSA-DES-CBC-SHA	SSLv3 TLSv1 TLSv1.1	DH	RSA	DES(56)	SHA1	0x0015	TLS_DHE_RSA_WITH_DES_CBC_SHA
SSL3-EXP-EDH-RSA-DES-CBC-SHA	SSLv3 TLSv1	DH(512)	RSA	DES(40)	DES(40)	0x0014	TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS1-EXP1024-RC4-MD5	TLSv1	RSA(1024)	RSA	RC4(56)	MD5 Export	0x0060	TLS_RSA_EXPORT1024_WITH_RC4_56_MD5
TLS1-EXP1024-RC2-CBC-MD5	TLSv1	RSA(1024)	RSA	RC2(56)	MD5 Export	0x0061	TLS_RSA_EXPORT1024_WITH_RC2_CBC_56_MD5
SSL2-EXP-RC2-CBC-MD5	SSLv2	RSA(512)	RSA	RC2(40)	MD5 Export	0x0080	SSL2_RC2_CBC_128_CBC_WITH_MD5
SSL3-ADH-RC4-MD5	SSLv3 TLSv1 TLSv1.1	DH	None	RC4(128)	MD5	0x0018	TLS_DH_anon_WITH_RC4_128_MD5
SSL3-ADH-DES-CBC-SHA	SSLv3 TLSv1 TLSv1.1	DH	None	DES(56)	SHA1	0x001b	TLS_DH_anon_WITH_3DES_EDE_CBC_SHA
SSL3-ADH-DES-CBC3-SHA	SSLv3 TLSv1 TLSv1.1	DH	None	3DES(168)	SHA1	0x001a	TLS_DH_anon_WITH_DES_CBC_SHA

© 1999-2017 Citrix Systems, Inc. All rights reserved.

TLS1-ADH-AES-128-CBC-SHA	SSLv3 TLSv1 TLSv1.1	DH	None	AES(128)	SHA1	0x0034	TLS_DH_anon_WITH_AES_128_CBC_SHA
TLS1-ADH-AES-256-CBC-SHA	SSLv3 TLSv1 TLSv1.1	DH	None	AES(256)	SHA1	0x003a	TLS_DH_anon_WITH_AES_256_CBC_SHA
SSL3-EXP-ADH-RC4-MD5	SSLv3 TLSv1	DH(512)	None	RC4(40)	MD5 Export	0x0017	TLS_DH_anon_EXPORT_WITH_RC4_40_MD5
SSL3-EXP-ADH-DES-CBC-SHA	SSLv3 TLSv1	DH(512)	None	DES(40)	SHA1 Export	0x0019	TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA

Note

1. The following curves are supported for ECDHE key exchange algorithms:
   - ECDHE 521 curve
   - ECDHE 384 curve
   - ECDHE 256 curve
   - ECDHE 224 curve

   For more information about the ECDHE ciphers supported on a NetScaler appliance, see Configuring ECDHE Ciphers.
2. The NetScaler VPX appliance supports only the SHA256 + RSA (Hex code 0x0401) signature hash algorithm in the certificate request message. If you use a different signature algorithm to sign the certificate verify message, the TLSv1.2 handshake fails during client authentication with the error message "Unsupported Certificate."
3. AES-GCM/SHA2 ciphers are supported only on the front end SSL entities on an MPX, VPX, or SDX appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cipher/Protocol Support Matrix on the NetScaler Appliance

Apr 27, 2017

From release 10.5 build 56.22, NetScaler MPX appliances support full hardware optimization for all ciphers. In earlier releases, part of ECDHE/DHE cipher optimization was done in software.

For the updated cipher/protocol support matrix, see http://docs.citrix.com/en-us/netscaler/12/ssl/cipher_protocl_support_matrix.html.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Server Certificate Support Matrix on the NetScaler Appliance

Apr 27, 2017

For information about the server certificate support matrix, see http://docs.citrix.com/en-us/netscaler/12/ssl/ssl-server-cert-support-matrix.html.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Support for MPX 5900 and MPX 8900 Platforms

Jan 18, 2018

> **Note**
>
> For supported software releases, see Cipher/Protocol Support Matrix on the NetScaler Appliance.

The MPX 5900/8900 appliances ship with Intel Coleto chips. Use the 'show hardware' command to identify whether your appliance has Coleto (COL) chips.

Code COPY

```
> sh hardware

    Platform: NSMPX-8900 8*CPU+4*F1X+6*E1K+1*E1K+1*COL 8955 30010

    Manufactured on: 10/18/2016

    CPU: 2100MHZ

    Host Id: 0

    Serial no: CRAC5CR8UA

    Encoded serial no: CRAC5CR8UA

  Done

>
```

Limitations

- DH 512 key exchange is not supported.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- DH and ECDHE ciphers are not supported with a 512-bit server certificate.
- SSLv3 and SSLv2 protocols are not supported.
- RC4 cipher renegotiation is not supported.
- DTLS is not supported.
- DNSSEC offload is not supported.
- GnuTLS is not supported.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the MPX 9700/10500/12500/15500 FIPS Appliances

Jan 01, 2018

The Federal Information Processing Standard (FIPS), issued by the US National Institute of Standards and Technologies, specifies the security requirements for a cryptographic module used in a security system. The NetScaler FIPS appliance complies with the second version of this standard, FIPS-140-2.

Note: Henceforth, all references to FIPS imply FIPS-140-2.
The FIPS appliance is equipped with a tamper-proof (tamper-evident) cryptographic module—and a Cavium CN1620-NFBE3-2.0-G on the MPX 9700/10500/12500/15500 FIPS appliances—designed to comply with the FIPS 140-2 Level-2 specifications. The Critical Security Parameters (CSPs), primarily the server's private-key, are securely stored and generated inside the cryptographic module, also referred to as the Hardware Security Module (HSM). The CSPs are never accessed outside the boundaries of the HSM. Only the superuser (nsroot) can perform operations on the keys stored inside the HSM.

The following table summarizes the differences between standard NetScaler and NetScaler FIPS appliances.

Setting	NetScaler appliance	NetScaler FIPS appliance
Key storage	On the hard disk	On the FIPS card
Cipher support	All ciphers	FIPS approved ciphers
Accessing keys	From the hard disk	Not accessible

Configuring a FIPS appliance involves configuring the HSM immediately after completing the generic configuration process. You then create or import a FIPS key. After creating a FIPS key, you should export it for backup. You might also need to export a FIPS key so that you can import it to another appliance. For example, configuring FIPS appliances in a high availability (HA) setup requires transferring the FIPS key from the primary node to the secondary node immediately after completing the standard HA setup.

You can upgrade the firmware version on the FIPS card from version 4.6.0 to 4.6.1, and you can reset an HSM that has been locked to prevent unauthorized logon. Only FIPS approved ciphers are supported on a NetScaler FIPS appliance.

This section includes the following details:

- Configuring the HSM
- Creating and Transferring FIPS Keys
- Configuring FIPS Appliances in a High Availability Setup
- Resetting a Locked HSM
- FIPS Approved Algorithms and Ciphers

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the HSM

Aug 30, 2016

Before you can configure the HSM of your NetScaler FIPS appliance, you must complete the initial hardware configuration. For more information, see Initial Configuration.

Configuring the HSM of your NetScaler FIPS appliance erases all existing data on the HSM. To configure the HSM, you must be logged on to the appliance as the superuser (nsroot account). The HSM is preconfigured with default values for the Security Officer (SO) password and User password, which you use to configure the HSM or reset a locked HSM. The maximum length allowed for the password is 14 alphanumeric characters. Symbols are not allowed.
Important: Do not perform the set ssl fips command without first resetting the FIPS card and restarting the MPX FIPS appliance.
Although the FIPS appliance can be used with the default password values, you should modify them before using it. The HSM can be configured only when you log on to the appliance as the superuser and specify the SO and User passwords.

Important: Due to security constraints, the appliance does not provide a means for retrieving the SO password. Store a copy of the password safely. Should you need to reinitialize the HSM, you will need to specify this password as the old SO password.
Before initializing the HSM, you can upgrade to the latest build of the software. To upgrade to the latest build, see Upgrading or Downgrading the System Software.

After upgrading, verify that the /nsconfig/fips directory has been successfully created on the appliance.

To configure the HSM on an MPX 9700/10500/12500/15500 FIPS appliances by using the command line interface

After logging on to the appliance as the superuser and completing the initial configuration, at the command prompt, type the following commands to configure the HSM and verify the configuration:

1. show ssl fips
2. reset ssl fips
3. reboot
4. set ssl fips -initHSM Level-2 <newSOpassword> <oldSOpassword> <userPassword> [-hsmLabel <string>]
5. save ns config
6. reboot
7. show ssl fips

Example

```
show fips
FIPS Card is not configured
Done
reset fips
reboot
Are you sure you want to restart NetScaler (Y/N)? [N]:y
set ssl fips -initHSM Level-2 sopin12345 so12345 user123 -hsmLabel cavium
This command will erase all data on the FIPS card. You must save the configuration
(saveconfig) after executing this command.

Do you want to continue?(Y/N)y
Done
save ns config
reboot
Are you sure you want to restart NetScaler (Y/N)? [N]:y
show fips
    FIPS HSM Info:
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
HSM Label            : NetScaler FIPS
Initialization       : FIPS-140-2 Level-2
HSM Serial Number    : 2.1G1008-IC000021
HSM State            : 2
HSM Model            : NITROX XL CN1620-NFBE
Firmware Version     : 1.1
Firmware Release Date : Jun04,2010

Max FIPS Key Memory   : 3996
Free FIPS Key Memory  : 3994
Total SRAM Memory     : 467348
Free SRAM Memory      : 62564
Total Crypto Cores    : 3
Enabled Crypto Cores  : 1
Done
Note: If you upgrade the firmware to version 2.2, the firmware release date is replaced with the firmware build.

> show fips
FIPS HSM Info:
HSM Label            : NetScaler FIPS
Initialization       : FIPS-140-2 Level-2
HSM Serial Number    : 3.0G1235-ICM000264
HSM State            : 2
HSM Model            : NITROX XL CN1620-NFBE
Hardware Version     : 2.0-G
Firmware Version     : 2.2
Firmware Build       : NFBE-FW-2.2-130009

Max FIPS Key Memory : 3996
Free FIPS Key Memory : 3958
Total SRAM Memory   : 467348
Free SRAM Memory    : 50524
Total Crypto Cores  : 3
Enabled Crypto Cores : 3
Done
```

To configure the HSM on an MPX 9700/10500/12500/15500 FIPS appliances by using the configuration utility

1. Navigate to Traffic Management > SSL > FIPS.
2. In the details pane, on the FIPS Infotab, click Reset FIPS.
3. In the navigation pane, click System.
4. In the details pane, click Reboot.
5. In the details pane, on the FIPS Info tab, click Initialize HSM.
6. In the Initialize HSM dialog box, specify values for the following parameters:
   - Security Officer (SO) Password*—new SO password
   - Old SO Password*—old SO password
   - User Password*—user password
   - Level—initHSM (Currently set to Level2 and cannot be changed)
   - HSM Label—hsmLabel

   *A required parameter

7. Click OK.
8. In the details pane, click Save.
9. In the navigation pane, click System.
10. In the details pane, click Reboot.
11. Under FIPS HSM Info, verify that the information displayed for the FIPS HSM that you just configured is correct.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating and Transferring FIPS Keys

Nov 04, 2016

After configuring the HSM of your FIPS appliance, you are ready to create a FIPS key. The FIPS key is created in the appliance's HSM. You can then export the FIPS key to the appliance's CompactFlash card as a secured backup. Exporting the key also enables you to transfer it by copying it to the /flash of another appliance and then importing it into the HSM of that appliance. You must enable SIM between two standalone nodes before you export and transfer the keys. In an HA setup, if one of the nodes is replaced with a new appliance, you must enable SIM between this new appliance and the existing appliance of the HA setup before you export or import FIPS keys.

Instead of creating a FIPS key, you can import an existing FIPS key or import an external key as a FIPS key. If you are adding a certificate-key pair of 2048 bits on the MPX 9700/10500/12500/15500 FIPS appliances, make sure that you have the correct certificate and key pair.

Note: If you are planning an HA setup, make sure that the FIPS appliances are configured in an HA setup before creating a FIPS key.

Creating a FIPS Key

Before creating a FIPS key, make sure that the HSM is configured.

To create a FIPS key by using the configuration utility

1. Navigate to Traffic Management > SSL > FIPS.
2. In the details pane, on the FIPS Keys tab, click Add.
3. In the Create FIPS Key dialog box, specify values for the following parameters:
   - FIPS Key Name*—fipsKeyName
   - Modulus*—modulus
   - Exponent*—exponent

   *A required parameter
4. Click Create, and then click Close.
5. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you just created are correct.

To create a FIPS key by using the command line interface

At the command prompt, type the following commands to create a FIPS key and verify the settings:

- create ssl fipsKey <fipsKeyName> -modulus <positive_integer> [-exponent (3 | F4)]
- show ssl fipsKey [<fipsKeyName>]

Example

```
create fipskey Key-FIPS-1 -modulus 2048 -exponent 3
show ssl fipsKey Key-FIPS-1
FIPS Key Name: Key-FIPS-1 Modulus: 2048   Public Exponent: 3 (Hex: 0x3)
```

Exporting a FIPS Key

Citrix recommends that you create a backup of any key created in the FIPS HSM. If a key in the HSM is deleted, there is no way to create the same key again, and all the certificates associated with it are rendered useless.

In addition to exporting a key as a backup, you might need to export a key for transfer to another appliance.

The following procedure provides instructions on exporting a FIPS key to the /nsconfig/ssl folder on the appliance's

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

CompactFlash and securing the exported key by using a strong asymmetric key encryption method.

To export a FIPS key by using the command line interface

At the command prompt, type:
export ssl fipsKey <fipsKeyName> -key <string>
Example

```
export fipskey Key-FIPS-1 -key Key-FIPS-1.key
```

To export a FIPS key by using the configuration utility

1. Navigate to Traffic Management > SSL > FIPS
2. In the details pane, on the FIPS Keys tab, click Export.
3. In the Export FIPS key to a file dialog box, specify values for the following parameters:
   - FIPS Key Name*—fipsKeyName
   - File Name*—key (To put the file in a location other than the default, you can either specify the complete path or click the Browse button and navigate to a location.)

   *A required parameter

4. Click Export, and then click Close.

Importing an Existing FIPS Key

To use an existing FIPS key with your FIPS appliance, you need to transfer the FIPS key from the hard disk of the appliance into its HSM.

Note: To avoid errors when importing a FIPS key, make sure that the name of the key imported is the same as the original key name when it was created.

To import a FIPS key on the MPX 9700/10500/12500/15500 FIPS appliances by using the command line interface

At the command prompt, type the following commands to import a FIPS key and verify the settings:
- import ssl fipsKey <fipsKeyName> -key <string> -inform SIM -exponent (F4 | 3)
- show ssl fipskey <fipsKeyName>

Example

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4
show ssl fipskey key-FIPS-2
FIPS Key Name: Key-FIPS-2 Modulus: 2048   Public Exponent: F4 (Hex value 0x10001)
```

To import a FIPS key by using the configuration utility

1. Navigate to Traffic Management > SSL > FIPS
2. In the details pane, on the FIPS Keys tab, click Import.
3. In the Import as a FIPS Key dialog box, select FIPS key file and set values for the following parameters:
   - FIPS Key Name*
   - Key File Name*—To put the file in a location other than the default, you can either specify the complete path or click Browse and navigate to a location.
   - Exponent*

   *A required parameter

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Click Import, and then click Close.
5. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you just imported are correct.

Importing External Keys

In addition to transferring FIPS keys that are created within the NetScaler appliance's HSM, you can transfer external private keys (such as those created on a standard NetScaler, Apache, or IIS) to a FIPS NetScaler appliance. External keys are created outside the HSM, by using a tool such as OpenSSL. Before importing an external key into the HSM, copy it to the appliance's flash drive under /nsconfig/ssl.

Importing an external key as a FIPS key on the MPX 9700/10500/12500/15500 FIPS appliances by using the command line interface

On the MPX 9700/10500/12500/15500 FIPS appliances, the -exponent parameter in the import ssl fipskey command is not required while importing an external key. The correct public exponent is detected automatically when the key is imported, and the value of the -exponent parameter is ignored.

The NetScaler FIPS appliance does not support external keys with a public exponent other than 3 or F4.

You do not need a wrap key on the MPX 9700/10500/12500/15500 FIPS appliances.

You cannot import an external, encrypted FIPS key directly to an MPX 9700/10500/12500/15500 FIPS appliance. To import the key you need to first decrypt the key, and then import it. To decrypt the key, at the shell prompt, type:

openssl rsa -in <EncryptedKey.key> > <DecryptedKey.out>

To import an external key as a FIPS key to an MPX 9700/10500/12500/15500 FIPS appliance by using the command line interface

1. Copy the external key to the appliance's flash drive.
2. If the key is in .pfx format, you must first convert it to PEM format. At the command prompt, type:
   - convert ssl pkcs12 <output file> -import -pkcs12File <input .pfx file name> -password <password>
3. At the command prompt, type the following commands to import the external key as a FIPS key and verify the settings:
   - import ssl fipsKey <fipsKeyName> -key <string> -inform PEM
   - show ssl fipskey<fipsKeyName>

Example

```
convert ssl pkcs12 iis.pem -password 123456 -import -pkcs12File iis.pfx
import fipskey Key-FIPS-2 -key iis.pem -inform PEM
show ssl fipskey key-FIPS-2
FIPS Key Name: Key-FIPS-2 Modulus: 0   Public Exponent: F4 (Hex value 0x10001)
```

Note: The modulus is incorrectly displayed as zero in the above example. The discrepancy does not affect SSL functionality.

To import an external key as a FIPS key to an MPX 9700/10500/12500/15500 FIPS appliance by using the configuration utility

1. If the key is in .pfx format, you must first convert it to PEM format.
   1. Navigate to Traffic Management > SSL.
   2. In the details pane, under Tools, click Import PKCS#12.
   3. In the Import PKCS12 File dialog box, set the following parameters:
      - Output File Name*

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- PKCS12 File Name*—Specify the .pfx file name.
- Import Password*
- Encoding Format

*A required parameter

2. Navigate to Traffic Management > SSL > FIPS
3. In the details pane, on the FIPS Keys tab, click Import.
4. In the Import as a FIPS Key dialog box, select PEM file, and set values for the following parameters:
   - FIPS Key Name*
   - Key File Name*—To put the file in a location other than the default, you can either specify the complete path or click Browse and navigate to a location.

   *A required parameter

5. Click Import, and then click Close.
6. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you just imported are correct.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring FIPS Appliances in a High Availability Setup

Jan 01, 2018

You can configure two appliances in a high availability (HA) pair as FIPS appliances. For information about configuring an HA setup, see High Availability.

Note: Citrix recommends that you use the configuration utility (GUI) for this procedure. If you use the command line (CLI), make sure that you carefully follow the steps as listed in the procedure. Changing the order of steps or specifying an incorrect input file might cause inconsistency that requires that the appliance be restarted. In addition, if you use the CLI, the create ssl fipskey command is not propagated to the secondary node. When you execute the command with the same input values for modulus size and exponent on two different FIPS appliances, the keys generated are not identical. You have to create the FIPS key on one of the nodes and then transfer it to the other node. But if you use the configuration utility to configure FIPS appliances in an HA setup, the FIPS key that you create is automatically transferred to the secondary node. The process of managing and transferring the FIPS keys is known as secure information management (SIM).
Important: On the MPX 9700/10500/12500/15500 FIPS appliances, the HA setup should be completed within six minutes. If the process takes longer than six minutes, the internal timer of the FIPS card expires and the following error message appears:
ERROR: Operation timed out or repeated, please wait for 10 mins and redo the SIM/HA configuration steps.

If this message appears, restart the appliance or wait for 10 minutes, and then repeat the HA setup procedure.

In the following procedure, appliance A is the primary node and appliance B is the secondary node.

To configure FIPS appliances in a high availability setup by using the command line interface

1. **On appliance A**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on to the appliance, using the administrator credentials.
3. Initialize appliance A as the source appliance. At the command prompt, type:
   init ssl fipsSIMsource <certFile>
4. Copy this <certFile> file to appliance B, in the /nconfig/ssl folder.
5. **On appliance B**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
6. Log on to the appliance, using the administrator credentials.
7. Initialize appliance B as the target appliance. At the command prompt, type:
   init ssl fipsSIMtarget <certFile> <keyVector> <targetSecret>
8. Copy this <targetSecret> file to appliance A.
9. **On appliance A**, enable appliance A as the source appliance. At the command prompt, type:
   enable ssl fipsSIMSource <targetSecret> <sourceSecret>
10. Copy this <sourceSecret> file to appliance B.
11. **On appliance B**, enable appliance B as the target appliance. At the command prompt, type:
    enable ssl fipsSIMtarget <keyVector> <sourceSecret>
12. **On appliance A**, create a FIPS key, as described in the "Creating a FIPS Key" section in Creating and Transferring FIPS Keys.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

13. Export the FIPS key to the appliance's hard disk, as described in the "Exporting a FIPS Key" section in Creating and Transferring FIPS Keys.
14. Copy the FIPS key to the hard disk of the secondary appliance by using a secure file transfer utility, such as SCP.
15. **On appliance B**, import the FIPS key from the hard disk into the HSM of the appliance, as described in the "Importing an Existing FIPS Key" section in Creating and Transferring FIPS Keys.

To configure FIPS appliances in a high availability setup by using the configuration utility

1. On the appliance to be configured as the source appliance, navigate to Traffic Management > SSL > FIPS.
2. In the details pane, on the FIPS Info tab, click Enable SIM.
3. In the Enable HA Pair for SIM dialog box, in the Certificate File Name text box, type the file name, with the path to the location at which the FIPS certificate should be stored on the source appliance.
4. In the Key Vector File Name text box, type the file name, with the path to the location at which the FIPS key vector should be stored on the source appliance.
5. In the Target Secret File Name text box, type the location for storing the secret data on the target appliance.
6. In the Source Secret File Name text box, type the location for storing the secret data on the source appliance.
7. Click OK. The FIPS appliances are now configured in HA mode.
8. Create a FIPS key, as described in the "Creating a FIPS Key" section in Creating and Transferring FIPS Keys. The FIPS key is automatically transferred from the primary to the secondary.

Example

In the following example, source.cert is the certificate on the source appliance, stored in the default directory, /nsconfig/ssl. This certificate must be transferred to the same location (/nsconfig/ssl) on the target appliance. The file target.secret is created on the target appliance and copied to the source appliance. The file source.secret is created on the source appliance and copied to the target appliance.

On the source appliance

```
init fipsSIMsource /nsconfig/ssl/source.cert
```

On the target appliance

```
init fipsSIMtarget /nsconfig/ssl/source.cert /nsconfig/ssl/target.key /nsconfig/ssl/target.secret
```

On the source appliance

```
enable fipsSIMsource /nsconfig/ssl/target.secret /nsconfig/ssl/source.secret
```

On the target appliance

```
enable fipsSIMtarget /nsconfig/ssl/target.key /nsconfig/ssl/source.secret
```

On the source appliance

```
create ssl fipskey fips1 -modulus 2048 -exponent f4
export fipskey fips1 -key /nsconfig/ssl/fips1.key
```

Copy this key into the hard disk of the target appliance.

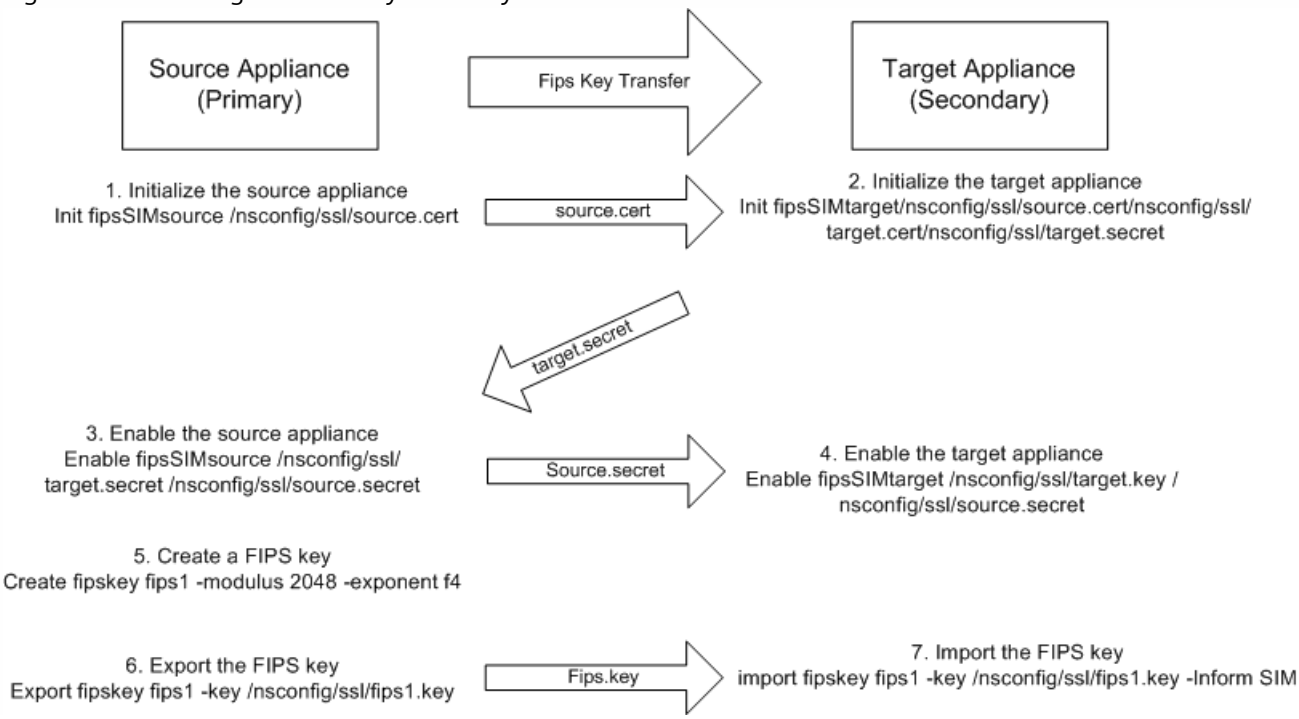
On the target appliance

```
import fipskey fips1 -key /nsconfig/ssl/fips1.key
```

The following diagram summarizes the transfer process.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Figure 1. Transferring the FIPS Key-Summary



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updating the Firmware to Version 2.2 on a FIPS Card

Sep 27, 2016

FIPS firmware version 2.2 supports TLS protocol versions 1.1 and 1.2. From the command line, you can update the firmware version of the FIPS card of a NetScaler MPX 9700/10500/12500/15500 FIPS appliance from version 1.1 to version 2.2.

For successful SIM key propagation from primary to secondary in a high availability (HA) pair, the Cavium firmware version on each appliance should be identical. Perform the firmware update on the secondary appliance first. If executed on the primary appliance first, the long-running update process causes a failover.

Limitations

- Secure renegotiation is supported only on SSL virtual servers and front-end SSL services.
- Creating a certificate signing request by using a key that was created on firmware version 1.1 and updated to firmware version 2.2 fails.
- You cannot create a 1024-bit RSA key on firmware version 2.2. However, if you have imported or created a 1024-bit FIPS key on firmware version 1.1 and you then update to firmware version 2.2, you can use that FIPS key on firmware version 2.2.
- 1024-bit RSA keys are not supported.
- Secure renegotiation using SSLv3 protocol is not supported.
- After you upgrade the firmware, TLSv1.1 and TLSv1.2 are disabled by default on the existing virtual server, internal, front end, and backend services. To use TLS 1.1/1.2, you must explicitly enable these protocols, on the SSL entities, after the upgrade.
- FIPS keys that are created in firmware version 2.2 are not available if you downgrade the firmware to version 1.1.

Prerequisites

Download the following files from the download page on www.citrix.com. The files must be stored in the /var/nsinstall directory on the appliance.

- FW 2.2 File: FW-2.2-130013
- FW 2.2 Signature File: FW-2.2-130013.sign

To update the FIPS firmware to version 2.2 on a standalone appliance

1. Log on to the appliance by using the administrator credentials.
2. At the prompt, type the following command to confirm that the FIPS card is initialized.
   show fips

   ```
   FIPS HSM Info:
     HSM Label  : NetScaler FIPS
     Initialization  : FIPS-140-2 Level-2
     HSM Serial Number : 3.0G1235-ICM000264
     HSM State  : 2
     HSM Model  : NITROX XL CN1620-NFBE

     Hardware Version : 2.0-G
     Firmware Version : 1.1
     Firmware Release Date : Jun04,2010

     Max FIPS Key Memory : 3996
     Free FIPS Key Memory : 3992
     Total SRAM Memory : 467348
     Free SRAM Memory : 62512
     Total Crypto Cores : 3
     Enabled Crypto Cores : 1
     Done
   ```
3. Save the configuration. At the prompt, type:
   save config
4. Perform the update. At the prompt, type:
   update ssl fips -fipsFW <path to the extracted contents>/CN16XX-NFBE-FW-2.2-130009

   and press Y when the following prompt appears:
   This command will update compatible version of the FIPS firmware. You must save the current configuration (saveconfig) before executing this command. You must reboot the system after execution of
   Done
   Note: You only need to specify the firmware file, because the firmware signature file is placed in the same location.
   The update takes up to ten seconds. The update command is blocking, which means that no other actions are executed until the command finishes. The command prompt reappears when execution of the command is completed.
5. Restart the appliance. At the prompt, type:
   reboot

   Are you sure you want to restart NetScaler (Y/N)? [N]:Y
6. Verify that the update is successful. At the prompt, type:
   show fips

   The firmware version displayed in the output should be 2.2. For example:

   ```
   > sh fips
    FIPS HSM Info:
     HSM Label  : NetScaler FIPS
     Initialization  : FIPS-140-2 Level-2
     HSM Serial Number : 2.1G1207-IC002429
     HSM State  : 2
   ```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
HSM Model  : NITROX XL CN1620-NFBE

Hardware Version : 2.0-G
Firmware Version : 2.2
Firmware Build      : NFBE-FW-2.2-130013
Max FIPS Key Memory : 3996
Free FIPS Key Memory : 3982
Total SRAM Memory : 467348
Free SRAM Memory : 50472
Total Crypto Cores : 3
Enabled Crypto Cores : 1
Done
```

To update the FIPS firmware to version 2.2 on appliances in a high availability pair

1. Log on to the secondary node and perform the update as described in To update the FIPS firmware to version 2.2 on a standalone NetScaler.
   Force the secondary node to become primary. At the prompt, type:

   force failover

   and press **Y** at the confirmation prompt.

2. Log on to the new secondary node (old primary) and perform the update as described in To update the FIPS firmware to version 2.2 on a standalone NetScaler.
3. Force the new secondary node to become primary again. At the prompt, type:
   force failover

   and press **Y** at the confirmation prompt.

To update the FIPS firmware to version 1.1 on a standalone appliance

1. Download the nfb_firmware-r1235_100604 and nfb_firmware-r1235_100604.sign files, to the same directory on the appliance, from the download page on www.citrix.com.

2. Log on to the appliance by using the administrator credentials.

3. At the prompt, type:

   update ssl fips -fipsFW /<full path to the file>/nfb_firmware-r1235_100604

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Resetting a Locked HSM

Jan 01, 2018

The HSM becomes locked (no longer operational) if you change the SO password, restart the appliance without saving the configuration, and make three unsuccessful attempts to change the password. This is a security measure for preventing unauthorized access attempts and changes to the HSM settings.

Important: To avoid this situation, save the configuration after initializing the HSM.
If the HSM is locked, you must reset the HSM and restart the appliance to restore the default passwords. You can then use the default passwords to access the HSM and configure it with new passwords. When finished, you must save the configuration and restart the appliance.

Caution: Do not reset the HSM unless it has become locked.

To reset a locked HSM by using the command line interface

At the command prompt, type the following commands to reset and re-initialize a locked HSM:

- reset ssl fips
- reboot -warm
- set ssl fips -initHSM Level-2 <new SO password> <old SO password> <user password> [-hsmLabel <string>]
- save ns config
- reboot -warm

Example

```
reset fips
reboot -warm
set fips -initHSM Level-2 newsopin123 sopin123 userpin123 -hsmLabel NSFIPS
saveconfig
reboot -warm
```

Note: The SO and User passwords are the default passwords.

To reset a locked HSM by using the configuration utility

1. Navigate to Traffic Management > SSL > FIPS
2. In the details pane, on the FIPS Info tab, click Reset FIPS.
3. Configure the HSM, as described in Configuring the HSM.
4. In the details pane, click Save.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

FIPS Approved Algorithms and Ciphers

Mar 24, 2017

For the updated list of ciphers supported on all FIPS appliances, see FIPS Approved CIphers.

For information about the Cipher/Protocol support matrix, see Cipher/Protocol Support Matrix.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Support for Thales nShield HSM

Jan 01, 2018

Note: This feature is available from release 11, build 62.10.

A non-FIPS NetScaler appliance stores the server's private key on the hard disk. On a FIPS appliance, the key is stored in a cryptographic module known as hardware security module (HSM). Storing a key in the HSM protects it from physical and software attacks. In addition, the keys are encrypted by using special FIPS approved ciphers.

Only the NetScaler MPX 9700/10500/12500/15500 FIPS appliances support a FIPS card. Support for FIPS is not available on other MPX appliances, or on the SDX and VPX appliances. This limitation is addressed by supporting a Thales nShield® Connect external HSM on all NetScaler MPX, SDX, and VPX appliances except the MPX 9700/10500/12500/15500 FIPS appliances.

Thales nShield Connect is an external FIPS-certified network-attached HSM. With a Thales HSM, the keys are securely stored as application key tokens on a remote file server (RFS) and can be reconstituted inside the Thales HSM only.

If you are already using a Thales HSM, you can now use a NetScaler ADC to optimize, secure, and control the delivery of all enterprise and cloud services.

Note:

- Thales HSMs comply with FIPS 140-2 Level 3 specifications, while the MPX FIPS appliances comply with level 2 specifications.
- You cannot decrypt the trace while using the Thales HSM, because the response from the HSM to the NetScaler appliance is encrypted and only the Hardserver can read it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Architecture Overview

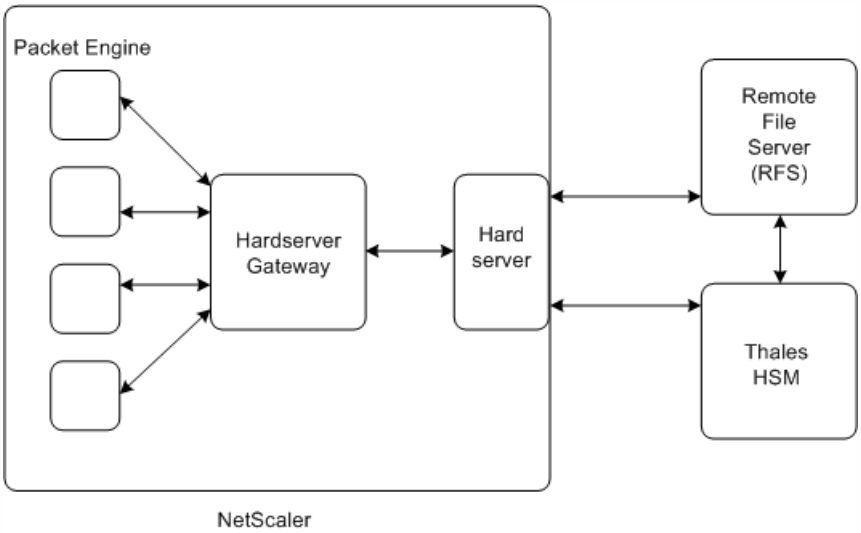
Aug 17, 2015

The three entities that are part of a NetScaler-Thales deployment are a Thales nShield Connect module, a remote file server (RFS), and a NetScaler ADC.

The Thales nShield Connect is a network-attached hardware security module. The RFS is used to configure the HSM and to store the encrypted key files.

Hardserver, a proprietary daemon provided by Thales, is used for communication between the client (ADC), the Thales HSM, and the RFS. It uses the IMPATH secure communication protocol. A gateway daemon, called the Hardserver Gateway, is used to communicate between the NetScaler packet engine and the Hardserver.

Note: The terms Thales nShield Connect, Thales HSM, and HSM are used interchangeably in this documentation.
The following figure illustrates the interaction between the different components.

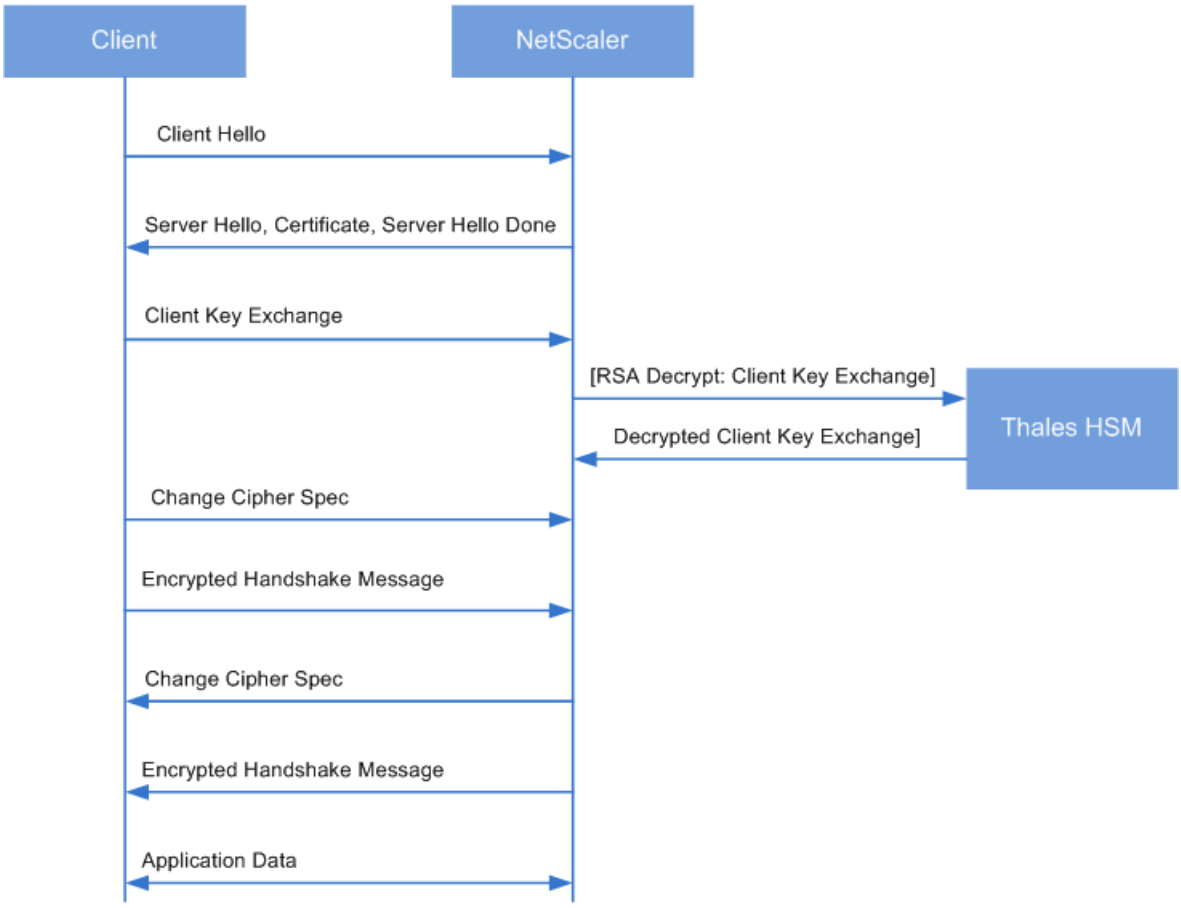


In a typical deployment, the RFS is used to securely store keys generated by the HSM. After the keys are generated, you can securely transfer them to the ADC and then use the NetScaler configuration utility or command line to load the keys to the HSM. A virtual server on the ADC uses Thales to decrypt the client key exchange to complete the SSL handshake. Thereafter, all the SSL operations are performed on the ADC.

Note: The terms keys and application key tokens are used interchangeably in this documentation.
The following figure illustrates the packet flow in the SSL handshake with the Thales HSM.

Figure 1. SSL Handshake Packets Flow Diagram with NetScaler Using Thales HSM

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Note: The communication between the ADC and the HSM uses a Thales proprietary communication protocol, called IMPATH.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Prerequisites

Aug 17, 2015

Before you can use a Thales nShield Connect with a NetScaler ADC, make sure that the following prerequisites are met:

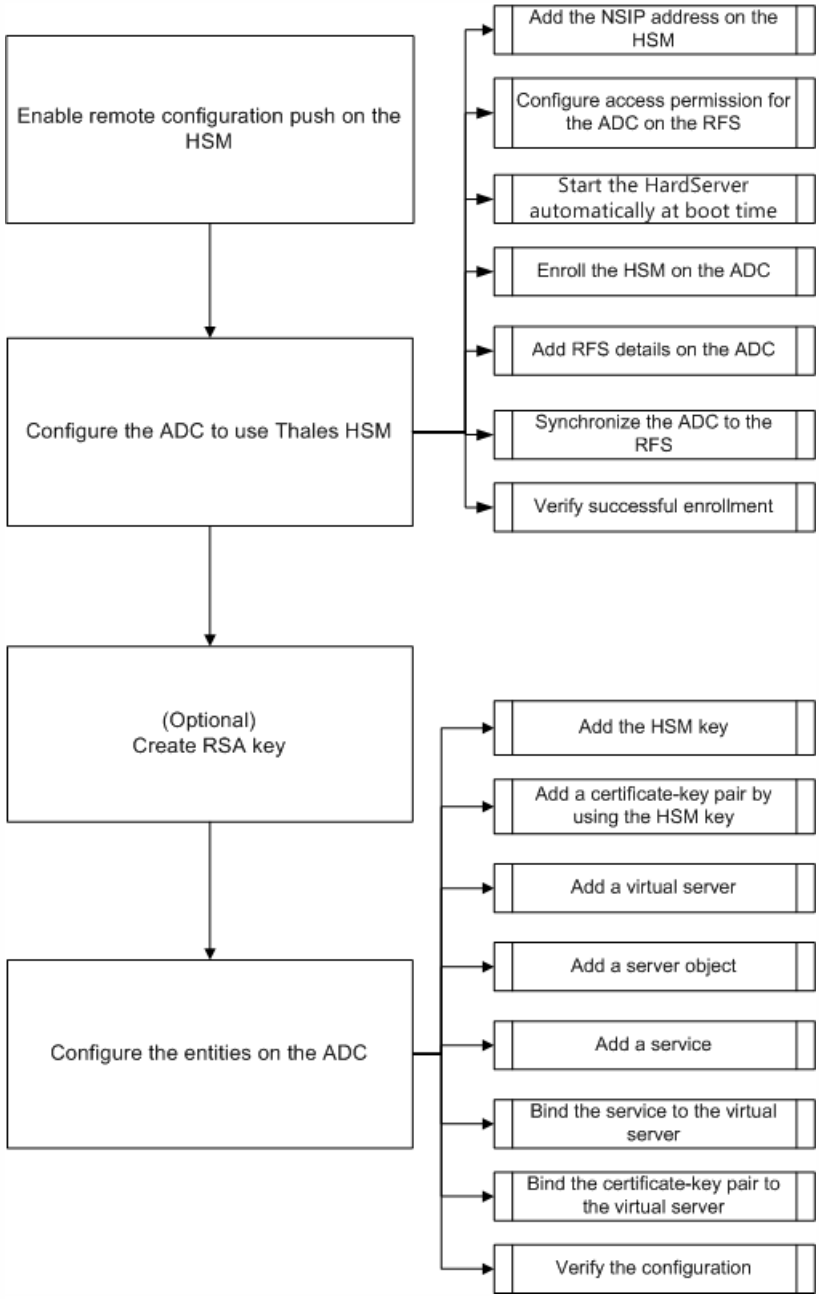
- A Thales nShield Connect device is installed in the network, ready to use, and accessible to the NetScaler ADC. That is, the NetScaler IP (NSIP) address is added as an authorized client on the HSM.
- A usable Security World exists. Security World is a unique key management architecture used by the Thales nShield line of HSMs. It protects and manages keys as application key tokens, enabling unlimited key capacity, and automatic key backup and recovery. For more information about creating a Security World, see the nShield Connect Quick Start Guide from Thales. You can also find the guide in the CD provided with the Thales HSM module at CipherTools-linux-dev-xx.xx.xx/document/nShield_Connect_Quick_Start_Guide.pdf.
  Note: Softcard or token/OCS protected keys are currently not supported on the NetScaler ADC.
- Licenses are available to support the number of clients that will be connected to the Thales HSM. The ADC and RFS are clients of the HSM.
- A remote file server (RFS) is installed in the network and is accessible to the NetScaler ADC.
- The Thales nShield Connect device, the RFS, and the NetScaler ADC can initiate connections with each other through port 9004.
- You are using NetScaler release 10.5 build 52.1115.e or later.
- The NetScaler appliance does not contain a FIPS Cavium card.
  Important: Thales HSM is not supported on the MPX 9700/10500/12500/15500 FIPS appliances.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the ADC-Thales Integration

Jan 01, 2018

The following flowchart depicts the tasks that you need to perform to use Thales HSM with a NetScaler ADC:



As shown in the above flowchart, you perform the following tasks:

1. Enable remote configuration push on the HSM.
2. Configure the ADC to use the Thales HSM.
   - Add the NSIP address on the HSM.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Configure access permission for the ADC on the RFS.
- Configure automatic start of the Hardserver at boot time.
- Enroll the HSM on the ADC.
- Add RFS details on the ADC.
- Synchronize the ADC to the RFS.
- Verify that Thales HSM is successfully enrolled on the ADC.

3. (Optional) Create an HSM RSA key.
4. Configure the entities on the NetScaler ADC.
   - Add the HSM key.
   - Add a certificate-key pair by using the HSM key.
   - Add a virtual server.
   - Add a server object.
   - Add a service.
   - Bind the service to the virtual server.
   - Bind the certificate-key pair to the virtual server.
   - Verify the configuration.

1. Configure the Thales HSM

You must specify the IP address of the RFS on the Thales HSM so that it accepts the configuration that the RFS pushes to it. Use the nShield Connect front panel on the Thales HSM to perform the following procedure.

To specify the IP address of a remote computer on the Thales HSM

1. Navigate to System Configuration > Config file options > Allow auto push.
2. Select ON, and specify the IP address of the computer (RFS) from which to accept the configuration.

Enabling Remote Configuration Push on the HSM

You must specify the IP address of the RFS on the Thales HSM so that it accepts the configuration that the RFS pushes to it. Use the nShield Connect front panel on the Thales HSM to perform the following procedure.

To specify the IP address of a remote computer on the Thales HSM

1. Navigate to System Configuration > Config file options > Allow auto push.
2. Select ON, and specify the IP address of the computer (RFS) from which to accept the configuration.

Configure the ADC to use the Thales HSM

Sample values used in this documentation:

NSIP address=10.217.2.43

Thales HSM IP address=10.217.2.112

RFS IP address=10.217.2.6

Add the NetScaler IP (NSIP) Address on the HSM

Typically you use the nShield Connect front panel to add clients to the HSM. For more information, see the nShield Connect Quick Start Guide.
Alternately, use the RFS to add the ADC as a client to the HSM. To do this, you must add the NSIP address in the HSM configuration on the RFS, and then push the configuration to the HSM. Before you can do this, you must know the electronic serial number (ESN) of the HSM.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To get the ESN of your HSM, run the following command on the RFS:

root@ns# /opt/nfast/bin/anonkneti <Thales HSM IP address>

Example
root@ns# /opt/nfast/bin/anonkneti 10.217.2.112
BD17-C807-58D9 5e30a698f7bab3b2068ca90a9488dc4e6c78d822
The ESN number is BD17-C807-58D9.

After you have the ESN number, use an editor, such as vi, to edit the HSM configuration file on the RFS.

vi /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config

In the hs_clients section, add the following entries:

# Amount of data in bytes to encrypt with a session key before session key# renegotiation, or 0 for unlimitied. (default=1024*1024*8b=8Mb).
# datalimit=INT
addr=10.217.2.43
clientperm=unpriv
keyhash=0000000000000000000000000000000000000000
esn=
timelimit=86400
datalimit=8388608

Note: Include one or more hyphens as delimiters to add multiple entries in the same section.
To push the configuration to the HSM, run the following command on the RFS:

/opt/nfast/bin/cfg-pushnethsm --address=<Thales HSM IP address> --force /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config

Example
/opt/nfast/bin/cfg-pushnethsm --address=10.217.2.112 --force
/opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config

Configure Access Permission for the ADC on the RFS

To configure access permission for the ADC on the RFS, run the following command on the RFS:
/opt/nfast/bin/rfs-setup --force -g --write-noauth <NetScaler IP address>

Example
[root@localhost bin]# /opt/nfast/bin/rfs-setup --force -g --write-noauth 10.217.2.43
Adding read-only remote_file_system entries
Ensuring the directory /opt/nfast/kmdata/local exists
Adding new writable remote_file_system entries
Ensuring the directory /opt/nfast/kmdata/local/sync-store exists
Saving the new config file and configuring the hardserver
Done
Verify that the ADC can reach both the RFS and Thales HSM by using port 9004.

Configure Automatic Start of the Hardserver at Boot Time

Create a file and then restart the appliance. Now, whenever you restart the appliance, and if this file is found, the Hardserver is automatically started.

At the shell prompt, type:
touch /var/opt/nfast/bin/thales_hsm_is_enrolled

At the command prompt, type:
reboot

Enroll the HSM on the ADC

Change directory to /var/opt/nfast/bin.
To add HSM details into the ADC configuration, run the following command on the ADC:

nethsmenroll --force <Thales_nShield_Connect_ip_address> $(anonkneti <Thales_nShield_Connect_ip_address>)

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

```
root@ns# ./nethsmenroll --force 10.217.2.112 $(anonkneti 10.217.2.112)
OK configuring hardserver's nethsm imports
```

This step adds the following entries after the line # ntoken_esn=ESN in the nethsm_imports section of the /var/opt/nfast/kmdata/config/config file.

```
…
local_module=0
remote_ip=10.217.2.112
remote_port=9004
remote_esn=BD17-C807-58D9
keyhash=5e30a698f7bab3b2068ca90a9488dc4e6c78d822
timelimit=86400
datalimit=8388608
privileged=0
privileged_use_high_port=0
ntoken_esn=
```

Change directory to /var/opt/nfast/bin and run the following command on the ADC:

```
touch "thales_hsm_is_enrolled"
```

Note: To remove an HSM that is enrolled on the ADC, type: ./nethsmenroll –-remove <NETHSM-IP>

Add RFS details on the ADC

To add RFS details, change directory to /var/opt/nfast/bin/ and then run the following command:

```
./rfs-sync --no-authenticate --setup <rfs_ip_address>
```

Example

```
./rfs-sync --no-authenticate --setup 10.217.2.6
No current RFS synchronization configuration.
Configuration successfully written; new config details:
Using RFS at 10.217.2.6:9004: not authenticating.
```

This step adds the following entries after the # local_esn=ESN line in the rfs_sync_client section of the /var/opt/nfast/kmdata/config/config file.

```
……
remote_ip=10.217.2.6
remote_port=9004
use_kneti=no
local_esn=
```

Note: To remove an RFS that is enrolled on the ADC, type: ./rfs_sync –remove

Example

```
./rfs-sync --no-authenticate --setup 10.217.2.6
No current RFS synchronization configuration.
Configuration successfully written; new config details:
Using RFS at 10.217.2.6:9004: not authenticating.
```

This step adds the following entries after the # local_esn=ESN line in the rfs_sync_client section of the /var/opt/nfast/kmdata/config/config file.

```
……
remote_ip=10.217.2.6
remote_port=9004
use_kneti=no
local_esn=
```

Synchronize the ADC to the RFS

To synchronize all the files, change directory to /var/opt/nfast/bin and then run the following command on the ADC:

```
./rfs-sync –-update
```

This command fetches all the World files, module files, and key files from the /opt/nfast/kmdata/local directory on the RFS and puts them into the /var/opt/nfast/kmdata/local directory on the ADC. Citrix recommends that you manually copy the World files, the module_XXXX_XXXX_XXXX files, where XXXX_XXXX_XXXX is the ESN of the enrolled HSM, and only the required RSA key and certificate files.

Verify that the Thales HSM is successfully enrolled on the ADC

After you synchronize the ADC to the RFS, do the following:

- Verify that the local Hardserver is UP and running. (nCipher server running).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Get the state of the configured HSMs, and verify that the values for the n_modules (number of modules) field and the km info fields are non-zero.
- Verify that the HSM is enrolled correctly and is usable (state 0x2 Usable) by the ADC.
- Load tests using sigtest run properly.

Change directory to /var/opt/nfast/bin, and at the shell prompt, run the following commands:
root@ns# ./chkserv root@ns# ./nfkminfo root@ns# ./sigtest
See Appendix for an example.

Create an HSM RSA Key

Only RSA keys are supported as HSM keys.

Note: Skip this step if keys are already present in the /opt/nfast/kmdata/local folder on the RFS.
Create an RSA key, a self-signed certificate, and a Certificate Signing Request (CSR). Send the CSR to a certificate authority to get a server certificate.

The following files are created in the example below:
- Embed RSA key: key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2c78
- Self-Signed Certificate: example_selfcert
- Certificate Signing Request: example_req

Note: The generatekey command is supported in strict FIPS 140-2 Level 3 Security World. An administrator card set (ACS) or an operator card set (OCS) is needed to control many operations, including the creation of keys and OCSs. When you run the generatekey command, you are prompted to insert an ACS or OCS card. For more information about strict FIPS 140-2 Level 3 Security World, see the nShield Connect User Guide.
The following example uses Level-2 Security World. In the example, the commands are in boldface type.

Example

```
[root@localhost bin]# ./generatekey embed
size: Key size? (bits, minimum 1024) [1024] > 2048
OPTIONAL: pubexp: Public exponent for RSA key (hex)? []
>
embedsavefile: Filename to write key to? []
> example
plainname: Key name? [] > example
x509country: Country code? [] > US
x509province: State or province? [] > CA
x509locality: City or locality? [] > Santa Clara
x509org: Organisation? [] > Citrix
x509orgunit: Organisation unit? [] > NS
x509dnscommon: Domain name? [] > www.citrix.com
x509email: Email address? [] > example@citrix.com
nvram: Blob in NVRAM (needs ACS)? (yes/no) [no] >
digest: Digest to sign cert req with? (md5, sha1, sha256, sha384, sha512)
  [default sha1] > sha512
key generation parameters:
 operation     Operation to perform          generate
 application   Application                   embed
 verify        Verify security of key        yes
 type          Key type                      RSA
 size          Key size                      2048
 pubexp        Public exponent for RSA key (hex)
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 embedsavefile  Filename to write key to          example
 plainname      Key name                          example
 x509country    Country code                      US
 x509province   State or province                 CA
 x509locality   City or locality                 Santa Clara
 x509org        Organisation                      Citrix
 x509orgunit    Organisation unit                 NS
 x509dnscommon  Domain name                       www.citrix.com
 x509email      Email address                     example@citrix.com
 nvram          Blob in NVRAM (needs ACS)         no
 digest         Digest to sign cert req with      sha512
Key successfully generated.
Path to key: /opt/nfast/kmdata/local/key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2c78
You have new mail in /var/spool/mail/root
```

Result

You have created a CSR (example_req), a self-signed certificate (example_selfcert), and an application key token file in embed format (/opt/nfast/kmdata/local/key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2c78)

Because the ADC supports keys in simple format only, you must convert the embed key to a simple key.

To convert the embed key to a simple key, run the following command on the RFS:

```
[root@localhost bin]# ./generatekey -r simple
from-application: Source application? (embed, simple) [embed] > embed
from-ident: Source key identifier? (c6410ca00af7e394157518cb53b2db46ff18ce29,
                                   2ed5428aaeae1e159bdbd63f25292c7113ec2c78)
  [default c6410ca00af7e394157518cb53b2db46ff18ce29]
> 2ed5428aaeae1e159bdbd63f25292c7113ec2c78
ident: Key identifier? [] > examplersa2048key
plainname: Key name? [] > examplersa2048key
key generation parameters:
 operation         Operation to perform    retarget
 application       Application             simple
 verify            Verify security of key  yes
 from-application  Source application      embed
 from-ident        Source key identifier   2ed5428aaeae1e159bdbd63f25292c7113ec2c78
 ident             Key identifier          examplersa2048key
 plainname         Key name                examplersa2048key
Key successfully retargetted.
Path to key: /opt/nfast/kmdata/local/key_simple_examplersa2048key
```

> **Important**
>
> When prompted for the source key identifier, enter **2ed5428aaeae1e159bdbd63f25292c7113ec2c78** as the embed key.

Result

A key with the prefix key_simple (for example key_simple_examplersa2048key) is created.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: examplersa2048key is the key identifier (ident) and is referred to as the HSM key name on the ADC. A key identifier is unique. All the simple files have the prefix key_simple.

Configure the Entities on the ADC

Before the ADC can process traffic, you must do the following:

1. Enable features.
2. Add a subnet IP (SNIP) address.
3. Add the HSM key to the ADC.
4. Add a certificate-key pair by using the HSM key.
5. Add a virtual server.
6. Add a server object.
7. Add a service.
8. Bind the service to the virtual server.
9. Bind the certificate-key pair to the virtual server.
10. Verify the configuration.

Enable features on the ADC

Licenses must be present on the ADC before you can enable a feature.

To enable a feature by using the command line

At the command prompt, run the following commands:

- enable feature lb
- enable feature ssl

To enable a feature by using the configuration utility

Navigate to System > Settings and, in the Modes and Features group, select Configure basic features, and then select SSL Offloading.

Add a subnet IP address

For more information about subnet IP addresses, see Configuring Subnet IP Addresses (SNIPs).

To add a SNIP address and verify the configuration by using the command line

At the command prompt, run the following commands:

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip

Example

```
> add ns ip 192.168.17.253 255.255.248.0 -type SNIP
Done
> show ns ip
     Ipaddress       Traffic Domain  Type          Mode    Arp     Icmp    Vserver  State
     ---------       --------------  ----          ----    ---     ----    -------  ------
1)    192.168.17.251   0            NetScaler IP    Active   Enabled  Enabled  NA       Enabled
2)    192.168.17.252   0            VIP             Active   Enabled  Enabled  Enabled  Enabled
3)    192.168.17.253   0            SNIP            Active   Enabled  Enabled  NA       Enabled
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Done

To add a SNIP address and verify the configuration by using the configuration utility

Navigate to System > Network > IPs, add an IP address, and select the IP Type as Subnet IP.

Copy the HSM key and certificate to the ADC

Use a secure file transfer utility to securely copy the key (key_simple_examplersa2048key) to the /var/opt/nfast/kmdata/local folder, and the certificate (example_selfcert) to the /nsconfig/ssl folder on the ADC.

Add the key on the ADC

All the keys have a key-simple prefix. When adding the key to the ADC, use the ident as the HSM key name. For example, if the key that you added is key_simple_XXXX, the HSM key name is XXXX.

Important:

- The HSM key name must be the same as the ident that you specified when you converted an embed key to a simple key format.
- The keys must be present in the /var/opt/nfast/kmdata/local/ directory on the ADC.

To add an HSM key by using the command line

At the shell prompt, run the following command:

add ssl hsmKey <hsmKeyName> -key <string>

Example

```
> add ssl hsmKey examplersa2048key –key key_simple_examplersa2048key
Done
```

To add an HSM key by using the configuration utility

Navigate to Traffic Management > SSL > HSM, and add an HSM key.

Add a certificate-key pair on the ADC

For information about certificate-key pairs, see Adding or Updating a Certificate-Key Pair.

To add a certificate-key pair by using the command line

At the command prompt, run the following command:

add ssl certKey <certkeyName> -cert <string> -hsmKey <string>

Example

```
> add ssl certKey key22 -cert example_selfcert -hsmKey examplersa2048key
Done
```

To add a certificate-key pair by using the configuration utility

Navigate to Traffic Management > SSL > Certificates, and add a certificate-key pair.

Add a virtual server

For information about a virtual server, see Configuring an SSL-Based Virtual Server.

To configure an SSL-based virtual server by using the command line

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, run the following command:

add lb vserver <name> <serviceType> <IPAddress> <port>

Example
> add lb vserver v1 SSL 192.168.17.252 443
To configure an SSL-based virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers, create a virtual server, and specify the protocol as SSL.

Add a server object

Before you can add a server object on the ADC, make sure that you have created a backend server. The following example uses the built-in python HTTP Server module on a Linux system.
Example

%python –m SimpleHTTPServer 80

To add a server object by using the command line

At the command prompt, run the following command:

add server <name> <IPAddress>

Example
> add server s1 192.168.17.246
To add a server object by using the configuration utilty

Navigate to Traffic Management > Load Balancing > Servers, and add a server.

Add a service

For more information, see Configuring Services.
To configure a service by using the command line

At the command prompt, run the following command:

add service <name> <serverName> <serviceType> <port>

Example
> add service sr1 s1 HTTP 80
To configure a service by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, and create a service.

Bind the service to the virtual server

For more information, see Binding Services to an SSL-Based Virtual Server.
To bind a service to a virtual server by using the command line

At the command prompt, run the following command:

bind lb vserver <name> <serviceName>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

```
> bind lb vserver v1 sr1
```

To bind a service to a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server, and click in the Services pane to bind a service to the virtual server.

Bind the certificate-key pair to the virtual server on the ADC

For more information, see Binding the Certificate-Key Pair to the SSL-Based Virtual Server.

To bind a certificate-key pair to a virtual server by using the command line

At the command prompt, run the following command:

```
bind ssl vserver <vServerName> -certkeyName <string>
```

Example

```
> bind ssl vserver v1 -certkeyName key22
Warning: Current certificate replaces the previous binding
```

To bind a certificate-key pair to a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open an SSL virtual server and, in Advanced Settings, click SSL Certificate.
3. Bind a server certificate to the virtual server.

Verify the configuration

To view the configuration by using the command line

At the command prompt, run the following commands:

- show lb vserver <name>
- show ssl vserver <vServerName>

Example

```
> show lb vserver v1
    v1 (192.168.17.252:443) - SSL   Type: ADDRESS
    State: UP
    Last state change was at Wed Oct 29 03:11:11 2014
    Time since last state change: 0 days, 00:01:25.220
    Effective State: UP
    Client Idle Timeout: 180 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    Appflow logging: ENABLED
    No. of Bound Services :  1 (Total)      1 (Active)
    Configured Method: LEASTCONNECTION
    Current Method: Round Robin, Reason: Bound service's state changed to UP
    Mode: IP
    Persistence: NONE
    Vserver IP and Port insertion: OFF
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Push: DISABLED  Push VServer:
    Push Multi Clients: NO
    Push Label Rule: none
    L2Conn: OFF
    Skip Persistency: None
    IcmpResponse: PASSIVE
    RHIstate: PASSIVE
    New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0
    Mac mode Retain Vlan: DISABLED
    DBS_LB: DISABLED
    Process Local: DISABLED
    Traffic Domain: 0

1) sr1 (192.168.17.246: 80) - HTTP State: UP    Weight: 1
Done
>

> sh ssl vserver v1
    Advanced SSL configuration for VServer v1:
    DH: DISABLED
    Ephemeral RSA: ENABLED        Refresh Count: 0
    Session Reuse: ENABLED        Timeout: 120 seconds
    Cipher Redirect: DISABLED
    SSLv2 Redirect: DISABLED
    ClearText Port: 0
    Client Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: DISABLED
    SSLv2: DISABLED  SSLv3: DISABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED
    Push Encryption Trigger: Always
    Send Close-Notify: YES

    ECC Curve: P_256, P_384, P_224, P_521

1)    CertKey Name: key22      Server Certificate

1)    Cipher Name: DEFAULT
    Description: Predefined Cipher Alias
 Done
```

To view the configuration by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers, and double-click an SSL virtual server to open it and view the configuration.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Limitations

Jan 25, 2017

- SSL version 3 (SSLv3) is not supported on an MPX appliance but is supported on a VPX virtual appliance. A VPX instance provisioned on an SDX appliance supports SSLv3 only if an SSL chip is not assigned to the instance.
- ECDHE/DHE ciphers and Export ciphers are not supported.
- SSL server key exchange using HSM keys is not supported.
- If you have added or removed keys after you last saved the configuration, you must save the configuration before you perform a warm restart. If you do not save the configuration, there will be a key mismatch between the ADC and the HSM.
- You cannot bind an HSM key to a DTLS virtual server.
- You cannot bind a certificate-key pair that is created by using an HSM key to an SSL service.
- You cannot use the NetScaler configuration utility to enroll the ADC as a client of the HSM or check the status of the HSM from the configuration utility.
- From release 11 build 62.x, SSL renegotiation is supported.
- You cannot sign OCSP requests by using a certificate-key pair that is created by using an HSM key.
- A certificate bundle with HSM keys is not supported.
- An error does not appear if the HSM key and certificate do not match. Therefore, while adding a certificate-key pair, you need to make sure that the HSM key and certificate match.
- Clustering and admin partitions are not supported.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Appendix

Aug 17, 2015

Example

Note: In the following example, the commands are in boldface type.

```
root@ns# ./chkserv
nCipher server running
root@ns# ./nfkminfo
World
 generation  2
 state       0x17a70000 Initialised Usable Recovery PINRecovery !ExistingClient RTC NVRAM FTO !AlwaysUseStrongPrimes SEEDebug
 n_modules   1
 hknso       cbec8c0c56c6b5e76b73147ef02d34a661eaa044
 hkm     bbb8d4839da5782be4d092735a7535538834dc91 (type Rijndael)
 hkmwk       1d572201be533ebc89f30fdd8f3fac6ca3395bf0
 hkre        01f21ecf43933ffdd45e74c3883525176c5c439c
 hkra        ac8ec5ee6bce00991bd97adce2091d9739b9b452
 hkmc        cf1b509abaad91995ed202d8f36613fc99433155
 hkp         c20910b2ed1ca62d6a2b0db67052a05f7bbfeb43
 hkrtc       bd811020a7c2f8df435a481c3767a89c2e13bc4f
 hknv        278b8012e48910d518a9ee91cff57233fb0c9093
 hkdsee      12230b0e31e3cec66324c0815f782cfb9249edd5
 hkfto       89dd6250b3d6149bcd15606f4553085e2fd6271a
 hkmnull     0100000000000000000000000000000000000000
 ex.client   none
 k-out-of-n  1/2
 other quora m=1 r=1 p=1 nv=1 rtc=1 dsee=1 fto=1
 createtime  2014-02-28 21:05:32
 nso timeout 10 min
 ciphersuite DLf1024s160mRijndael

Module #1
 generation 2
 state      0x2 Usable
 flags      0x10000 ShareTarget
 n_slots    2
 esn        BD17-C807-58D9
 hkml       70289a6edba00ddc7e3f6d6f5a49edc963e822f2

Module #1 Slot #0 IC 0
 generation    1
 phystype      SmartCard
 slotlistflags 0x2 SupportsAuthentication
 state         0x2 Empty
 flags         0x0
 shareno       0
 shares
 error         OK
No Cardset

Module #1 Slot #1 IC 0
 generation    1
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
 phystype      SoftToken
 slotlistflags 0x0
 state         0x2 Empty
 flags         0x0
 shareno       0
 shares
 error         OK
No Cardset

No Pre-Loaded Objects

root@ns# ./sigtest
 Hardware module #1 speed index 5792 recommended minimum queue 19
Found 1 module; using 19 jobs
Making 1024-bit RSAPrivate key on module #1;
  using Mech_RSApPKCS1 and PlainTextType_Bignum.
Generated and exported key from module #1.
Imported keys on module #1
 1,      3059 1223.6, 3059 overall
 2,      8698 2989.76, 4349 overall
 3,     14396 4073.06, 4798.67 overall
 4,     20091 4721.83, 5022.75 overall
 5,     25799 5116.3, 5159.8 overall
 6,     31496 5348.58, 5249.33 overall
 7,     37192 5487.55, 5313.14 overall
 8,     42780 5527.73, 5347.5 overall
 9,     45777 4515.44, 5086.33 overall
 10,     51457 4981.26, 5145.7 overall
 11,     57151 5266.36, 5195.55 overall
 12,     62813 5424.61, 5234.42 overall
 13,     68496 5527.97, 5268.92 overall
 14,     74182 5591.18, 5298.71 overall
 15,     79832 5614.71, 5322.13 overall
 16,     85518 5643.23, 5344.88 overall
 17,     88412 4543.54, 5200.71 overall
 18,     94086 4995.72, 5227 overall
 19,     99778 5274.23, 5251.47 overall
 20,    105469 5440.94, 5273.45 overall
 21,    111133 5530.16, 5292.05 overall
 22,    116838 5600.1, 5310.82 overall
 23,    122522 5633.66, 5327.04 overall
 24,    128175 5641.4, 5340.62 overall
 25,    131072 4543.64, 5242.88 overall
 26,    136762 5002.18, 5260.08 overall
 27,    142415 5262.51, 5274.63 overall
 28,    148125 5441.51, 5290.18 overall
 29,    153816 5541.3, 5304 overall
 30,    159414 5563.98, 5313.8 overall
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Jul 22, 2013

If the SSL feature does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for Troubleshooting

Updated: 2013-07-22

For best results, use the following resources to troubleshoot an SSL issue on a NetScaler appliance:

- The relevant ns.log file
- The latest ns.conf file
- The messages file
- The relevant newnslog file
- Trace files
- A copy of the certificate files, if possible
- A copy of the key file, if possible
- The error message, if any

In addition to the above resources, you can use the Wireshark application customized for the NetScaler trace files to expedite troubleshooting.

Troubleshooting SSL Issues

Updated: 2013-07-22

To troubleshoot an SSL issue, proceed as follows:

- Verify that the NetScaler appliance is licensed for SSL Offloading and load balancing.
- Verify that SSL Offloading and load balancing features are enabled on the appliance.
- Verify that the status of the SSL virtual server is not displayed as DOWN.
- Verify that the status of the service bound to the virtual server is not displayed as DOWN.
- Verify that a valid certificate is bound to the virtual server.
- Verify that the service is using an appropriate port, preferably port 443.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL FAQs

Jan 21, 2018

For the list of FAQs, see SSL FAQs.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring an SDX 14000 FIPS Appliance

Mar 28, 2017

> **Note**
>
> This platform is supported in release 11.0 build 70.x and later.

A Citrix NetScaler SDX appliance is a multitenant platform on which you can provision and manage multiple virtual NetScaler instances. The SDX appliance addresses cloud computing and multitenancy requirements by allowing a single administrator to configure and manage the appliance and delegate the administration of each hosted instance to tenants.

A NetScaler SDX 14030/14060/14080 FIPS appliance provides the capabilities of an SDX appliance with FIPS functionality. It is equipped with a tamper-proof (tamper-evident) cryptographic module—a Cavium CNN3560-NFBE-G—designed to comply with the FIPS 140-2 Level-2 specifications. The Critical Security Parameters (CSPs), primarily the server's private-key, are securely stored and generated inside the cryptographic module, also referred to as the Hardware Security Module (HSM). The CSPs are never accessed outside the boundaries of the HSM. Only the superuser (nsroot) can perform operations on the keys stored inside the HSM.

A NetScaler SDX 14030/14060/14080 FIPS appliance contains one FIPS HSM module with 63 cores. The FIPS HSM module can be partitioned up to a maximum of 32 partitions. The SDX administrator can assign dedicated key storage, cryptographic resources, and number of crypto SSL FIPS cores to each partition. Keys and resources allocated to a partition are dedicated and secure and cannot be accessed or shared by another partition.

The FIPS HSM partition that you create can be assigned or attached to a VPX instance at the time of provisioning the instance, or later by editing the instance. The FIPS partition created and attached to an instance acts like a virtual HSM module for that particular instance.

The VPX instances on an SDX 14030/14060/14080 FIPS appliance are assigned a FIPS virtual function (VF) partition, which is treated as an isolated FIPS virtual card or HSM. Therefore, the steps to configure a FIPS partition inside a VPX instance are similar to the steps to configure an MPX FIPS appliance. For compliance details, see the security policy details on the U.S. National Institute of Standards and Technology (NIST) website.

> **Important**
>
> Each key includes a private and a public key. As a result, it occupies two key spaces. Therefore, the maximum number of keys is limited to one less than half the key store size.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Limitations

Feb 16, 2017

1. SSL renegotiation using the SSLv3 protocol is not supported on the back end of an SDX FIPS appliance.
2. 1024-bit and 4096-bit keys and an exponent value of 3 are not supported.
3. Backup and restore is not supported.
4. Cluster and administrative domains are not supported.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Terminology

Feb 16, 2017

Zeroize: Reset the HSM. All the data on the HSM is deleted. This is a mandatory step before the HSM is initialized.

Initialize: Set the HSM capabilities. The NetScaler SDX FIPS appliance complies with FIPS-140-2 level 2. You can create partitions after you initialize the chip.

Key store size: Number of keys that can be stored on a partition. A maximum of 102235 keys can be specified. The maximum number of keys that can be stored is one less than half the number specified. For example, if you specify 100, you can create only 49 keys because one of the keys is the RSA key pair that consumes 2 key stores.

Crypto Core Capacity: Number of crypto cores assigned to a partition. A maximum of 63 cores are available.

SSL Context: Number of concurrent SSL connections that can be created on a partition.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Initializing the HSM

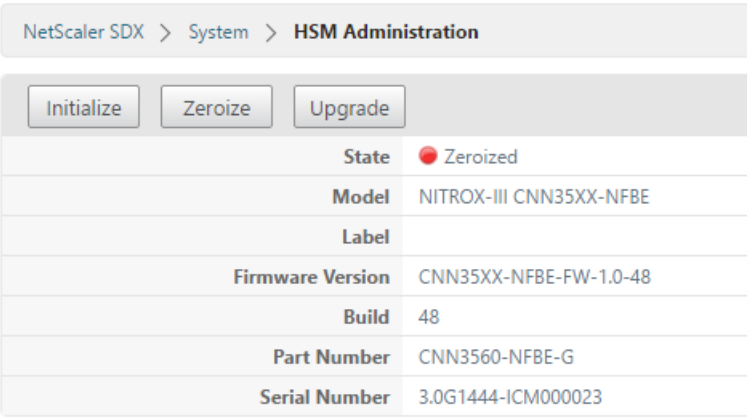
Oct 05, 2016

Before initializing the HSM, you must first zeroize it.

To zeroize the HSM by using the Management Service

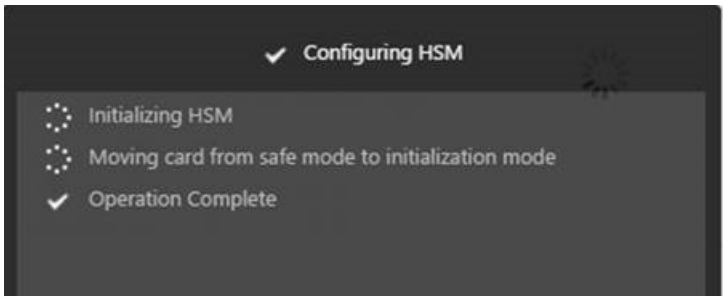
1. Open a browser and log on to the appliance.
2. On the **Configuration** tab, navigate to **System** > **HSM Administration**, and in the details plane, click **Zeroize**.

All data is wiped from the FIPS chip, and the state appears as "Zeroized." Any HSM partitions created earlier are deleted.



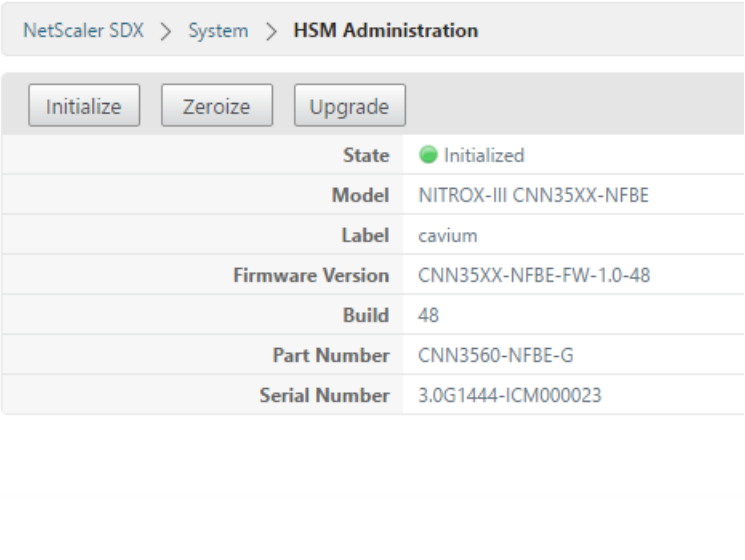
To initialize the HSM by using the Management Service

1. On the **Configuration** tab, navigate to **System** > **HSM Administration**, and in the details plane, click **Initialize**.
2. Type a new user name, specify a password, and click **OK**.



The card state appears as "Initialized."

© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating Partitions

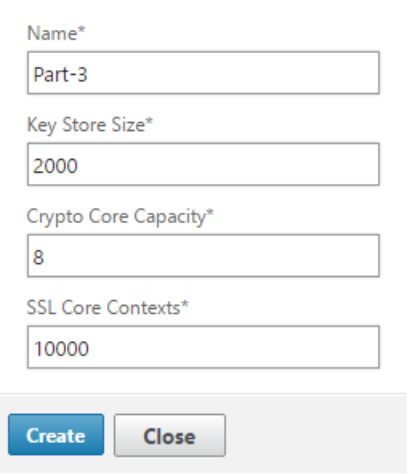
Oct 05, 2016

Create partitions for different tenants and specify the cryptographic resources for each partition. Each instance is assigned one partition, and a partition can be assigned to only one instance. Deleting an instance deletes the partition assigned to the instance. As a result, the partition data is also deleted and not left unsecured or accessible later. Number of keys and SSL context assignment depends on your application. For information about number of cores to assign, see the NetScaler datasheet.

> **Important**
>
> After you assign a key store size and cores to an HSM partition, you cannot change them at run time. You must first detach the partition from the instance.

To create a partition by using the Management Service

1. On the Configuration tab, navigate to **System** > **HSM Administration** > **Partitions**, and in the details plane, click **Add**.
2. Specify a name for the partition, and the resources to be assigned to this partition.
3. Click **OK**.



The summary page displays all the partitions that were created. Some partitions are assigned an instance while some are free partitions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler SDX > System > HSM Administration > **Partitions**

Total Keys	Available Keys	Total Crypto Cores	Available Crypto Cores	Total SSL Contexts	Available SSL Contexts
102,235	**97,035**	**63**	**23**	**1,000,000**	**610,000**

Add Edit Delete

Name	Key Store Size	Crypto Core Capacity	SSL Core Contexts	Instance Name
Part-3	2000	8	10000	
Part-4	200	2	10000	
Partition-1234	100	4	20000	
Partition-12345	300	4	20000	
Partition-5	300	8	100000	
Part-6	200	8	200000	
Part-1	100	2	10000	NSVPX-1-10.217.202.35
Part-2	2000	4	20000	NSVPX-2-10.217.202.36

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Provisioning a New Instance or Modifying an Existing Instance and Assigning a Partition

Oct 05, 2016

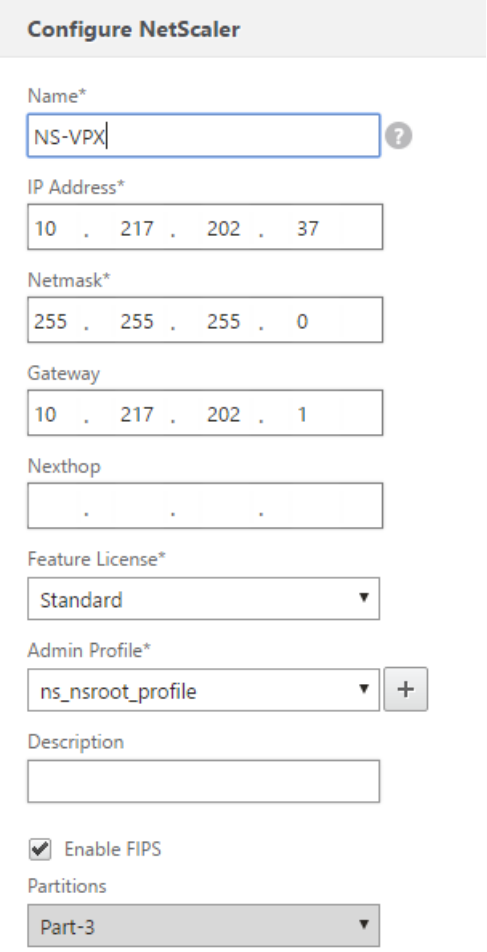
After creating the partitions, you must assign them to instances.

Important

- You can attach only one FIPS partition to an instance.
- An instance with a FIPS partition can be assigned only one CPU core.
- You can assign either a FIPS partition or an SSL core to an instance, but not both.

To provision a new instance or modify an existing instance

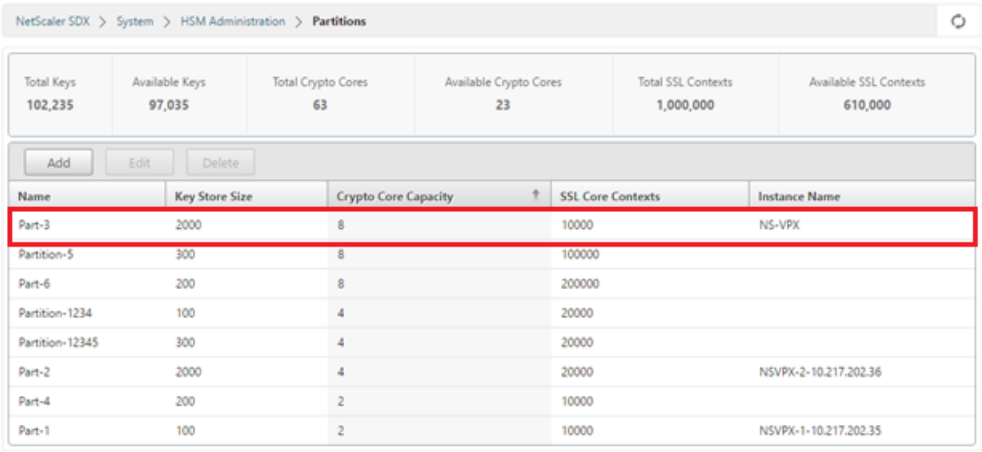
1. On the Configuration tab, navigate to **NetScaler** > **Instances**, and add or modify an instance.
2. Select **Enable FIPS**, and from the **Partitions** list, select a partition to attach to this instance.



You can verify that the partition is attached to an instance by using either the GUI or the CLI.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

In the GUI, navigate to **System** > **HSM Administration** > **Partitions**. The instance name attached to the partition is displayed.



To unassign a FIPS partition¸ navigate to **NetScaler** > **Instances**. Edit the instance and clear the **Enable FIPS** check box.

In the CLI, at the command prompt, type the following commands:

> **show fips**

FIPS Card is not configured

Done

>

If you see the following output, see the troubleshooting section for debugging.

ERROR: Operation not permitted - no FIPS card present in the system

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the HSM for an Instance on an SDX 14030/14060/14080 FIPS Appliance

Oct 05, 2016

You must first check the state of your FIPS card to verify that the driver loaded correctly, and then initialize the card.

At the command prompt, type:

> show fips

FIPS Card is not configured

Done

If the driver is not loaded correctly, the message “ERROR: Operation not permitted - no FIPS card present in the system” appears.

Initializing the FIPS Card

> **Important**
>
> Verify that the /nsconfig/fips directory has successfully been created on the appliance.

Do not save the configuration before you restart the appliance for the third time.

Perform the following steps to initialize the FIPS card:

1. Reset the FIPS card.
2. Restart the appliance.
3. Set the security officer password for partitions 0 and 1, and the user password for partition 1.
   Note: The set or reset command takes more than 60 seconds to run.
4. Save the configuration.
5. Verify that the password encrypted key for the master partition (master_pek.key) has been created in the /nsconfig/fips/ directory.
6. Restart the appliance.
7. Verify that the FIPS card is UP.

To initialize the FIPS card by using the NetScaler command line

At the command prompt, type the following commands:

> **reset fips**

> **reboot**

> **set fips -initHSM Level-2** <soPassword> <oldsoPassword> <userPassword> -**hsmLabel** <string>

> **saveconfig**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **reboot**

> **show fips**

Example

```
> reset fips

 Done

> reboot

> set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS

This command will erase all data on the FIPS card. You must save the configuration (saveconfig) after executing this command. Do you w

Done

> saveconfig

Done

> reboot

> show fips

FIPS HSM Info:

HSM Label : NSFIPS

Initialization : FIPS-140-2 Level-2

HSM Serial Number : 3.0G1532-ICM000228

HSM State : 2

HSM Model : NITROX-III CNN35XX-NFBE
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Hardware Version : 0.0-G

Firmware Version : 1.0

Firmware Build : NFBE-FW-1.0-48

Max FIPS Key Memory : 1000

Free FIPS Key Memory : 1000

Total SRAM Memory : 557396

Free SRAM Memory : 238088

Total Crypto Cores : 4

Enabled Crypto Cores : 4

 Done

>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating a FIPS Key for an Instance on an SDX 14030/14060/14080 FIPS Appliance

Feb 16, 2017

You can create a FIPS key on your instance or import an existing FIPS key into the instance. An SDX 14030/14060/14080 FIPS appliance supports only 2048-bit and 3072-bit keys and an exponent value of F4. For PEM keys, an exponent is not required. Verify that the FIPS key is created correctly. Create a certificate signing request and a server certificate. Finally, add the certificate-key pair to your instance.

> **Note**
>
> 1024-bit and 4096-bit keys and an exponent value of 3 are not supported.

To create a FIPS key by using the NetScaler command line

At the command prompt, type:

create ssl fipsKey <fipsKeyName> -modulus <positive_integer> [-exponent F4]

Example COPY

```
> create fipsKey  f1 -modulus 2048 -exponent F4

Done
```

To import a FIPS key by using the NetScaler command line

At the command prompt, type:

import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-wrapKeyName <string>] [-iv<string>] [-exponent F4]

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4

Done

>import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM

Done
```

Verify that the FIPS key is created or imported correctly by running the **show fipskey** command.

Example COPY

```
> show fipskey

1)     FIPS Key Name: Key-FIPS-2

Done
```

To create a certificate signing request by using the NetScaler command line

At the command prompt, type:

create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName <string>) [-keyform (DER | PEM) {-PEMPassPhrase }] -countryName <string> -stateName <string> -organizationName<string> [-organizationUnitName <string>] [-localityName <string>] [-commonName <string>] [-emailAddress <string>] {-challengePassword } [-companyName <string>] [-digestMethod (SHA1 | SHA256)]

Example COPY

```
>create certreq f1.req –fipsKeyName f1 -countryName US -stateName CA -organizationName Citrix -companyName Citrix -commonNam

Done
```

To create a server certificate by using the NetScaler command line

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

create ssl cert <certFile> <reqFile> <certType> [-keyFile <input_filename>] [-keyform (DER | PEM) {-PEMPassPhrase }] [-days <positive_integer>] [-certForm (DER | PEM)] [-CAcert <input_filename>] [-CAcertForm (DER | PEM)] [-CAkey <input_filename>] [-CAkeyForm (DER | PEM)] [-CAserial <output_filename>]

Example COPY

```
> create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-root.key -CAserial ns-root.srl -days 1000

Done
```

The above example creates a server certificate using a local root CA on the appliance.

To add a certificate-key pair by using the NetScaler command line

At the command prompt, type:

add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>] [-expiryMonitor (ENABLED | DISABLED) [-notificationPeriod <positive_integer>]] [-bundle (YES | NO)]

Example COPY

```
> add certkey cert1 -cert f1.cert -fipsKey f1

Done
```

After creating the FIPS key and server certificate, you can add the generic SSL configuration. Enable the features that are required for your deployment. Add servers, services, and SSL virtual servers. Bind the certificate-key pair and the service to the SSL virtual server, and save the configuration.

Example COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
>enable ns feature SSL LB

Done

>add server s1 10.217.2.5

Done

>add service sr1 s1 HTTP 80

Done

>add lb vserver v1 SSL 10.217.2.172 443

Done

>bind ssl vserver v1 –certkeyName cert1

Done

>bind lb vserver v1 sr1

Done

> saveconfig

Done
```

For information about configuring secure HTTPS and secure RPC, see http://docs.citrix.com/en-us/netscaler/11/getting-started-with-netscaler/configure-fips-first-time.html.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Upgrading the FIPS Firmware on a VPX Instance

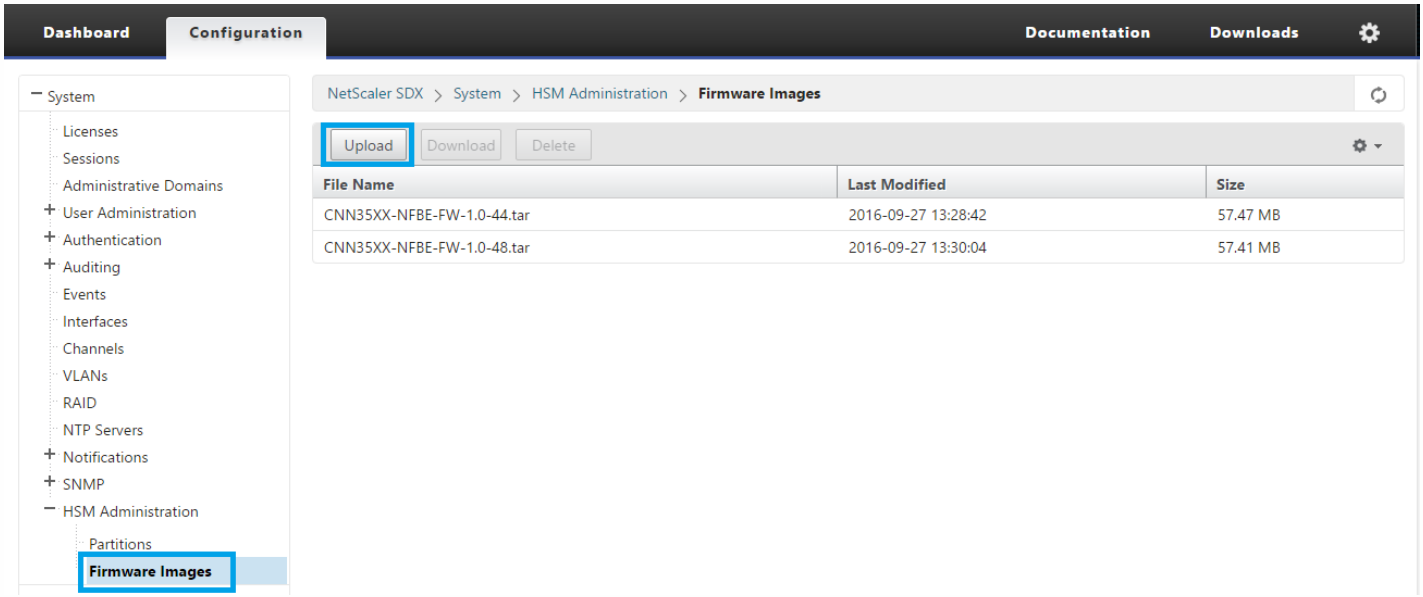
Oct 05, 2016

FIPS firmware updates are released from time to time. Download the latest firmware from the Citrix download page and upload it to the appliance. The upgrade process might take up to 10 minutes to complete. The instance is restarted after the upgrade.

To upgrade the FIPS firmware on a VPX instance

1. Navigate to **System** > **HSM Administration** > **Firmware Images**.

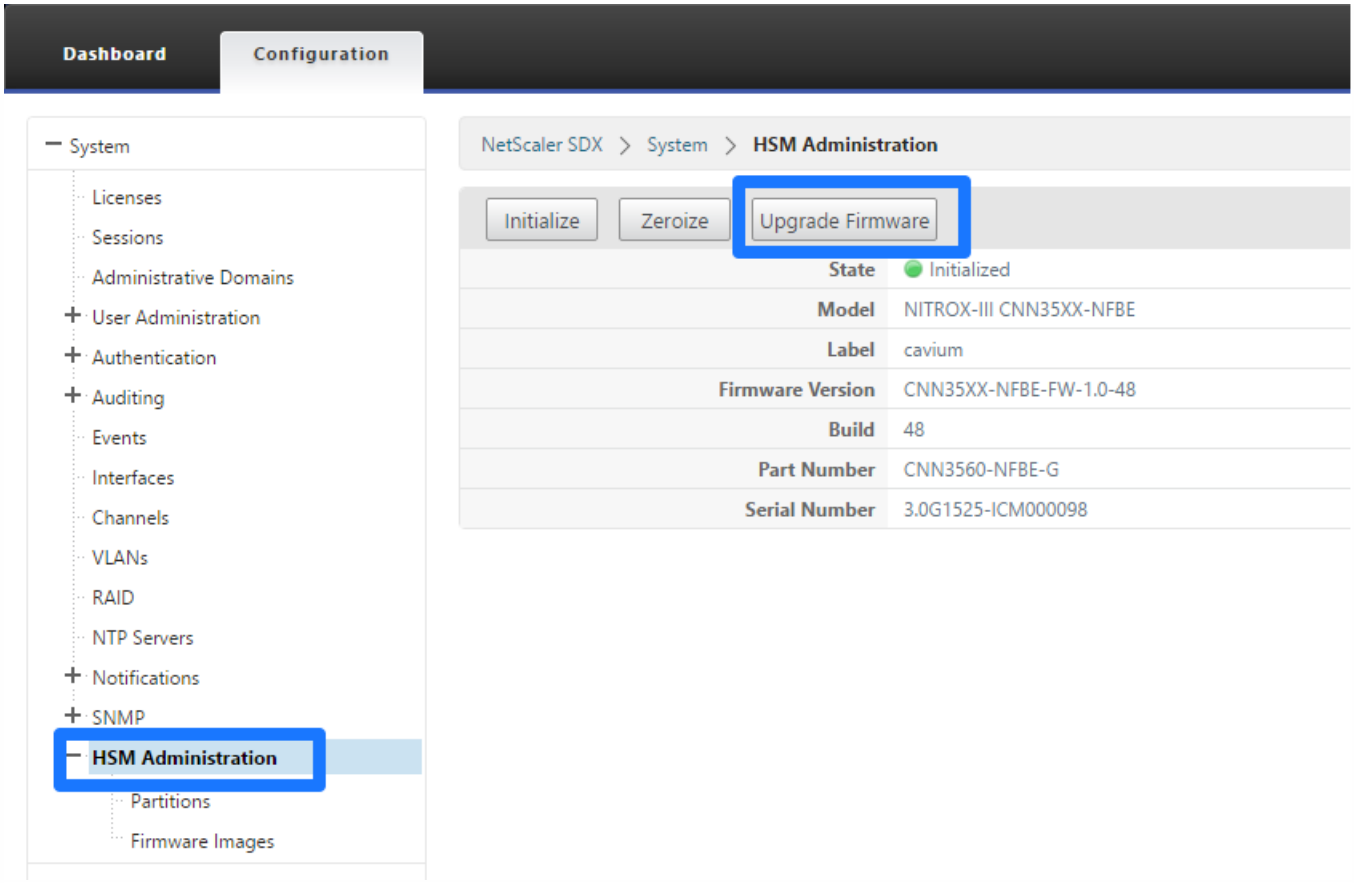
2. Select **Upload**.



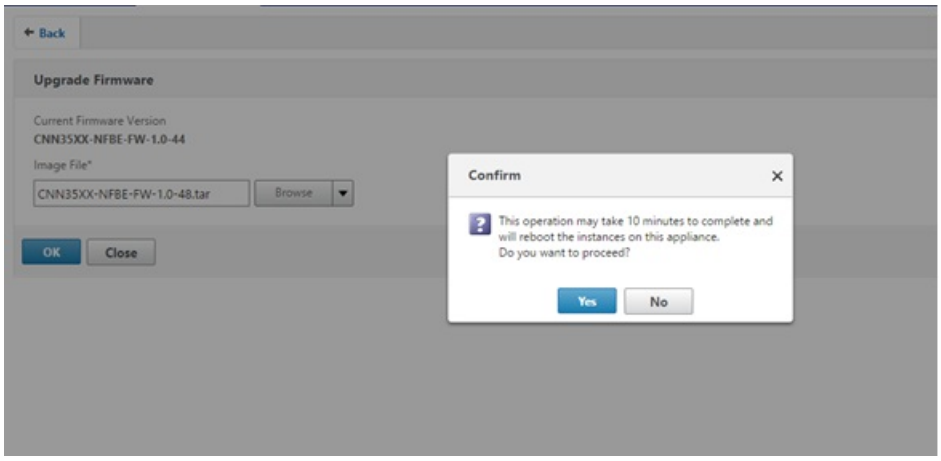
3. Navigate to the folder that contains the firmware image and select the file.

4. Navigate to **System** > **HSM Administration**, and select **Upgrade Firmware**.

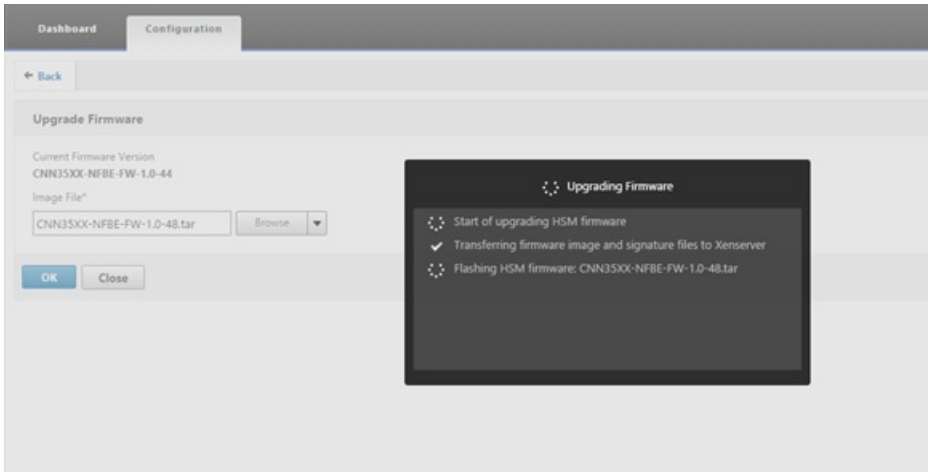
© 1999-2017 Citrix Systems, Inc. All rights reserved.



5. Select the firmware image to upgrade to, and click **OK**.



© 1999-2017 Citrix Systems, Inc. All rights reserved.



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Optimization

May 13, 2016

The NetScaler optimization features reduce transaction times between the clients and the servers, and they reduce bandwidth consumption. They also enhance server performance by offloading some tasks and making others more efficient.

Client Keep-Alive	Handles multiple requests on a single client connection. The client does not have to negotiate a new connection for each request to the server.
HTTP Compression	Compresses HTTP responses sent from the servers to compression-aware browsers. The smaller responses reduce download time and save bandwidth.
Integrated Caching	Stores responses to client requests. Subsequent requests for the same content are served from the NetScaler cache instead of being forwarded to the origin server.
Front End Optimization	Reduces the load and render time of web pages by simplifying and optimizing the content served to the client browser. Note: Supported from NetScaler 10.5 onwards.
Content Accelerator	Stores server responses on a Citrix ByteMobile T2100 appliance. Note: Supported from NetScaler 10.1 onwards.
SPDY (Speedy)	Acts as a SPDY gateway between clients and your servers, providing SPDY support without the need to configure/upgrade SPDY on the servers. Note: Supported from NetScaler 10.1 onwards.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Client Keep-Alive

Aug 06, 2014

The client keep-alive feature enables multiple client requests to be sent on a single client connection. This feature helps in a transaction management environment where typically the server closes the client connection after serving the response. The client then opens a new connection for each request and spends more time on the transaction.

Client keep-alive resolves this issue by keeping the connection between the client and the appliance (client-side connection) open even after the server closes the connection with the appliance. This allows sending multiple client requests using a single connection and saves the round trips associated with opening and closing a connection. Client keep-alive is most beneficial in SSL sessions.

Client keep-alive is also useful under either of the following conditions:

- When the server does not support client keep-alive.
- When the server supports client keep-alive but an application on the server does not support client keep-alive.

Note: Client keep-alive is applicable for HTTP and SSL traffic.
Client-keep alive can be configured globally to be able to handle all traffic. It can also be configured to be active only on specific services.

In client keep-alive environment, the configured services intercept the client traffic and the client request is directed to the origin server. The server sends the response and closes the connection between the server and the appliance. If a "Connection: Close" header is present in the server response, the appliance corrupts this header in the client-side response, and the client-side connection is kept open. As a result, the client does not have to open a new connection for the next request; instead, the connection to the server is reopened.

Note: If a server sends back two "Connection: Close" headers, only one is edited. This results in significant delays on the client rendering of the object because a client does not assume that the object has been delivered completely until the connection is actually closed.

Configuring Client Keep-Alive

Updated: 2014-08-12

Client keep-alive, by default, is disabled on the NetScaler, both globally and at service level. Therefore, you must enable the feature at the required scope.

Note: If you enable client keep-alive globally, it is enabled for all services, regardless of whether you enable it at the service level.
Additionally, if required, you can configure some HTTP parameters to specify the maximum number of HTTP connections retained in the connection reuse pool, enable connection multiplexing, and enable persistence Etag.

Note: When Persistent ETag is enabled, the ETag header includes information about the server that served the content. This ensures that cache validation conditional requests or browser requests, for that content, always reaches the same server.

To configure client keep-alive by using the command line interface

At the command prompt, do the following:

1. Enable client keep-alive on the NetScaler.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- At global level
  enable ns mode cka

- At service level
  set service <name> -CKA YES

  Note: Client keep-alive can be enabled only for HTTP and SSL services.

2. Configure the required HTTP parameters on the HTTP profile that is bound to the service(s).
   set ns httpProfile <name> -maxReusePool <value> -conMultiplex ENABLED -persistentETag ENABLED

   Note: Configure these parameters on the nshttp_default_profile HTTP profile, to make them available globally.

To configure client keep-alive by using the configuration utility

1. Enable client keep-alive on the NetScaler.
   - At global level
     Navigate to System > Settings, click Configure Modes and select Client side Keep Alive.

   - At service level
     Navigate to Traffic Management > Load Balancing > Services, and select the required service. In the Settings grouping, enable Client Keep-Alive.

2. Configure the required HTTP parameters on the HTTP profile that is bound to the service(s).
   Navigate to System > Profiles, and on HTTP Profiles tab, select the required profile and update the required HTTP parameters.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Compression

May 20, 2015

For websites with compressible content, the NetScaler HTTP compression feature implements lossless compression to alleviate latency, long download times, and other network-performance problems by compressing the HTTP responses sent from servers to compression-aware browsers. You can improve server performance by offloading the computationally intensive compression task from your servers to the NetScaler appliance.

The following table describes the capabilities of the HTTP compression feature:

Functionality	Description
Compression Ratio	Compression ratio depends on the types of files in the responses, but is always significant, noticeably reducing amount of data transmitted over the network.
Browser Awareness	NetScaler serves compressed data to compression aware browsers only, reducing the transaction time between the client and the server. Most modern web browsers support HTTP compression.
Compression blocking	You can define content filters to selectively block compression by applying built-in actions.
Compression Caching	With the integrated caching feature enabled, subsequent requests for the same content are served from the local cache, reducing the number of round trips to the server and improving transaction times.
HTTPS Support	Compression is particularly useful on SSL connections, because it reduces the amount of content that has to be encrypted, either on the server or by the NetScaler appliance, and decrypted by the client.
Intelligent Response Filtering	The NetScaler compression engine intelligently filters server responses on the basis of defined compression parameters. For example, the compression engine detects zero-content-length responses and compressed responses and does not compress them. The detection of compressed responses enables origin sites to use server-based compression in conjunction with the NetScaler compression feature.
Compression Switching	The NetScaler appliance transparently directs requests from compression aware clients to compression capable servers, so that responses to those clients are compressed, and responses to other clients are not delayed by compression processing.

How Compression Works

A NetScaler ADC can compress both static and dynamically generated data. It applies the GZIP or the DEFLATE compression algorithm to remove extraneous and repetitive information from the server responses and represent the original information in a more compact and efficient format. This compressed data is sent to the client's browser and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

uncompressed as determined by the browser's supported algorithm or algorithms (GZIP or DEFLATE).

NetScaler compression treats static and dynamic content differently.

- Static files are compressed only once, and a compressed copy is stored in local memory. Subsequent client requests for cached files are serviced from that memory.
- Dynamic pages are dynamically created each time a client requests them.

When a client sends a request to the server:

1. The client request arrives at the NetScaler ADC. The ADC examines the headers and stores information about what kind of compression, if any, the browser supports.
2. The ADC forwards the request to the server and receives the response.
3. The NetScaler compression engine examines the server response for compressibility by matching it against policies.
4. If the response matches a policy associated with a compression action, and the client browser supports a compression algorithm specified by the action, the NetScaler ADC applies the algorithm and sends the compressed response to the client browser.
5. The client applies the supported compression algorithm to decompress the response.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring HTTP Compression

Nov 07, 2017

By default, compression is disabled on the NetScaler ADC. You must enable the feature before configuring it. If the feature is enabled, the ADC compresses server requests specified by compression policies.

To configure HTTP compression, do the following:

- Enable HTTP Compression
- Configure compression actions
- Configure compression policies
- Bind the compression policies to global bind points or to virtual servers
- Optionally, configure global compression parameters

Enabling HTTP Compression

Compression can be enabled for HTTP and SSL services only. You can enable it globally, so that it applies to all HTTP and SSL services, or you can enable it just for specific services.

To enable compression by using the command line interface

At the command prompt, enter one of the following commands to enable compression globally or for a specific service:

- enable ns feature cmp
  OR
- set service <name> -CMP YES

To configure compression by using the configuration utility

Do one of the following:

- To enable compression globally, navigate to **System** > **Settings**, click **Configure Basic Features**, and select **HTTP Compression**.
- To enable compression for a specific service, navigate to **Traffic Management** > **Load Balancing** > **Services**, select the service, and click **Edit**. In the **Settings** group, click the pencil icon and enable **Compression**.

Configuring a Compression Action

A compression action specifies the action to take when a request or response matches the rule (expression) in the policy with which the action is associated. For example, you can configure a compression policy that identifies requests that will be sent to a particular server, and associate the policy with an action that compresses the server's response.

There are four built-in compression actions:

- COMPRESS: Uses the GZIP algorithm to compress data from browsers that support either GZIP or both GZIP and DEFLATE. Uses the DEFLATE algorithm to compress data from browsers that support only the DEFLATE algorithm. If the browser does not support either algorithm, the browser's response is not compressed.
- NOCOMPRESS: Does not compress data.
- GZIP: Uses the GZIP algorithm to compress data for browsers that support GZIP compression. If the browser does not support the GZIP algorithm, the browser's response is not compressed.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- DEFLATE: Uses the DEFLATE algorithm to compress data for browsers that support the DEFLATE algorithm. If the browser does not support the DEFLATE algorithm, the browser's response is not compressed. After creating an action, you associate the action with one or more compression policies.

To create a compression action by using the command line interface

At the command prompt, enter the following command to create a compression action:

add cmp action <name> <cmpType> [-addVaryHeader <addVaryHeader> -varyHeaderValue <string>]

To create a compression action by using the configuration utility

Navigate to **Optimization** > **HTTP Compression** > **Actions** , click **Add** , and create a compression action to specify the type of compression to be performed on the HTTP response.

Configuring a Compression Policy

A compression policy contains a rule, which is a logical expression that enables the NetScaler appliance to identify the traffic that should be compressed.

When the NetScaler ADC receives an HTTP response from a server, it evaluates the built-in compression policies and any custom compression policies to determine whether to compress the response and, if so, the type of compression to apply. Priorities assigned to the policies determine the order in which the policies are matched against the requests.

The following table lists the built-in HTTP compression policies. These policies are activated globally when you enable compression.

Built-in Classic or Default Syntax Policy	**Description**
ns_nocmp_mozilla_47 ns_adv_nocmp_mozilla_47	Prevents compression of CSS files when a request is sent from a Mozilla 4.7 browser.
ns_cmp_mscss ns_adv_cmp_mscss	Compresses CSS files when the request is sent from a Microsoft Internet Explorer browser.
ns_cmp_msapp ns_adv_cmp_msapp	Compresses files that are generated by the following applications:s • Microsoft Office Word • Microsoft Office Excel • Microsoft Office PowerPoint
ns_cmp_content_type ns_adv_cmp_content_type	Compresses data when the response containsContent-Type header and contains text.
ns_nocmp_xml_ie	Prevents compression when a request is sent, from a Microsoft Internet Explorer

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Built-in Classic or Default Syntax Policy	Description
ns_adv_nocmp_xml_ie	browser and the response contains a Content-Type header and contains text or xml.

To create a compression policy by using the command line interface

At the command prompt, enter the following command to create a compression policy:

add cmp policy <name> -rule <expression> -resAction <string>

To create a compression policy by using the configuration utility

Navigate to **Optimization** > **HTTP Compression** > **Policies** , click **Add** , and create a compression policy by specifying the condition and the corresponding action to be executed.

Binding a Compression Policy

To put a compression policy into effect, you must bind it either globally, so that it applies to all traffic that flows through the NetScaler ADC, or to a specific virtual server, so that the policy applies only to requests whose destination is the VIP address of that virtual server.

When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

To bind a compression policy by using the command line interface

At the command prompt, enter one of the following commands to bind a compression policy globally or to a specific virtual server:

- bind cmp global <policyName> [-priority <positive_integer>] [-state (ENABLED | DISABLED)]...
- bind lb vserver <vserverName> -policyName <policyName> -priority <positive_integer>. Repeat this command for each virtual server to which you want to bind the compression policy.

To bind a compression policy by using the configuration utility

Do one of the following:

- At global level Navigate to **Optimization** > **HTTP Compression** > **Policies**, click **Policy Manager** and bind the required policies by specifying the relevant **Bind Point** and **Connection Type** (Request/Response).
- At virtual server level
  - For load balancing virtual server, Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, select the required virtual server, click **Policies**, and bind the relevant policy.
  - For content switching **virtual server**, Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers**, select the required virtual server, click **Policies**, and bind the relevant policy.

Setting the Global Compression Parameters for Optimal Performance

Many users accept the default values for the global compression parameters, but you might be able provide more effective compression by customizing these settings.

Note: After you configure the global compression parameters, you do not have to reboot your appliance. They get applied to the new flows immediately.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The following table describes the compression parameters that you can set on the NetScaler ADC.

Compression Parameters	Description
Quantum size	Size, in KB, of the buffer maintained for accumulating server responses. The responses are compressed when the buffer size exceeds this value. For example, if you set the quantum size to 50 KB, the NetScaler ADC compresses the buffer's contents when its size becomes larger than 50 KB. Minimum value: 1. Maximum value: 63488. Default: 57344.
Compression level	Level of compression to apply to server responses. Possible values: Best Speed, Best Compression, optimal.
Minimum HTTP response size	Minimum size, in bytes, of an HTTP response that is compressed. Responses smaller than the value specified by this parameter are sent without being compressed.
Bypass compression on CPU usage	NetScaler CPU usage, as a percentage, at or above which no compression is done. Default: 100.
Policy Type*	Type of policies used for compression. Possible values: Classic, Default Syntax. Default: Classic.
Allow Server-side compression	Allow servers to send compressed data to the NetScaler ADC.
Compress push packet	Upon receipt of a packet with a TCP PUSH flag, compress the accumulated packets immediately, without waiting for the quantum buffer to be filled.
External Cache	Issue a private response directive indicating that the response message is intended for a single user and must not be cached by a shared or proxy cache.

To configure global compression parameters by using the command line interface

At the command prompt, enter the following command to configure compression parameters that apply globally:

set cmp parameter -cmpLevel <cmpLevel> -quantumSize <integer> [-addVaryHeader (ENABLED | DISABLED) [-varyHeaderValue <string>]]...

Note: Vary header parameters are available from NetScaler 10.5 onwards.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure global compression parameters by using the configuration utility

Navigate to **Optimization** > **HTTP Compression** , click **Change Compression Settings** , and set the relevant parameters.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Evaluating Your Compression Configuration

Apr 20, 2015

You can view the compression statistics in the dashboard utility or in an SNMP monitor. The dashboard utility displays summary and detailed statistics in a tabular and graphic format.

Optionally, you can also view statistics for a compression policy, including the number of hits that the policy counter increments during the policy based compression.

Note:

- For more information about the statistics and charts, see the Dashboard help on the Citrix NetScaler appliance.
- For more information about SNMP, see SNMP.

To View Compression Statistics by Using the Command Line Interface

At the command prompt, enter the following commands to display the compression statistics:

1. To display compression statistics summary
   stat cmp
   Note: The stat cmp policy command displays statistics for default syntax compression policies only.
2. To display compression policy hits and details
   show cmp policy <name>
3. To display detailed compression statistics
   stat cmp -detail

To View Compression Statistics by Using the Dashboard

In the Dashboard utility, you can display the following types of compression statistics:

- Select Compression to display a summary of the compression statistics.
- To display detailed compression statistics by protocol type, click the Details
- To display the rate of requests processed by the compression feature, click the Graphical View tab.

To View Compression Statistics by Using SNMP

You can view the following compression statistics by using the SNMP network management application.

- Number of compression requests (OID: 1.3.6.1.4.1.5951.4.1.1.50.1)
- Number of compressed bytes transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.2)
- Number of compressible bytes received (OID: 1.3.6.1.4.1.5951.4.1.1.50.3)
- Number of compressible packets transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.4)
- Number of compressible packets received (OID: 1.3.6.1.4.1.5951.4.1.1.50.5)
- Ratio of compressible data received and compressed data transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.6)
- Ratio of total data received to total data transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.7)

To View Additional Compression Statistics by Using the Configuration Utility

1. To display HTTP compression statistics:
   Navigate toOptimization > HTTP Compression and click Statistics.
2. To display statistics of a compression policy
   Navigate to Optimization > HTTP Compression > Policies> select the policy, and click Statistics.
3. To display statistics of a compression policy label

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Navigate to Optimization > HTTP Compression > Policies> select a policy label, and click Statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Offloading HTTP Compression to the NetScaler ADC

Nov 04, 2015

Performing compression on a server can affect the server's performance. A NetScaler ADC placed in front of your web servers and configured for HTTP compression offloads compression of both static and dynamic content, saving server CPU cycles and resources.

You can offload compression from the Web servers in either of two ways:

- Disable compression on the web servers, enable the NetScaler Compression feature at a global level, and configure services for compression.
- Leave the compression feature enabled on the web servers and configure the NetScaler appliance to remove the "Accept Encoding" header from all HTTP client requests. The servers then send uncompressed responses. The NetScaler ADC compresses the server responses before sending them to the clients.

Note: The second option does not work if the servers automatically compress all responses. The NetScaler ADC does not attempt to compress a response that is already compressed.

The Servercmp parameter enables the Netscaler appliance to handle offload HTTP compression. By default, this parameter is set ON for the server to send compressed data to the Netscaler appliance. To offload HTTP compression, you need to set the servercmp parameter to OFF. At the command prompt, enter the following commands:

set service <service name> -CMP YES

Repeat this command for each service for which you want to enable compression.

show service <service name>

Repeat this command for each service, to verify that compression is enabled.

Save config

set cmp parameter –serverCmp OFF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Integrated Caching

Oct 03, 2017

The integrated cache provides in-memory storage on the Citrix NetScaler appliance and serves Web content to users without requiring a round trip to an origin server. For static content, the integrated cache requires little initial setup. After you enable the integrated cache feature and perform basic setup (for example, determining the amount of NetScaler appliance memory the cache is permitted to use), the integrated cache uses built-in policies to store and serve specific types of static content, including simple Web pages and image files. You can also configure the integrated cache to store and serve dynamic content that is usually marked as non-cacheable by Web and application servers (for example, database records and stock quotes).

> **Note**
>
> The term Integrated Cache can be interchangeably used with AppCache; note that from a functionality point of view, both terms mean the same.

When a request or response matches the rule (logical expression) specified in a built-in policy or a policy that you have created, the NetScaler appliance performs the action associated with the policy. By default, all policies store cached objects in and retrieve them from the Default content group, but you can create your own content groups for different types of content.

To enable the NetScaler appliance to find cached objects in a content group, you can configure selectors, which match cached objects against expressions, or you can specify parameters for finding objects in the content group. If you use selectors (which Citrix recommends), configure them first, so that you can specify selectors when you configure content groups. Next, set up any content groups that you want to add, so that they are available when you configure the policies. To complete the initial configuration, create policy banks by binding each policy to a global bind point or a virtual server, or to a label that can be called from other policy banks.

You can tune the performance of the integrated cache, using methods such as pre-loading cached objects before they are scheduled to expire. To manage the handling of cached data once it leaves the NetScaler appliance, you can configure caching-related headers that are inserted into responses. The integrated cache can also act as a forward proxy for other cache servers.

Note: Integrated caching requires some familiarity with HTTP requests and responses. For information about the structure of HTTP data, see *Live HTTP Headers* at "http://livehttpheaders.mozdev.org/."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How the Integrated Cache Works

Sep 04, 2015

The integrated cache monitors HTTP and SQL requests that flow through the Citrix NetScaler appliance and compares the requests with stored policies. Depending on the outcome, the integrated cache feature either searches the cache for the response or forwards the request to the origin server. For HTTP requests, the integrated cache feature can also serve partial content from the cache in response to single byte-range and multi-part byte-range requests.

Cached data can be compressed if the client accepts compressed content. You can configure expiration times for a content group, and you can selectively expire entries in a content group.

Data that is served from the integrated cache is a cache hit, and data served from the origin is a cache miss, as described in the following table.

Table 1. Cache Hits and Misses

Transaction Type	Specifies
Cache Hit	Responses that the NetScaler appliance serves from the cache, including: ● Static objects, for example, image files and static Web pages ● 200 OK pages ● 203 Non-Authoritative Response pages ● 300 Multiple Choices pages ● 301 Moved Permanently pages ● 302 Found pages ● 304 Not Modified pages These responses are known as positive responses. The NetScaler appliance also caches the following negative responses: ● 307 Temporary Redirect pages ● 403 Forbidden pages ● 404 Not Found pages ● 410 Gone pages To further improve performance, you can configure the NetScaler appliance to cache additional types of content.
Storable Cache Miss	For a storable cache miss, the NetScaler appliance fetches the response from the origin server, and stores the response in the cache before serving it to the client.
Non-Storable Cache Miss	A non-storable cache miss is inappropriate for caching. By default, any response that contains the following status codes is a non-storable cache miss:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Transaction Type	**Specifies** • 201, 202, 204, 205, 206 status codes • All 4xx codes, except 403, 404 and 410 • 5xx status codes

Note: To integrate dynamic caching with your application infrastructure, use the NITRO API to issue cache commands remotely. For example, you can configure triggers that expire cached responses when a database table is updated.
To ensure the synchronization of cached responses with the data on the origin server, you configure expiration methods. When the NetScaler appliance receives a request that matches an expired response, it refreshes the response from the origin server.

Note: Citrix recommends that you synchronize the times on the NetScaler appliance and the back-end server(s).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example of Dynamic Caching

Sep 11, 2014

Dynamic caching evaluates HTTP requests and responses based on parameter-value pairs, strings, string patterns, or other data. For example, suppose that a user searches for Bug 31231 in a bug reporting application. The browser sends the following request on the user's behalf:

```
GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&Template=view&TableId=1000
Host: mycompany.net
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9) Gecko/2008052906 Firefox/3.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
. . .
```

In this example, GET requests for this bug reporting application always contain the following parameters:

- IssuePage
- RecordID
- Template
- TableId

GET requests do not update or alter the data, so you can configure these parameters in caching policies and selectors, as follows:

- You configure a caching policy that looks for the string mybugreportingsystem and the GET method in HTTP requests. This policy directs matching requests to a content group for bugs.
- In the content group for bugs, you configure a hit selector that matches various parameter-value pairs, including IssuePage, RecordID, and so on.

Note that a browser can send multiple GET requests based on one user action. The following is a series of three separate GET requests that a browser issues when a user searches for a bug based on a bug ID.

```
GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&Template=view&TableId=1000
GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=viewbtns&RecordId=31231&TableId=1000
GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=viewbody&RecordId=31231&tableid=1000
```

To fulfill these requests, multiple responses are sent to the user's browser, and the Web page that the user sees is an assembly of the responses.

If a user updates a bug report, the corresponding responses in the cache should be refreshed with data from the origin server. The bug reporting application issues HTTP POST requests when a user updates a bug report. In this example, you configure the following to ensure that POST requests trigger invalidation in the cache:

- A request-time invalidation policy that looks for the string mybugreportingsystem and the POST HTTP request method, and directs matching requests to the content group for bug reports.
- An invalidation selector for the content group for bug reports that expires cached content based on the RecordID parameter. This parameter appears in all of the responses, so the invalidation selector can expire all relevant items in the cache.

The following excerpt shows a POST request that updates the sample bug report.

```
POST /mybugreportingsystem/mybugreport.dll?TransitionForm HTTP/1.1\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) Opera 7.23 [en]\r\n
Host: mybugreportingsystem\r\n
Cookie:ttSearch.134=%23options%3Afalse%23active%23owner%3Afalse%23unowned%3Afalse%23submitter%3Afalse%23incsub%3Atrue;
Cookie2: $Version=1\r\n
. . .
\r\n
ProjectId=2&RecordId=31231&TableId=1000&TransitionId=1&Action=Update&CopyProjectId=0&ReloadForm=0&State=&RecordLockId=49873+issues+in+HTTP&F43. . .
```

When the Citrix NetScaler appliance receives this request, it does the following:

- Matches the request with an invalidation policy.
- Finds the content group that is named in the policy.
- Applies the invalidation selector for this content group and expires all responses that match RecordID=31231.

When a user issues a new request for this bug report, the NetScaler appliance goes to the origin server for updated copies of all the responses that are associated with the report instance, stores the responses in the content group, and serves them to the user's browser, which reassembles the report and displays it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting Up the Integrated Cache

Dec 04, 2017

To use the integrated cache, you must install the license and enable the feature. After you enable the integrated cache, the Citrix® NetScaler® appliance automatically caches static objects as specified by built-in policies and generates statistics on cache behavior. (Built-in policies have an underscore in the initial position of the policy name.)

Even if the built-in policies are adequate for your situation, you might want to modify the global attributes. For example, you might want to modify the amount of NetScaler appliance memory allocated to the integrated cache.

If you would like to observe cache operation before changing settings, see "Displaying Cached Objects and Cache Statistics."

Note: The NetScaler cache is an in-memory store that is purged when you restart the appliance.
This section includes the following details:

- Installing the Integrated Cache License
- Enabling Integrated Caching
- Configuring Global Attributes for Caching
- Built-in Content Group, Pattern Set, and Policies for the Integrated Cache

Installing the Integrated Cache License

Updated: 2013-10-28

An integrated cache license is required. For information about licenses, see information about obtaining NetScaler licenses at " http://support.citrix.com/article/ctx121062."

To install the license for the Integrated Caching feature

1. Obtain a license code from Citrix, go to the command line interface, and log in.
2. At the command line interface, copy the license file to the /nsconfig/license folder.
3. Reboot the NetScaler appliance by using the following command:

   reboot

Enabling Integrated Caching

Updated: 2015-05-20

When you enable integrated caching, the NetScaler appliance begins caching server responses. If you have not configured any policies or content groups, the built in policies store cached objects in the Default content group.

To enable integrated caching by using the command line interface

At the command prompt, type one of the following commands to enable or disable integrated caching:

enable ns feature IC

To enable integrated caching by using the configuration utility

Navigate to System > Settings, click Configure Basic Features, and select Integrated Caching.

Configuring Global Attributes for Caching

Updated: 2014-08-08

Global attributes apply to all cached data. You can specify the amount of NetScaler memory allocated to the integrated cache, Via header insertion, a criterion for verifying that a cached object should be served, the maximum length of a POST body permitted in the cache, whether to bypass policy evaluation for HTTP GET requests, and an action to take when a policy cannot be evaluated.

The cache memory capacity is limited only by the memory of the hardware appliance. Also, any packet engine (the central distribution hub of all incoming TCP requests) in the nCore NetScaler appliance is aware of objects cached by other packet engines in the nCore NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: When the default global memory limit is set as 0 and the Integrated Caching (IC) feature is enabled, the appliance does not cache any objects. For caching, you must explicitly configure the global memory limit. However, if you enable "set aaa parameter enableStaticPageCaching" option, there will be some default memory configured in the appliance. This memory is insufficient for caching large Objects and so it is necessary to assign a higher memory limit for IC. You can perform this by configuring the "set cache parameter –memLimit" command. The new setting is applied only after you save the configuration and reboot the appliance.

You can modify the global memory limit configured for caching objects. However, when you update the global memory limit to a value lower than the existing value (for example, from 10 GB to 4 GB), if a higher amount of memory (greater than 4 GB) is already being used to cache objects, the NetScaler continues using that amount of memory.

This means that even though the integrated caching limit is configured to some value, the actual limit used can be higher. This excessive memory is however released when the objects are removed from cache.

The output of the show cache parameter command indicates the configured value (Memory Usage limit) and the actual value being used (Memory usage limit (active value)).

To configure global settings for caching by using the command line interface

At the command prompt, type:

set cache parameter [-memLimit <MBytes>] [-via <string>] [-verifyUsing <criterion>] [-maxPostLen <positiveInteger>] [-prefetchMaxPending <positiveInteger>] [-enableBypass (YES | NO)] [-undefAction (NOCACHE | RESET)]

To configure global settings for caching by using the configuration utility

Navigate to Optimization > Integrated Caching, click Change Cache Settings, and configure the global settings for caching.

Built-in Content Group, Pattern Set, and Policies for the Integrated Cache

Updated: 2013-08-23

The Citrix NetScaler appliance includes a built-in integrated caching configuration that you can use for caching content. The configuration consists of a content group called ctx_cg_poc, a pattern set called ctx_file_extensions, and a set of integrated cache policies. In the content group ctx_cg_poc, only objects that are 500 KB or smaller are cached. The content is cached for 86000 seconds, and the memory limit for the content group is 512 MB. The pattern set is an indexed array of common file extensions for file-type matching.

The following table lists the built-in integrated caching policies. By default, the policies are not bound to any bind point. You must bind them to a bind point if you want the NetScaler appliance to evaluate traffic against the policies. The policies cache objects in the ctx_cg_poc content group.

Table 1. Built-in Integrated Caching Policies

Integrated caching policy name	Policy rule	Policy action	Description
_cacheVPNStaticObjects	HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).CONTAINS_INDEX(\"ctx_file_extensions\").BETWEEN(101,150)	CACHE	Check whether the requested file is an image file (by comparing the extension of the requested object with the extensions in ctx_file_extensions, indexes 101 through 150) and then cache the file.
_cacheTCPVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(\".css\")	CACHE	Cache all cascading style sheet content.
_cacheOCVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(\".pdf\")	CACHE	Cache all PDF content.
_cacheWFStaticObjects	HTTP.REQ.URL.ENDSWITH(\".js\")	CACHE	Cache all JSS content.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Integrated caching policy name	Policy rule	Policy action	Description
_mayNoCacheReq	HTTP.RES.HEADER(\"Content-Type\").CONTAINS(\"application/x-javascript\")	CACHE	Cache all JavaScript content.
_noCacheRest	TRUE	NOCACHE	Do not cache any content.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Selectors and Basic Content Groups

Aug 26, 2013

You can configure selectors and apply them to content groups. When you add a selector to one or more content groups, you specify whether the selector is to be used for identifying cache hits or identifying cached objects to be invalidated (expired). Selectors are optional. Alternatively, you can configure content groups to use hit parameters and invalidation parameters. However, Citrix recommends that you configure selectors.

After configuring selectors, or deciding to use parameters instead, you are ready to set up a basic content group. After creating the basic content group, you need to decide how objects should be expired from the cache, and configure cache expiration. You can further modify the cache as described in "Improving Cache Performance" and "Configuring Cookies, Headers, and Polling", but you might first want to configure caching policies.

Note: Content group parameters and selectors are used only at request time, and you typically associate them with policies that use MAY_CACHE or MAY_NOCACHE actions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Advantages of Selectors

Oct 28, 2013

A selector is a filter that locates particular objects in a content group. If you do not configure a selector, the Citrix® NetScaler® appliance looks for an exact match in the content group. This can lead to multiple copies of the same object residing in a content group. For example, a content group that does not have a selector may need to store URLs for host1.domain.com\mypage.htm, host2.domain.com\mypage.htm, and host3.domain.com\mypage.htm. In contrast, a selector can match just the URL (mypage.html, using the expression http.req.url) and the domain (.com, using the expression http.req.hostname.domain), allowing the requests to be satisfied by the same URL.

Selector expressions can perform simple matching of parameters (for example, to find objects that match a few query string parameters and their values). A selector expression can use Boolean logic, arithmetic operations, and combinations of attributes to identify objects (for example, segments of a URL stem, a query string, a string in a POST request body, a string in an HTTP header, a cookie). Selectors can also perform programmatic functions to analyze information in a request. For example, a selector can extract text in a POST body, convert the text into a list, and extract a specific item from the list.

For more information about expressions and what you can specify in an expression, see "Policies and Expressions."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using Parameters Instead of Selectors

Jul 10, 2013

Although Citrix recommends the use of selectors with a content group, you can instead configure hit parameters and invalidation parameters. For example, suppose that you configure three hit parameters in a content group for bug reports: BugID, Issuer, and Assignee. If a request contains BugID=456, with Issuer=RohitV and Assignee=Robert, the NetScaler appliance can serve responses that match these parameter-value pairs.

Invalidation parameters in a content group expire cached entries. For example, suppose that BugID is ab invalidation parameter and a user issues a POST request to update a bug report. An invalidation policy directs the request to this content group, and the invalidation parameter for the content group expires all cached responses that match the BugID value. (The next time a user issues a GET request for this report, a caching policy can enable the NetScaler appliance to refresh the cached entry for the report from the origin server.)

Note that the same parameter can be used as a hit parameter or an invalidation parameter.

Content groups extract request parameters in the following order:

- URL query
- POST body
- Cookie header

After the first occurrence of a parameter, regardless of where it occurred in the request, all its subsequent occurrences are ignored. For example, if a parameter exists both in the URL query and in the POST body, only the one in the URL query is considered.

If you decide to use hit and invalidation parameters for a content group, configure the parameters when you configure the content group.

Note: Citrix recommends that you use selectors rather than parameterized content groups, because selectors are more flexible and can be adapted to more types of data.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Selector

Aug 07, 2014

A content group can use a hit selector to retrieve cache hits or use an invalidation selector to expired cached objects and fetch new ones from the origin server.

A selector contains a name and a logical expression, called an *advanced expression*.

For more information about advanced expressions, see "Policies and Expressions."

To configure a selector, you assign it a name and enter one or more expressions. As a best practice, a selector expression should include the URL stem and host, unless there is a strong reason to omit them.

To configure a selector by using the command line interface

At the command prompt, type:

add cache selector <selectorName> (<rule> ...)
For information about configuring the expression or expressions, see "To configure a selector expression by using the command line interface."

Examples

```
>add cache selector product_selector "http.req.url.query.value(\"ProductId\")" "http.req.url.query.value(\"BatchNum\")" "http.req.url.query.value(\"depotLocation\")"
> add cache selector batch_selector "http.req.url.query.value(\"ProductId\")" "http.req.url.query.value(\"BatchId\")" "http.req.url.query.value(\"depotLocation\")"
> add cache selector product_id_selector "http.req.url.query.value(\"ProductId\")"
> add cache selector batchnum_selector "http.req.url.query.value(\"BatchNum\")" "http.req.url.query.value(\"depotLocation\")"
> add cache selector batchid_selector "http.req.url.query.value(\"depotLocation\")" "http.req.url.query.value(\"BatchId\")"
```

To configure a selector by using the configuration utility

Navigate to Optimization > Integrated Caching > Cache Selectors, and add the cache selector.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

About Content Groups

Mar 16, 2012

A content group is a container for cached objects that can be served in a response. When you first enable the integrated cache, cacheable objects are stored in a content group named Default. You can create new content groups that have unique properties. For example, you can define separate content groups for image data, bug reports, and stock quotes, and you can configure the stock quote content group to be refreshed more often than the other groups.

You can configure expiration of an entire content group or selected entries in a content group.

The data in a content group can be static or dynamic, as follows:

- **Static content groups**. Finds an exact match between the URL stem and host name on the request and the URL stem and host name of the response.
- **Dynamic content groups**. Looks for objects that contains particular parameter-value pairs, arbitrary strings, or string patterns. Dynamic content groups are useful when caching data that is updated frequently (for example, a bug report or a stock quote).

Process overview: Serving a hit from a content group

1. A user enters search criteria for an item, such as a bug report, and clicks the Find button in an HTML form.
2. The browser issues one or more HTTP GET requests. These requests contain parameters (for example, the bug owner, bug ID, and so on).
3. When the NetScaler appliance receives the requests, it searches for a matching policy, and if it finds a caching policy that matches these requests, it directs the requests to a content group.
4. The content group looks for appropriate objects in the content group, usually based on criteria that you configure in a selector.
   For example, the content group can retrieve responses that match NameField=username and BugID=ID.
5. If it finds matching objects, the NetScaler appliance can serve them to the user's browser, where they are assembled into a complete response (for example, a bug report).

Example: Invalidating an object in a content group

1. A user modifies data (for example, the user modifies the bug report and clicks the Submit button).
2. The browser sends this data in the form of one or more HTTP requests. For example, it can send a bug report in the form of several HTTP POST requests that contain information about the bug owner and bug ID.
3. The NetScaler appliance matches the requests against invalidation policies. Typically, these policies are configured to detect the HTTP POST method.
4. If the request matches an invalidation policy, the NetScaler appliance searches the content group that is associated with this policy, and expires responses that match the configured criteria for invalidation.
   For example, an invalidation selector can find responses that match NameField=username and BugID=ID.
5. The next time the NetScaler appliance receives a GET request for these responses, it fetches refreshed versions from the origin server, caches the refreshed responses, and serves these responses to the user's browser, where they are assembled into a complete bug report.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting Up a Basic Content Group

Aug 07, 2014

By default, all cached data is stored in the default content group. You can configure additional content groups and specify these content groups in one or more policies.

You can configure content groups for static content, and you must configure content groups for dynamic content. You can modify the configuration of any content group, including the default group.

To set up a basic content group by using the command line interface

At the command prompt, type:

add cache contentgroup <name> (-hitSelector <hitSelectorName> -invalSelector <invalidationSelectorName> | -hitParams <hitParamName> -invalParams <invalidationParamName>) -type <type> [-relExpiry <sec> | -relExpiryMilliSec <msec>] [-heurExpiryParam <positiveInteger>]

Examples

> add cache contentgroup Products_Details –hitSelector product_selector –invalSelector id_selector
> add cache contentgroup bugrep -hitParams IssuePage RecordID Template TableId -invalParams RecordID -relExpiry 864000

To set up a basic content group by using the configuration utility

Navigate to Optimization > Integrated Caching > Content Groups, and create the content group.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expiring or Flushing Cached Objects

Oct 28, 2013

If a response does not have an Expires header or a Cache-Control header with an expiration time (Max-Age or Smax-Age), you must expire objects in a content group by using one of the following methods:

- Configure content group expiration settings to determine whether and how long to keep the object.
- Configure an invalidation policy and action for the content group. For more information, see "Configuring Policies for Caching and Invalidation."
- Expire the content group or objects within it manually.

After a cached response expires, the NetScaler appliance refreshes it the next time the client issues a request for the response. By default, when the cache is full, the NetScaler appliance replaces the least recently used response first.

The following list describes methods for expiring cached responses using settings for a content group. Typically, these methods are specified as a percent or in seconds:

- **Manual**. Manually invalidate all responses in a content group or all responses in the cache.
- **Response-based**. Specific expiration intervals for positive and negative responses. Response-based expiry is considered only if the Last-Modified header is missing in the response.
- **Heuristic expiry**. For responses that have a Last-Modified header, heuristic expiry is a percentage of the time since the response was modified (calculated as current time minus the Last-Modified time, multiplied by the heuristic expiry value). For example, if a Last-Modified header indicates that a response was updated 2 hours ago, and the heuristic expiry setting is 10%, cached objects expire after 0.2 hours. This method assumes that frequently updated responses need to be expired more often.
- **Absolute or relative**. Specify an exact (absolute) time when the response expires every day, in HH:MM format, local time or GMT. Local time may not work in all time zones.
  Relative expiration specifies a number of seconds or milliseconds from the time a cache miss causes a trip to the origin server to the expiration of the response. If you specify relative expiration in milliseconds, enter a multiple of 10. This form of expiration works for all positive responses. Last-Modified, Expires, and Cache-Control headers in the response are ignored.

  Absolute and relative expiration override any expiration information in the response itself.

- **On download**. The option Expire After Complete Response Received expires a response as soon as it is downloaded. This is useful for frequently updated responses, for example, stock quotes. By default, this option is disabled.
  Enabling both Flash Cache and Expire After Complete Response Received accelerates the performance of dynamic applications. When you enable both options, the NetScaler appliance fetches only one response for a block of simultaneous requests.

- **Pinned**. By default, when the cache is full the NetScaler appliance replaces the least recently used response first. The NetScaler appliance does not apply this behavior to content groups that are marked as pinned.

If you do not configure expiration settings for a content group, the following are additional options for expiring objects in the group:

- Configure a policy with an INVAL action that applies to the content group.
- Enter the names of content groups when configuring a policy that uses an INVAL action.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How Expiration Methods Are Applied

Expiration works differently for positive and negative responses. Positive and negative responses are described in the table, *Expiration of Positive and Negative Responses* mentioned below.

The following are rules of thumb for understanding the expiration method that is applied to a content group:

- You can control whether the NetScaler appliance evaluates response headers when deciding whether to expire an object.
- Absolute and relative expiration cause the NetScaler appliance to ignore the response headers (they override any expiration information in the response).
- Heuristic expiration settings and "Weak Positive" and "Weak Negative" expiration (labeled as **Default** values in the configuration utility) cause the NetScaler appliance to examine the response headers. These settings work together as follows:
  - The value in an Expires or Cache-Control header overrides these content group settings.
  - For positive responses that lack an Expires or Cache-Control header but have a Last-Modified header, the NetScaler appliance compares heuristic expiration settings with the header value.
  - For positive responses that lack an Expires, Cache-Control, or Last-Modified header, NetScaler appliance uses the "weak positive" value.
  - For negative responses that lack an Expires or Cache-Control header, NetScaler appliance uses the "weak negative" value.

The following table describes how these methods are applied.

Table 1. Expiration of Positive and Negative Responses

Response Type	Expiration Header Type	Content Group Setting	Period the Object Remains in the Cache
Positive	any header	Expire Content After (relExpiry) with no other settings	Use the value of the Expire Content After setting.
Positive	any header	Expire Content At (absExpiry) with no other settings	Subtract current date from the value of the Expire Content At setting.
Positive	any header	Expire Content After (relExpiry) and Expire content at (absExpiry)	Use the smaller of the two values for the content group settings. See the previous rows in this table.
Positive	Last-Modified (with any other headers)	Heuristic (heurExpiry Param) with any other setting	Subtract the Last-Modified date from the current date, multiply the result by the value of the heuristic expiry setting, and then divide by 100.
Positive	Last-Modified (with	Default (positive)	Use the value of the Default (positive) expiry setting.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Type	Expiration Header Type	Content Group Setting	Period the Object Remains in the Cache
	any other headers)	(weakPosRel Expiry) and no other setting	
Positive	Expires or Cache-Control: Max-Age header is present Last-Modified header is absent	Heuristic (heurExpiry Param), Default (positive) (weakPosRel Expiry), or both	Subtract the current date from the Expires or the Cache-Control:Max-Age date.
Positive	no caching headers	Default (positive) (weakPosRel Expiry) and any other expiration setting.	Use the value of the Default (positive) setting.
Positive	no caching headers	Heuristic (heurExpiry Param) is present Default (positive) (weakPosRel Expiry) setting is absent	If the Last-Modified header is absent, the response is not cached or it is cached with an Already Expired status. If the Last-Modified header is present, use the heuristic expiry value.
Negative	Expires or Cache-Control:Max-Age	Expire Content After (relExpiry), Expire Content At (absExpiry), or both settings	Subtract the current date from the value of the Expires header, or use the value of the Cache-Control:Max-Age header.
Negative	Expires or Cache-Control headers are absent	Expire Content After (relExpiry), Expire Content At (absExpiry), or both settings	Response is not cached, or is cached with an Already Expired status.
Negative	Expires or Cache-Control:Max-Age	Any setting	Subtract the current date from the Expires or Cache-Control:Max-Age date.
Negative	Expires and Cache-Control:Max-Age headers are absent	Default (negative) (weakNegRel Expiry)	Use the value of the Default (negative) setting.
Negative	Expires and Cache-Control:Max-Age headers are absent	Any setting other than Default (negative) (weakNegRel Expiry)	Object is not cached or is cached with an Already Expired status.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Expiring a Content Group Manually

Aug 04, 2014

You can manually expire all of the entries in a content group.

To manually expire all responses in a content group by using the command line interface

At the command prompt, type:

expire cache contentGroup <name>

To manually expire all responses in a content group by using the configuration utility

Navigate to Optimization > Integrated Caching > Content Groups, select the content group, and click Invalidate to expire all the responses in a content group.

To manually expire all responses in the cache by using the configuration utility

Navigate to Optimization > Integrated Caching > Content Groups, and click Invalidate All to expire all the responses in cache.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Periodic Expiration of a Content Group

Aug 08, 2014

You can configure a content group so that it performs selective or full expiration of its entries. The expiration interval can be fixed or relative.

To configure content group expiration by using the command line interface

At the command prompt, type:

set cache contentgroup <name> (-relExpiry | -relExpiryMilliSec | -absExpiry | -absExpiryGMT | -heurExpiryParam | -weakPosRelExpiry | -weakNegRelExpiry | -expireAtLastBye) <expirationValue>

To configure content group expiration by using the configuration utility

Navigate to Optimization > Integrated Caching > Content Groups, select the content group, and specify expiry method.

Expiring Individual Responses

Expiring a response forces the NetScaler appliance to fetch a refreshed copy from the origin server. Responses that do not have validators, for example, ETag or Last-Modified headers, cannot be revalidated. As a result, flushing these responses has the same effect as expiring them.

To expire a cached response in a content group for static data, you can specify a URL that must match the stored URL. If the cached response is part of a parameterized content group, you must specify the group name as well as the exact URL stem. The host name and the port number must be the same as in the host HTTP request header of the cached response. If the port is not specified, port 80 is assumed.

To expire individual responses in a content group by using the command line interface

At the command prompt, type:

expire cache object -url <URL> -host <hostName> [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET | POST]

To expire individual responses in a content group by using the command line interface (selector-based)

At the command prompt, type the following command:

expire cache object -locator <positiveInteger>

To expire a cached response by using the configuration utility

Navigate to Optimization > Integrated Caching > Cached Objects, select the cached response, and expire.

To expire a response by using the Lookup tool (selector-based)

Navigate to Optimization > Integrated Caching > Cached Objects, click Search and, set the search criteria to find the required cached response and expire.

Flushing Responses in a Content Group

You can remove, or flush, all responses in a content group, some responses in a group, or all responses in the cache. Flushing

© 1999-2017 Citrix Systems, Inc. All rights reserved.

a cached response frees up memory for new cached responses.

Note: To flush responses for more than one object at a time, use the configuration utility method. The command line interface does not offer this option.

To flush responses from a content group by using the command line interface

At the command prompt, type:

flush cache contentGroup <name> [-query <queryString> | [-selectorValue <selectorExpressionIDList> -host <hostName>]]

To flush responses from a content group by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Content Groups.
2. In details pane, flush the responses as follows:
   - To flush all responses in all content groups, click Invalidate All, and flush all the responses.
   - To flush responses in a particular content group, select the content group, click Invalidate, and flush all the responses.

Note: If this content group uses a selector, you can selectively flush responses by entering a string in the Selector value text box, entering a host name in the Host text box. Then click Flush and OK. The Selector value can be a query string of up to 2319 characters that is used for parameterized invalidation.
If the content group uses an invalidation parameter, you can selectively flush responses by entering a string in the Query field.

If the content group uses an invalidation parameter and Invalidate objects belonging to target host is configured, enter strings in the Query and Host fields.

To flush a cached response by using the command line interface

At the command prompt, type:

flush cache object -locator <positiveInteger> | -url <URL> -host <hostName> [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET | POST]

To flush a cached response by using the configuration utility

Navigate to Optimization > Integrated Caching > Cached Objects, select the cached object, and flush.

Deleting a Content Group

You can remove a content group if it is not used by any policy that stores responses in the cache. If the content group is bound to a policy, you must first remove the policy. Removing the content group removes all responses stored in that group.

You cannot remove the Default, BASEFILE, or Deltajs group. The Default group stores cached responses that do not belong in any other content group.

To delete a content group by using the command line interface

At the command prompt, type:

rm cache contentgroup<name>

To delete a content group by using the configuration utility

Navigate to Optimization > Integrated Caching > Contetn Groups, select the content group, and delete.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Policies for Caching and Invalidation

Aug 30, 2013

Policies enable the integrated cache to determine whether to try to serve a response from the cache or the origin. The Citrix NetScaler appliance provides built-in policies for integrated caching, and you can configure additional policies. When you configure a policy, you associate it with an action. An action either caches the objects to which the policy applies or invalidates (expires) the objects. Typically, you based caching policies on information in GET and POST requests. You typically base invalidation policies on the presence of the POST method in requests, along with other information. You can use any information in a GET or POST request in a caching or an invalidation policy.

You can view some of the built-in policies in the integrated cache's Policies node in the configuration utility. The built-in policy names begin with an underscore (_).

Actions determine what the NetScaler appliance does when traffic matches a policy. The following actions are available:

- **Caching actions**. Policies that you associate with the CACHE action store responses in the cache and serve them from the cache.
- **Invalidation actions**. Policies that you associate with the INVAL action immediately expire cached responses and refresh them from the origin server. Note that for Web-based applications, invalidation policies often evaluate POST requests.
- **"Do not cache" actions**. Policies that you associate with a NOCACHE action never store objects in the cache.
- **"Provisionally cache" actions**. Policies that you associate with a MAYCACHE or MAYNOCACHE action depend on the outcome of additional policy evaluations.

Although the integrated cache does not store objects specified by the LOCK method, you can invalidate cached objects upon receipt of a LOCK request. For invalidation policies only, you can specify LOCK as a method by using the expression http.req.method.eq("lock"). Unlike policies for GET and POST requests, you must enclose the LOCK method in quotes because the NetScaler appliance recognizes this method name as a string only.

After you create a policy, you bind it to a particular point in the overall processing of requests and responses. Although you create a policy before binding it, you should understand how the bind points affect the order of processing before you create your policies.

The policies bound to a particular bind point constitute a policy bank. You can use goto expressions to modify the order of execution in a policy bank. You can also invoke policies in other policy banks. In addition, you can create labels and bind policies to them. Such a label is not associated with a processing point, but the policies bound to it can be invoked from other policy banks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Actions to Associate with Integrated Caching Policies

Mar 16, 2012

The following table describes actions for integrated caching policies.

Table 1. Actions That You Can Associate with an Integrated Caching Policy

Action	Specifies
CACHE	Serves a response from the cache if the response has not expired. If the response must be fetched from the origin server, the NetScaler appliance caches the response before serving it. Even data that is updated and accessed frequently can be cached. For example, stock quotes are updated frequently, but they can be cached so that they can be served quickly to multiple users. If necessary, cached data can be refreshed immediately after it is downloaded. A CACHE action can be overridden by built-in policies.
NOCACHE	Always fetches the response from the origin server and marks the response as non-storable. You typically configure NOCACHE policies for data that is sensitive or personalized.
MAY_CACHE	Used in a request-time policy, this setting provisionally enables a response to be stored in a content group, pending evaluation of response-time policies. The following are possible: • If a matching response-time policy has a CACHE action but does not specify a content group, the response is stored in the Default group unless built-in policies override this policy. • If a matching response-time policy has a CACHE action and specifies the same content group as the one in the request-time policy, the response is stored in the named content group unless built-in policies override this policy. • If a matching response-time policy has a CACHE action but specifies a different content group from the one in the request-time policy, a NOCACHE action is applied. • If a matching response-time policy has a NOCACHE action, perform a NOCACHE action. • If there is no matching response-time policy, a CACHE action is applied, unless a built-in policy overrides this policy.
MAY_NOCACHE	For a request-time policy, this setting provisionally prevents caching the response. At response time, one of following actions is taken: • If no response-time policy matches the request, the final action is NOCACHE. • If a matching response-time policy contains a CACHE action, the final action is CACHE, unless built-in policies override this policy. • If a matching response-time policy contains a NOCACHE action, the final action is NOCACHE. • If a matching response-time policy has a CACHE action but does not specify a content group, the final action is to CACHE the response in the Default content group, unless built-in policies override this policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Action	Specifies
INVAL	Expires cached responses. Depending on how the policy and the content group are configured, all responses in one or more content groups are expired, or selected objects in the content group are expired. Note: You can specify INVAL actions in request-time policies only.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Bind Points for a Policy

Oct 28, 2013

You can bind the policy to one of the following bind points:

- **A global policy bank**. These are the request-time default, request-time override, response-time default, and response-time override policy banks, as described in "Order of Policy Evaluation."
- **A virtual server**. Policies that you bind to a virtual server are processed after the global override policies and before the global default policies, as described in "Order of Policy Evaluation." Note that when binding a policy to a virtual server, you bind it to either request-time or response-time processing.
- **An ad-hoc policy label**. A policy label is a name assigned to a policy bank. In addition to the global labels, the integrated cache has two built-in custom policy labels:
  - **_reqBuiltinDefaults**. This policy label, by default, is invoked from the request-time default policy bank.
  - **_resBuiltinDefaults**. This policy label, by default, is invoked from the response-time default policy bank.

  You can also define new policy labels. Policies bound to a user-defined policy label must be invoked from within a policy bank for one of the built-in bind points. For more information about creating a policy label, see "Configuring a Policy Label in the Integrated Cache." For more information about policy label invocation, see "Configuring a Policy Bank for Caching."

Important: You should bind a policy with an INVAL action to a request-time override or a response-time override bind point. To delete a policy, you must first unbind it.

Order of Policy Evaluation

For an advanced policy to take effect, you must ensure that the policy is invoked at some point during the NetScaler appliance's processing of traffic. To specify the invocation time, you associate the policy with a bind point. The following are the bind points, listed in order of evaluation:

- **Request-time override**. If a request matches a request-time override policy, by default request-time policy evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.
- **Request-time load balancing virtual server**. If policy evaluation cannot be completed after all the request-time override policies are evaluated, the NetScaler appliance processes request-time policies that are bound to load balancing virtual servers. If the request matches one of these policies, evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.
- **Request-time content switching virtual server**. Policies that are bound to this bind point are evaluated after request-time policies that are bound to load balancing virtual servers.
- **Request-time default**. If policy evaluation cannot be completed after all request-time, virtual server-specific policies are evaluated, the NetScaler appliance processes request-time default policies. If the request matches a request-time default policy, by default request-time policy evaluation ends and the NetScaler appliance stores the action that is associated with the matching policy.
- **Response-time override**. Similar to request-time override policy evaluation.
- **Response-time load balancing virtual server**. Similar to request-time virtual server policy evaluation.
- **Response-time content switching virtual server**. Similar to request-time virtual server policy evaluation.
- **Response-time default**. Similar to request-time default policy evaluation.

You can associate multiple policies with each bind point. To control the order of evaluation of the policies associated with the bind point you configure a priority level. In the absence of any other flow control information, policies are evaluated according to priority level, starting with the lowest numeric priority value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

After all integrated caching policies have been evaluated, if there are conflicting actions specified in request-time and response-time policies, the NetScaler appliance determines the final action as specified in the table, "Actions That You Can Associate with an Integrated Caching Policy."

Note: Request-time policies for POST data or cookie headers must be invoked during request-time override evaluation, because the built-in request-time policies in the integrated cache return a NOCACHE action for POST requests and a MAY_NOCACHE action for requests with cookies. Note that you would associate MAY_CACHE or MAY_NOCACHE actions with a request-time policy that points to a parameterized content group. The response time policy determines whether the transaction is stored in the cache.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Policy in the Integrated Cache

Aug 07, 2014

You configure new policies to handle data that the built-in policies cannot process. You configure separate policies for caching, preventing caching from occurring, and for invalidating cached data. Following are the main components of a policy for integrated caching:

- Rule: A logical expression that evaluates an HTTP request or response.
- Action: You associate a policy with an action to determine what to do with a request or response that matches the policy rule.
- Content groups: You associate the policy with one or more content groups to identify where the action is to be performed.

To configure a policy for caching by using the command line interface

At the command prompt, type:

add cache policy <policyName> -rule <expression> -action CACHE | MAY_CACHE | NOCACHE | MAY_NOCACHE [-storeInGroup <contentGroupName>] [-undefAction NOCACHE | RESET]

Examples

```
> add cache policy image_cache -rule "http.req.url.contains(\"jpg\") || http.req.url.contains(\"jpeg\")" -action CACHE -storeingroup myImages_group -undefaction NOCACHE
> add cache policy bugReportPolicy -rule "http.req.url.query.contains(\"IssuePage\")" -action CACHE -storeInGroup bugReportGroup
> add cache policy my_form_policy -rule "http.req.header(\"Host\")contains(\"my.company.com\") && http.req.method.eq(\"GET\") && http.req.url.query.contains(\"v=7\")" -action CACHE -storeInGroup my_
> add cache policy viewproducts_policy –rule "http.req.url.contains(\"viewproducts.aspx\")" –action CACHE -storeInGroup Product_Details
```

To configure a policy for invalidation by using the command line interface

At the command prompt, type:

add cache policy <policyName> -rule <expression> -action INVAL [-invalObjects "<contentGroupName1>[,<selectorName1>"]. . .]] | [-invalGroup <contentGroupName1>[, <contentGroupName2>. . .]] [-undefAction NOCACHE | RESET]

Examples

```
> add cache policy invalidation_events_policy -rule "http.req.header(\"Host\")contains(\"my.company.com\") && http.req.method.eq(\"GET\") && http.req.url.query.contains(\"v=8\") -action INVAL -invalObje
> add cache policy inval_all -rule "http.req.method.eq(\"POST\") && http.req.url.contains(\"jpeg\")" -action INVAL -invalGroups myImages_group myApps_group PDF_group
> add cache policy bugReportInvalidationPolicy -rule "http.req.url.query.contains(\"TransitionForm\")" -action INVAL -invalObjects bugReport
> add cache policy editproducts_policy –rule "http.req.url.contains(\"editproducts.aspx\")" –action INVAL -invalObjects "Product_Details,batchnum_sel" "Products_In_Depots,batchid_sel"
```

To configure a policy for caching or invalidation by using the configuration utility

Navigate to Optimization > Integrated Caching > Policies, and create the new policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Globally Binding an Integrated Caching Policy

Aug 08, 2014

When you globally bind a policy, it is available to all virtual servers on the NetScaler appliance.

To bind an integrated caching policy globally by using the command line interface

At the command prompt, type:

bind cache global <policy> -priority <positiveInteger> [-type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT] [-gotoPriorityExpression <expression>] [-invoke <labelType> <labelName>]

Example

```
> bind cache global myCachePolicy -priority 100 -type req_default
```

Note that the type argument is optional for globally bound policies, to maintain backward compatibility with policies that you defined using earlier versions of the NetScaler appliance. If you omit the type, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. If the rule contains both request time and response time parameters, it is bound to RES_DEFAULT. Following is an example of a binding that omits the type.

```
> bind cache global myCache Policy 200
```

To bind an integrated caching policy globally by using the configuration utility

Navigate to Optimization > Integrated Caching, click Cache Policy Manager, and bind policies by specifying the relevant bind point and connection type (Request/Response).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding an Integrated Caching Policy to a Virtual Server

Aug 08, 2014

When you bind a policy to a virtual server, it is available only to requests and responses that match the policy and that flow through the relevant virtual server.

When using the configuration utility, you can bind the policy using the configuration dialog box for the virtual server. This enables you to view all of the policies from all NetScaler modules that are bound to this virtual server. You can also use the Policy Manager configuration dialog for the integrated cache. This enables you to view only the integrated caching policies that are bound to the virtual server.

To bind an integrated caching policy to a virtual server by using the command line interface

At the command prompt, type:

- bind lb vserver <name>@ -policyName <policyName> -priority <positiveInteger> -type (REQUEST | RESPONSE)
- bind cs vserver <name>@ -policyName <policyName> -priority <positiveInteger> -type (REQUEST | RESPONSE)

To bind an integrated caching policy to a virtual server by using the configuration utility (virtual server method)

- CS Virtual Server - Navigate toTraffic Management > Content Switching > Virtual Servers, select the virtual server, and bind relevant cache policies.
- LB Virtual Server - Navigate toTraffic Management > Load Balancing > Virtual Servers, select the virtual server, and bind relevant cache policies.

To bind an integrated caching policy to a virtual server by using the configuration utility (Policy Manager method)

Navigate to Optimization > Integrated Caching, click Cache Policy Manager, and bind cache policies by specifying the relevant bind point and connection type.
Note: You can bind cache policies to both LB virtual server and CS virtual server by selecting the appropriate bind point.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example: Caching Compressed and Uncompressed Versions of a File

Mar 16, 2012

By default, a client that can handle compression can be served uncompressed responses or compressed responses in gzip, deflate, compress, and pack200-gzip format. If the client handles compression, an Accept-Encoding:compression format header is sent in the request. The compression type accepted by the client must match the compression type of the cached object. For example, a cached .gzip file cannot be served in response to a request with an Accept-Encoding:deflate header.

A client that cannot handle compression is served a cache miss if the cached response is compressed.

For dynamic caching, you need to configure two content groups, one for compressed data and one for uncompressed versions of the same data. The following is an example of configuring the selectors, content groups, and policies for serving uncompressed files from the cache to clients that cannot handle compression, and serving compressed versions of the same files to client that can handle compression.

```
add cache selector uncompressed_response_selector http.req.url "http.req.header(\"Host\")"
add cache contentGroup uncompressed_group -hitSelector uncompressed_responst_selector -invalSelector uncomp_resp_sel
add cache policy cache_uncompressed -rule "HTTP.REQ.URL.CONTAINS(\"xyz\") && !HTTP.REQ.HEADER(\"Accept-Encoding\").EXISTS" -action CACHE -storeInGroup uncompressed_group
bind cache global cache_uncompressed -priority 100 -gotoPriorityExpression END -type REQ_OVERRIDE
add cache selector compressed_response_selector HTTP.REQ.URL "HTTP.REQ.HEADER(\"Host\")" "HTTP.REQ.HEADER(\"Accept-Encoding\")"
add cache contentGroup compressed_group -hitSelector compressed_response_selector
add cache policy cache_compressed -rule "HTTP.REQ.URL.CONTAINS(\"xyz\") && HTTP.REQ.HEADER(\"Accept-Encoding\").EXISTS" -action CACHE -storeInGroup compressed_group
bind cache global cache_compressed -priority 200 -gotoPriorityExpression END -type REQ_OVERRIDE
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Policy Bank for Caching

Aug 08, 2014

All of the policies that are associated with a particular bind point are collectively known as a policy bank. In addition to configuring priority levels for policies in a bank, you can modify the order of evaluation order in a bank by configuring Goto expressions. You can further modify the evaluation order by invoking an external policy bank from within the current policy bank. You can also configure new policy banks, to which you assign your own labels. Because such policy banks are not bound to any point in the processing cycle, they can be invoked only from within other policy banks. For convenience, policy banks whose labels do not correspond to a built-in bind point are called policy labels.

In addition to controlling order of policy evaluation by binding the policy and assigning a priority level, as described in "Binding Policies That Use the Default Syntax", you can establish the flow within a bank of policies by configuring a Goto expression. A Goto expression overrides the flow that is determined by the priority levels. You can also control the evaluation flow by invoking an external policy bank after evaluating an entry in the current bank. Evaluation always returns to the current bank after evaluation has completed for the external bank.

The following table summarizes the entries to control evaluation in a policy bank.

Table 1. Entries to Control Evaluation Flow in a Policy Bank

Attribute	Specifies
Name	The name of a policy, or, to invoke another policy bank without evaluating the policy, the keyword NOPOLICY. You can specify NOPOLICY more than once in a policy bank, but you can specify a named policy only once.
Priority	An integer. The lower the integer, the higher the priority.
Goto Expression	Determines the next policy or policy bank to evaluate. You can provide one of the following values: • **NEXT**: Go to the policy with the next higher priority. • **END**: Stop evaluation. • **USE_INVOCATION_RESULT**: Applicable if this entry invokes another policy bank. If the final Goto in the invoked bank has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy bank performs a NEXT. • **Positive number**: Priority number of the next policy to be evaluated. • **Numeric expression**: Expression that produces the priority number of the next policy to be evaluated. The Goto can only proceed forward in a policy bank. Omitting the Goto expression is the same as specifying END.
Invocation Type	Designates a policy bank type. The value can be one of the following: • **Request Vserver**: Invokes request-time policies that are associated with a virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Attribute	Specifies
	• **Response Vserver**: Invokes response-time policies that are associated with a virtual server. • **Policy label**: Invokes another policy bank, as identified by the policy label for the bank.
Invocation Name	Name of a virtual server or a policy label, depending on the value that you specified for the Invocation Type.

The integrated cache has two built-in policy labels, and you can configure additional policy labels:

- **_reqBuiltInDefaults**: This policy label is invoked from the request-time default bind point.
- **_resBuiltInDefaults**: This policy label is invoked from the response-time default bind point.

Note: For information about creating policy labels, see "Configuring a Policy Label in the Integrated Cache."

To invoke a policy label in a caching policy bank by using the command line interface

At the command prompt, type:

bind cache policylabel <labelName> -policname<policyName> -priority<priority> [-gotoPriorityExpression <gotopriorityExpression>] [-invoke <labelType> <labelName>]

To invoke a policy label in a caching policy bank by using the configuration utility

1. Navigate to Optimization > Integrated Caching, click Cache policy manager, and specify the relevant bind point (Override Global or Default Global) and connection type to view the list of policies bound to this bind point.
2. If you want to invoke a policy label without evaluating a policy, click NOPOLICY.
   Note: To invoke an external policy bank, click the field in the Invoke Type column, and select the type of policy bank that you want to invoke at this point in the policy bank. This can be a global label or a virtual server bank. In the Invoke Name field, enter the label or virtual server name. See "Entries to Control Evaluation Flow in a Policy Bank" for details.

To invoke a caching policy label in a virtual server policy bank by using the command line interface

At the command prompt, type:

- bind lb vserver <name>@ -policyName <policyName>|<NOPOLICY-CACHE> -priority <positiveInteger> -gotoPriorityExpression <expression> -type REQUEST|RESPONSE -invoke <labelType> <labelName>
- bind cs vserver <name> -policyName <policyName>|<NOPOLICY-CACHE> -priority <positiveInteger> -gotoPriorityExpression <expression> -type REQUEST|RESPONSE -invoke <labelType> <labelName>

For more information, see "Entries to Control Evaluation Flow in a Policy Bank."

To invoke a caching policy label in a virtual server policy bank by using the configuration utility

1. Navigate to Traffic Management > Load Balancing/Contetn Switching > Virtual Servers, select the virtual server, and click Policies.
2. If you are configuring an existing entry in this bank, skip this step. If you are adding a new policy to this policy bank, or you want to use the “dummy” NOPOLICY entry, click Add, and do one of the following:
   - To configure a new policy, click Cache and configure the new policy as described in "Configuring a Policy in the Integrated Cache."
   - To invoke a policy bank without processing a policy a rule, select the NOPOLICY-CACHE option.

   Note: To invoke an external policy bank, click the field in the Invoke Type column, and select the type of policy bank that you want to invoke at this point in the policy bank. This can be a global label or a virtual server bank. In the Invoke Name

© 1999-2017 Citrix Systems, Inc. All rights reserved.

field, enter the label or virtual server name. See "Entries to Control Evaluation Flow in a Policy Bank" for details.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Policy Label in the Integrated Cache

Aug 08, 2014

In addition to configuring policies in a policy bank for one of the built-in bind points or a virtual server, you can create caching policy labels and configure banks of policies for these new labels.

A policy label for the integrated cache can be invoked only from one of the bind points that you can view in the Policy Manager in the **Integrated Caching** details pane (request override, request default, response override, or response default) or the built-in policy labels _reqBuiltinDefaults and _resBuiltinDefaults. You can invoke a policy label any number of times unlike a policy, which can only be invoked once.

The configuration utility provides an option to rename a policy label. Renaming a policy label does not affect the process of evaluation of the policies bound to the label.

Note: You can use the NOPOLICY “dummy” policy to invoke any policy label from another policy bank. The NOPOLICY entry is a placeholder that does not process a rule.

To configure a policy label for caching by using the command line interface

At the command prompt, type the following command to create a policy label and verify the configuration:

- add cache policylabel <labelName> -evaluates (REQ|RES)
- show cache policylabel <labelName>

Invoke this policy label from a policy bank. For more information, see "Configuring a Policy Bank for Caching."

To configure a policy label for caching by using the configuration utility

Navigate to Optimization > Integrated Caching > Policy Labels, add a policy label, and bind the cached policies.
Note: To ensure that the NetScaler ADC processes the policy label at the right time, configure an invocation of this label in one of the policy banks that are associated with the built-in bind points

To rename a policy label by using the configuration utility

Navigate to Optimization > Integrated Caching > Policy Labels, select the policy label, and rename.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Unbinding and Deleting an Integrated Caching Policy and Policy Label

Aug 08, 2014

You can unbind a policy from a policy bank, and you can delete it. To delete the policy, you must first unbind it. You can also remove a policy label invocation and delete a policy label. To delete the policy label, you must first remove any invocations that you have configured for the label.

You cannot unbind or delete the labels for the built-in bind points (request default, request override, response default, and response override).

To unbind a global caching policy by using the command line interface

At the command prompt, type:

unbind cache global <policy>

To unbind a virtual server-specific caching policy by using the command line interface

At the command prompt, type:

(unbind lb vserver | unbind cs vserver) <vserverName> -policyName <policyName> -type (REQUEST | RESPONSE)

To delete a caching policy by using the command line interface

At the command prompt, type:

rm cache policy <policyName>

To unbind a caching policy by using the configuration utility

Navigate to Optimization > Integrated Caching, click Cache Policy Manager, and unbind policies by specifying the relevant bind point and connection type (Request/Response).

To delete a policy label invocation by using the configuration utility

1. Navigate to Optimization > Integrated Caching, click Cache policy manager, and specify the relevant bing point (LB virtual server or CS virtual server) and connection type to view the list of cache policies bound to this virtual server.
2. In the policy Invoke column, clear the entry.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Caching Support for Database Protocols

Sep 02, 2013

The integrated cache monitors database requests that flow through the Citrix® NetScaler® appliance and caches them as determined by the cache policies. Users have to configure the cache polices for MYSQL and MSSQL protocols, because the NetScaler does not provide any default policies for these protocols. When configuring the protocols, remember that request based policies currently support CACHE and INVAL actions, while response based policies currently support only NOCACHE action. After configuring the policies, bind them to virtual servers. MYSQL and MSSQL policies, both request and response, can be bound only to virtual servers

Before creating a cache policy, create a cache content group of type MYSQL or MSSQL. When you create a MYSQL or MSSQL cache content group, associate at least one hit selector with it. See "Setting Up a Basic Content Group" for setting up cache content groups.

The following example illustrates the procedure for configuring and verifying cache support for SQL protocols.

```
> enable feature IC
> set cache parameter -memlimit 100
> add cache selector sel1 mssql.req.query.text

> add cache contentgroup cg1  -type "MSSQL" -hitselector "sel1" -invalselector "inval_sel" -relExpiry "500" -maxResSize
 "100"
> add cache policy cp1  -rule "mssql.req.query.command.contains(\"select\")" -action "CACHE" -storeInGroup "cg1"
> add cache policy cp2  -invalObjects "cg1" -rule "mssql.req.query.text.contains(\"insert\")" -action "INVAL"
> add db user user1  -password "Pass1"
> add service svc_sql_1 10.102.147.70 mssql 64834 -healthMonitor "NO" -downstateflush "ENABLED"
> add lb vserver lb_mssql1 mssql 10.102.147.77 1433  -lbmethod "roundrobin"
> bind lb vserver lb_mssql1 svc_sql_1
> bind lb vserver lb_mssql1 -policyName cp1  -type "REQUEST" -priority "2"
> bind lb vserver lb_mssql1 -policyName cp2  -type "REQUEST" -priority "1"

> show cache selector sel1
    Name:sel1
    Expressions:
    1)mssql.req.query.text
> show cache policy cp1
    Name:cp1
    Rule:mssql.req.query.command.contains("select")
    CacheAction:CACHE
    Stored in group: cg1
    UndefAction:Use Global
    Hits:2
    Undef Hits:0
    Policy is bound to following entities
    1) Bound to:
      REQ VSERVER lb_mssql1
      Priority:2
      GotoPriorityExpression: END
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: The methods for reducing flash crowds, as explained in "Reducing Flash Crowds", are not supported for MYSQL and MSSQL protocols.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Expressions for Caching Policies and Selectors

Oct 28, 2013

A request-time expression examines data in request-time transaction, and a response-time expression examines data in a response-time transaction. In a policy for caching, if an expression matches data in a request or response, the Citrix NetScaler appliance takes the action associated with the policy. In a selector, request-time expressions are used to find matching responses that are stored in a content group.

Before configuring policies and selectors for the integrated cache, you need to know, at minimum, the host names, paths, and IP addresses that appear in HTTP request and response URLs. And you probably need to know the format of entire HTTP requests and responses. Programs such as Live HTTP Headers (http://livehttpheaders.mozdev.org/) or HTTPFox (https://addons.mozilla.org/en-US/firefox/addon/6647) can help you investigate the structure of the HTTP data that your organization works with.

Following is an example of an HTTP GET request for a stock quote program:

```
GET /quote.dll?page=dynamic&mode=data&mode=stock&symbol=CTXS&page=multi&selected=CTXS&random=0.00792039478975548 HTTP/1.1
Host: quotes.mystockquotes.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9) Gecko/2008052906 Firefox/3.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate,compress,pack200-gzip
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://quotes.mystockquotes.com/quote.dll?mode=stock&symbol=CTXS&page=multi&selected=CTXS
Cookie: __qca=1210021679-72161677-10297606
```

When configuring an expression, note the following limitations:

Table 1. Restrictions on Request-Time and Response-Time Expressions

Expression Type	Restrictions
Request	Do not configure request-time expressions in a policy with a CACHE or NOCACHE action. Use MAY_CACHE or MAY_NOCACHE instead.
Response	Configure response-time expressions in caching policies only. ● Selectors can use only request-time expressions. ● Do not configure response-time expressions in a policy with an INVAL action. Do not configure response-time expressions in a policy with a CACHE action and a parameterized content group. Use the MAY_CACHE action.

Note: For a comprehensive discussion of advanced expressions, see "Policies and Expression."

Expression Syntax

Following are basic components of the syntax:

- Separate keywords with periods (.), as follows:
  http.req.url
- Enclose string values in parentheses and quotes, as follows:
  http.req.url.query.contains("this")
- When configuring an expression from the command line, you must escape internal quote marks (the quotes that delimit values in the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

expression, as opposed to the quotes that delimit the expression). One method is to use a slash, as followings:
\"abc\"

Selector expressions are evaluated in order of appearance, and multiple expressions in a selector definition are joined by a logical AND. Unlike selector expressions, you can specify Boolean operators and modify the precedence in an advanced expression for a policy rule.

Configuring an Expression in a Caching Policy or a Selector

Updated: 2014-08-04

Note that on the command line, the syntax for a policy expression is somewhat different from a selector expression. For a comprehensive discussion of advanced expressions, see "Policies and Expressions."

To configure a policy expression by using the command line interface

1. Start the policy definition as described in "Globally Binding an Integrated Caching Policy."
2. To configure the policy rule, delimit the entire rule in quotes, and delimit string values within the rule in escaped quotes.

   The following is an example:

   "http.req.url.contains(\"jpg\")"
3. To add Boolean values, insert &&, ||, or ! operators.

   The following are examples:

   "http.req.url.contains(\"jpg\") || http.req.url.contains(\"jpeg\")"
   "http.req.url.query.contains(\"IssuePage\")"
   "http.req.header(\"Host\")contains(\"my.company.com\") && http.req.method.eq(\"GET\") && http.req.url.query.contains(\"v=7\")"
4. To configure an order of evaluation for the constituent parts of a compound

   "http.req.url.contains(\"jpg\") || (http.req.url.contains(\"jpeg\") && http.req.method.eq(\"GET\"))"

To configure a selector expression by using the command line interface

1. Start the selector definition as described in "About Content Groups."
2. To configure the selector expression, delimit the entire rule in quotes, and delimit string values within the rule in escaped quotes.

   The following is an example:

   "http.req.url.contains(\"jpg\")"
3. You cannot add Boolean values, insert &&, ||, or ! operators. Enter each expression element delimited in quotes. Multiple expressions in the definition are treated as a compound expression joined by logical ANDs.

   The following are examples:

   "http.req.url.query.value(\"ProductId\")" "http.req.url.query.value(\"BatchNum\")" "http.req.url.query.value(\"depotLocation\")"

To configure a policy or selector expression by using the configuration utility

1. Start the policy or selector definition as described in "To configure a policy for caching or invalidation by using the configuration utility" or "To configure a selector by using the configuration utility."
2. In the Expression field, you can either manually type the default syntax by clicking Switch to Classic Syntax or create new expression using Expression Editor.
3. To insert an operator between two parts of a compound expression, click the Operators button and select the operator type. The following is an example of a configured expression with a Boolean OR (signaled by double vertical bars, ||):

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Click Frequently Used Expressions drop-down to insert the commonly used expressions.
5. To test the expression, click the Evaluate. In the Expression Evaluator dialog box, select the Flow Type that matches the expression. In the data field, paste the HTTP request or response that you hope to parse using the expression, and click Evaluate.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Displaying Cached Objects and Cache Statistics

Oct 12, 2016

You can view particular cached objects, and you can view summary statistics on cache hits, misses, and memory usage. The statistics provide insight on the amount of data that is being served from the cache, what items are responsible for the largest performance benefit, and what you can tune to improve cache performance.

This section includes the following details:

- Viewing Cached Objects
- Finding Particular Cached Responses
- Viewing Cache Statistics

Viewing Cached Objects

Updated: 2014-08-04

After enabling caching, you can view details for cached objects. For example, you can view the following items:

- Response sizes and header sizes
- Status codes
- Content groups
- ETag, Last-Modified, and Cache-Control headers
- Request URLs
- Hit parameters
- Destination IP addresses
- Request and response times

To view a list of cached objects by using the command line interface

At the command prompt, type:

show cache object

Table 1. Properties of Cached Objects

Properties	Specifies
Response size (bytes)	The size of the response header and body.
Response header size (bytes)	The size of the header portion of the response.
Response status code	The status code sent with the response.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Properties	Specifies
ETag	The ETag header inserted in the response. Typically, this header indicates whether the response has changed recently.
Last-Modified	The Last-Modified header inserted in the response. This header indicates the date that the response was last changed.
Cache-Control	The Cache-Control header inserted in the response.
Date	The Date header that indicates when the response was sent.
Contentgroup	The content group where the response is stored.
Complex match	If this object was cached on the basis of parameterized values, this field value is YES.
Host	The host specified in the URL that requested this response.
Host port	The listen port for the host specified in the URL that requested this response.
URL	The URL issued for the stored response.
Destination IP	The IP address of the server from which this response was fetched.
Destination port	The listen port for the destination server.
Hit parameters	If the content group that stores the response uses hit parameters, they are listed in this field.
Hit selector	If this content group uses a hit selector, it is listed in this field.
Inval selector	If this content group uses an invalidation selector, it is listed in this field.
Selector Expressions	If this content group uses a selector, this field displays the expression that defines the selection rule.
Request time	The time in milliseconds since the request was issued.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Properties	Specifies
Response time	Time in milliseconds since the cache started to receive the response.
Age	Amount of time the object has been in the cache.
Expiry	Amount of time after which the object is marked as expired.
Flushed	Whether the response has been flushed after expiry.
Prefetch	If Prefetch has been configured for this content group, the amount of time before expiry during which the object is fetched from the origin. Prefetch does not apply to negative objects (for example, 404 "object not found" responses).
Current readers	Approximately the current number of hits being served. When a response with a Content-Length header object is being downloaded, the current misses and the current readers values are each typically 1. When a chunked response object is being downloaded, the current misses value is typically 1, but the current readers value is typically 0, because the chunked response that is served to the client does not come from the integrated caching buffers.
Current misses	The current number of requests that resulted in a cache miss and fetching from the origin server. This value is typically 0 or 1. If Poll Every Time is enabled for a content group, the count can be greater than 1.
Hits	The number of cache hits for this object.
Misses	The number of cache misses for this object.
Compression format	The type of compression applied to this object. Compression formats include gzip, deflate, compress, and pack200-gzip.
HTTP version in response	The version of HTTP that was used to send the response.
Weak etag present in response	Strong etag headers change if the bits of an entity change. Strong headers are based on the octet values of an object. Weak etag headers change if the meaning of an entity changes. Weak etag values are based on semantic identity. Weak etags values start with a "W."
Negative marker cell	A marker object is cacheable, but it does not yet meet all the criteria for being cached. For example, the object may exceed the maximum response size for the content group. A marker cell is created for objects of this type. The next time a user sends a request for this object, a cache miss is served.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Properties	Specifies
Reason marker created	The reason a marker cell was created (for example, “Waiting for minhit,” “Content-length response data is not in group size limit”).
Auto poll every time	If the integrated cache receives an already expired 200 OK response with validators (either the Last-Modified or the ETag response headers) it stores the response and marks it as Auto-PET (automatically poll every time).
NetScaler Etag inserted in response	A variation of the ETag header generated by the NetScaler appliance. A value of YES appears if the NetScaler inserts an Etag in the response.
Full response present in cache	Indicates whether this is a complete response.
Destination IP verified by DNS	Indicates whether DNS resolution was performed when storing the object.
Object stored through a cache forward proxy	Indicates whether this response was stored due to a forward proxy that is configured in the integrated cache.
Object is a Delta basefile	A response that is delta-compressed.
Waiting for minhits	Indicates whether this content group requires a minimum number of origin server hits before caching a response.
Minhit count	If this content group requires a minimum number of origin server hits before caching an object, this field displays a count of the number of hits received so far.
HTTP Request Method	The method, GET or POST, used in the request that obtained this object.
Stored by policy	The name of the caching policy that caused this object to be stored. A value of NOT AVAILABLE indicates that the policy has been deactivated or deleted. A value of NONE indicates that the object did not match a visible policy, but was stored according to internal criteria for caching.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Properties	Specifies
Application firewall metadata exists	This parameter is used when the application firewall and the integrated cache are both enabled. The application firewall analyzes the contents of a response page, stores its metadata (for example, URLs and forms contained in page), and exports the metadata with the response to the cache. The cache stores the page and the metadata, and when the cache serves the page, it sends the metadata back to the request's session.
HTTP callout object, name, type, response	These cells indicate whether this data was stored as a result of an HTTP Callout expression, and provide information about various aspects of the callout and the corresponding response. For more information about HTTP callouts, see "HTTP Callouts".

To view cached objects by using the configuration utility

Navigate to Optimization > Integrated Caching > Cache Objects. You can view all the cached objects and sort them accordingly as per your requirement.

Finding Particular Cached Responses

Updated: 2014-08-08

You can find individual items in the cache based on search criteria. There are different methods for finding cached items, depending on whether the content group that contains the data uses hit and invalidation selectors, as follows:

- If the content group uses selectors, you can only conduct the search using the Locator ID for the cached item.
- If the content group does not use selectors, you conduct the search using criteria such as URL, host, content group name, and so on.

When searching for a cached response, you can locate some items by URL and host. If the response is in a content group that uses a selector, you can find it only by using a Locator number (for example, 0x00000000ad7af00000050). To save a Locator number for later use, right-click the entry and select **Copy**. For more information about selectors, see "Configuring Selectors and Basic Content Groups."

To display cached responses in content groups that do not have a selector by using the command line interface

At the command prompt, type:

show cache object [-locator <positiveInteger>] | [(-url <URL> (-host <hostName> [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET | POST])) | [-httpStatus<positive integer>] | -group <contentGroupName> | -ignoreMarkerObjects (ON | OFF) | -includeNotReadyObjects (ON | OFF)]

To display cached responses in content groups that have a selector by using the command line interface

At the command prompt, type:

show cache object -locator <locatorString> MarkerObjects (ON | OFF) | -includeNotReadyObjects (ON | OFF) | [-httpStatus<positive integer>]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To display cached responses in content groups that do not have a selector by using the configuration utility

Navigate to Optimization > Integrated Caching > Cache Objects, click Search, and set the search criteria to view the required cached response.
If you have not yet configured any content groups, all of the objects are in the Default group.

To display cached responses in content groups that have a selector by using the configuration utility

Navigate to Optimization > Integrated Caching > Cache Objects, click Search, and set the selector search criteria to view the required cached response.

Viewing Cache Statistics

Updated: 2013-10-28

The following table summarizes the detailed cache statistics that you can view.

Table 2. Integrated Cache Statistics

Counter	Specifies
Hits	Responses that are found in and served from the integrated cache. Includes static objects such as image files, pages with status codes 200, 203, 300, 301, 302, 304, 307, 403, 404, 410, and responses that match a user-defined policy with a CACHE action..
Misses	Intercepted HTTP requests where the response was ultimately fetched from origin server.
Requests	Total cache hits plus total cache misses.
Non-304 hits	If the user requests an item more than once, and the item in the cache is unchanged since the last time the NetScaler appliance served it, the NetScaler appliance serves a 304 response instead of the cached object. This statistic indicates how many items the NetScaler appliance served from the cache, excluding 304 responses.
304 hits	Number of 304 (object not modified) responses the NetScaler appliance served from the cache.
304 hit ratio (%)	Percentage of 304 responses that the NetScaler appliance served, relative to other responses.
Hit ratio (%)	Percentage of responses that the NetScaler appliance served from the cache (cache hits) relative to responses that could not be served from the cache.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Counter	Specifies
Origin bandwidth saved (%)	An estimate of the processing capacity that the NetScaler appliance saved on the origin server due to serving responses from the cache.
Bytes served by the NetScaler	Total number of bytes that the NetScaler appliance served from the origin server and the cache.
Bytes served by cache	Total number of bytes that the NetScaler appliance served from the cache.
Byte hit ratio(%)	Percentage of data that the NetScaler appliance served from the cache, relative to all of the data in all served responses.
Compressed bytes from cache	Amount of data, in bytes, that the NetScaler appliance served in compressed form.
Storable misses	If the NetScaler appliance does not find a requested object in the cache, it fetches the object from the origin server. This is known as a cache miss. A storable cache miss can be stored in the cache.
Non-storable misses	A non-storable cache miss cannot be stored in the cache.
Misses	All cache misses.
Revalidations	Max-Age setting in a Cache-Control header determines, in number of seconds, when an intervening cache must revalidate the content with the integrated cache before serving it to the user. For more information, see "Inserting a Cache-Control Header."
Successful revalidations	Number of re-validations that have been performed. For more information, see "Inserting a Cache-Control Header."
Conversions to conditional req	A user-agent request for a cached PET object is always converted to a conditional request and sent to the origin server. For more information, see "Polling the Origin Server Every Time a Request Is Received."
Storable miss ratio (%)	Storable cache misses as a percentage of non-storable cache misses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Counter	Specifies
Successful reval ratio (%)	Successful revalidations as a percentage of all revalidation attempts. For more information, see "Inserting a Cache-Control Header."
Expire at last byte	Number of times that the cache expired content immediately after receiving the last body byte. Only applicable to positive responses, as described in the table "Cache Hits and Misses." For more information, see "Example of Performance Optimization."
Flashcache misses	If you enable Flash Cache, the cache allows only one request to reach the server, eliminating flash crowds. This statistic indicates the number of Flash Cache requests that were cache misses. For more information, "Queuing Requests to the Cache."
Flashcache hits	Number of Flash Cache requests that were cache hits. For more information, see "Queuing Requests to the Cache."
Parameterized inval requests	Requests that match a policy with an invalidation (INVAL) action and a content group that uses an invalidation selector or parameters to selectively expire cached objects in the group.
Full inval requests	Requests that match an invalidation policy where the invalGroups parameter is configured and expires one or more content groups.
Inval requests	Requests that match an invalidation policy and result in expiration of specific cached responses or entire content groups.
Parameterized requests	Number of cache requests that were processed using a policy with a parameterized content group.
Parameterized non-304 hits	Number of cache requests that were processed using a policy with a parameterized content group, where full cached response was found, and the response was not a 304 (object not updated) response.
Parameterized 304 hits	Number of cache requests that were processed using a policy with a parameterized content group, where the cached object was found, and the object was a 304 (object not updated) response.
Total parameterized hits	Number of cache requests that were processed using a policy with a parameterized content group, where the cached object was found.
Parameterized	Percentage of 304 (object not updated) responses that were found using a parameterized policy,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Counter	Specifies
304 hit ratio (%)	relative to all cache hits.
Poll every time requests	If Poll Every Time is enabled, the NetScaler appliance always consults the origin server before serving a stored object. For more information, see "Polling the Origin Server Every Time a Request Is Received."
Poll every time hits	Number of times a cache hit was found using the Poll Every Time method. For more information, see "Polling the Origin Server Every Time a Request Is Received."
Poll every time hit ratio (%)	Percentage of cache hits using the Poll Every Time method, relative to all searches for cached objects using Poll Every Time. For more information, see "Polling the Origin Server Every Time a Request Is Received."
Maximum memory (KB)	Maximum amount of memory in the NetScaler appliance that is allocated to the cache. For more information, see "Configuring Global Attributes for Caching."
Maximum memory active value (KB)	Maximum amount of memory (active value) that will be set after the memory is actually allocated to the cache. For more information, see "How to Configure the Integrated Caching Feature of a NetScaler Appliance for various Scenarios."
Utilized memory (KB)	Amount of memory that is actually being used.
Memory allocation failures	Number of failed attempts to utilize memory for the purpose of storing a response in the cache.
Largest response so far	Largest response in bytes found in either the cache or the origin server and sent to the client.
Cached objects	Number of objects in the cache, including responses that have not yet been fully downloaded and responses that have been expired but not yet flushed.
Marker objects	Marker objects are created when a response exceeds the maximum or minimum response size for the content group, or has not yet received the minimum number of hits for the content group.
Hits being	Number of hits that have been served from the cache.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Counter	Specifies
Misses being handled	Responses that were fetched from the origin server, stored in the cache, and then served. Should approximate the number for storable misses. Does not include non-storable misses.

To view summary cache statistics by using the command line interface

At the command prompt, type:

```
stat cache
```

To view specific cache statistics by using the command line interface

At the command prompt, type:

```
stat cache -detail
```

View Stat Cache Detail COPY

```
> stat cache -detail

Integrated Cache Statistics - Detail

Integrated Cache Statistics - Summary

                        Rate (/s)        Total

Hits                        0                0

Misses                      0                0

Requests                    0                0

Hit ratio(%)               --                0

Origin bandwidth saved(%)          --                0

Cached objects              --                0

Marker objects              --                0

                        Rate (/s)        Total
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Requests                                0              0


Hit Statistics

                                Rate (/s)          Total

Non-304 hits                            0              0

304 hits                                0              0

Sql hits                                0              0

Hits                                    0              0

304 hit ratio(%)                       --              0

Hit ratio(%)                           --              0

Origin bandwidth saved(%)              --              0


Byte Statistics

                                Rate (/s)          Total

Bytes served by NetScaler             648       55379204

Bytes served by cache                   0              0

Byte hit ratio(%)                      --              0

Compressed bytes from cache             0              0


Miss Statistics
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
                                Rate (/s)            Total

Storable misses                          0                0

Non-storable misses                      0                0

Misses                                 0                0

Revalidations                          0                0

Successful revalidations                 0                0

Conversions to conditional req                 0                0

Storable miss ratio(%)                  --                0

Successful reval ratio(%)                --                0



Flashcache Statistics

                                Rate (/s)            Total

Expire at last byte                      0                0

Flashcache misses                        0                0

Flashcache hits                          0                0



Invalidation Statistics

                                Rate (/s)            Total

Parameterized inval requests                   0                0

Full inval requests                      0                0
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Inval requests                    0          0



Parameterized Caching Statistics

                          Rate (/s)        Total

Parameterized requests              0          0

Parameterized non-304 hits          0          0

Parameterized 304 hits              0          0

Total parameterized hits            0          0

Parameterized 304 hit ratio(%)     --          0



Poll Every Time (PET) Statistics

                          Rate (/s)        Total

Poll every time requests            0          0

Poll every time hits                0          0

Poll every time hit ratio(%)       --          0



Memory Usage Statistics

                          Total

Maximum memory(KB)                  0

Maximum memory active value(KB)     0
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Utilized memory(KB)                    0

Memory allocation failures             0

Largest response so far(B)             0

Cached objects                         0

Marker objects                         0



Hits being served                      0

Misses being handled                   0

 Done
```

To view summary cache statistics by using the configuration utility

1. Click the Dashboard tab at the top of the page.
2. Scroll down to the Integrated Caching section of the window.
3. To see detailed statistics, click the More... link at the bottom of the table.

To view specific cache statistics by using the configuration utility

1. Click the Reporting tab at the top of the page.
2. Under Built-In Reports, expand Integrated Cache, and then click the report with the statistics you want to view.
3. To save the report as a template, click Save As and name the report. The saved report appears underCustom Reports.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Improving Cache Performance

May 19, 2015

You can improve the performance of integrated cache, including handling simultaneous requests for the same cached data, avoiding delays that are associated with refreshing cached responses from the origin server, and ensuring that a response is requested often enough to be worth caching.

This section includes the following details:

- Reducing Flash Crowds
- Caching a Response after a Client Halts a Download
- Requiring a Minimum Number of Server Hits before Caching
- Example of Performance Optimization

Reducing Flash Crowds

Updated: 2015-05-20

Flash crowds occur when many users simultaneously request the same data. All of the requests in a flash crowd can become cache misses if you configured the cache to serve hits only after the entire object is downloaded.

The following techniques can reduce or eliminate flash crowds:

- **PREFETCH**: Refreshes a positive response before it expires to ensure that it never becomes stale or inactive.
  For more information, see "Refreshing a Response Prior to Expiration" section.

- **Cache buffering**: Starts serving a response to multiple clients as soon as it receives the response header from the origin server, rather than waiting for the entire response to be downloaded.
  The only limit on the number of clients that can download a response simultaneously is the available system resources.

  The Citrix NetScaler appliance downloads and serves responses even if the client that initiated the download halts before the download is complete. If the size of the response exceeds the cache size or if the response is chunked, the cache stops storing the response, but service to the clients is not disrupted.

- **Flash Cache**: Flash Cache queues requests to the cache, and allows only one request to reach the server at a time.
  For more information, see "Queuing Requests to the Cache" section.

Refreshing a Response Before Expiration

To ensure that a cached response is fresh whenever it is needed, the PREFETCH option refreshes a response before its calculated expiration time. The prefetch interval is calculated after receiving the first client request. From that point onward, the NetScaler appliance refreshes the cached response at a time interval that you configure in the PREFETCH parameter.

This setting is useful for data that is updated frequently between requests. It does not apply to negative responses (for example, 404 messages).

To configure prefetch for a content group by using the command line interface

At the command prompt, type:

set cache contentgroup <name> -prefetch YES [-prefetchPeriod <seconds> | -prefetchPeriodMilliSec <milliseconds>] [-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

prefetchMaxPending <positiveInteger>]
To configure prefetch for a content group by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Content Groups, and select the content group.
2. On Others tab, in the Flash Crowd and Prefetch group, select Prefetch option, and specify the values in Interval and Maximum number of pending prefetches text boxes.

Queuing Requests to the Cache

The Flash Cache option queues requests that arrive simultaneously (a flash crowd), retrieves the response, and distributes it to all the clients whose requests are in the queue. If, during this process, the response becomes non-cacheable, the NetScaler appliance stops serving the response from the cache and instead serves the origin server's response to the queued clients. If the response is not available, the clients receive an error message.

Flash Cache is disabled by default. You cannot enable Poll Every Time (PET) and Flash Cache on the same content group.

One disadvantage of Flash Cache is if the server replies with an error (for example, a 404 that is quickly remedied), the error is fanned out to the waiting clients.

Note: If Flash Cache is enabled, in some situations the NetScaler appliance is unable to correctly match the Accept-Encoding header in the client request with the Content-Encoding header in the response. The NetScaler appliance can assume that these headers match and mistakenly serve a hit. As a work-around, you can configure Integrated Caching policies to disallow serving hits to clients that do not have an appropriate Accept-Encoding header.
To enable Flash Cache by using the command line interface

At the command prompt, type:

set cache contentgroup <contentGroupName> -flashcache yes
To enable Flash Cache by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Content Groups, and select the content group.
2. On Others tab, in the Flash Crowd and Prefetch group, select Prefetch option.

Caching a Response after a Client Halts a Download

Updated: 2014-08-08

You can set the Quick Abort parameter to continue caching a response, even if the client halts a request before the response is in the cache.

If the downloaded response size is less than or equal to the Quick Abort size, the NetScaler appliance stops downloading the response. If you set the Quick Abort parameter to 0, all downloads are halted.

To configure quick abort size by using the command line interface

At the command prompt, type:

set cache contentgroup <name> -quickAbortSize <integerInKBytes>

To configure quick abort size by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Content Groups, and select the content group.
2. On Memory tab, set the relevant value in Quick Abort: Continue caching if more than text box.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Requiring a Minimum Number of Server Hits before Caching

Updated: 2015-05-19

You can configure the minimum number of times that a response must be found on the origin server before it can be cached. You should consider increasing the minimum hits if the cache memory fills up quickly and has a lower-than-expected hit ratio.

The default value for the minimum number of hits is 0. This value caches the response after the first request.

To configure the minimum number of hits that are required before caching by using the command line interface

At the command prompt, type:

set cache contentgroup <name> -minhits <positiveInteger>

To configure the minimum number of hits that are required before caching by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Content Groups, and select the content group.
2. On Memory tab, set the relevant value in Do not cache, if hits are less than text box.

Example of Performance Optimization

Updated: 2013-10-28

In this example, a client accesses a stock quote. Stock quotes are highly dynamic. You configure the integrated cache to serve the same stock quote to concurrent clients without sending multiple requests to the origin server. The stock quote expires after it is downloaded to all of the clients, and the next request for a quote for the same stock is fetched from the origin server. This ensures that the quote is always up to date.

The following task overview describes the steps to configure the cache for the stock quote application.

Task overview: Configuring caching for a stock quote application

1. Create a content group for stock quotes.
   For more information, see "About Content Groups."

   Configure the following for this content group:

   - On the Expiry Method tab, select the Expire after complete response received check box.
   - On the Others tab, select the Flash Cache check box, and click Create.
2. Add a cache policy to cache the stock quotes.
   For more information, see "Configuring a Policy in the Integrated Cache."

   Configure the following for the policy:

   - In the Action and Store in Group lists, select CACHE and select the group that you defined in the previous step.
   - Click Add, and in the Add Expression dialog box configure an expression that identifies stock quote requests, for example:
     http.req.url.contains("cgi-bin/stock-quote.pl")

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

3. Activate the policy.
   For more information, see "Globally Binding an Integrated Caching Policy." In this example, you bind this policy to request-time override processing and set the priority to a low value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Cookies, Headers, and Polling

May 20, 2015

This section describes the procedures to configure how the cache manages cookies, HTTP headers, and origin server polling, including modifying default behavior that causes the cache to diverge from documented standards, overriding HTTP headers that might cause cacheable content to not be stored in the cache, and configuring the cache to always poll the origin for updated content under specialized circumstances.

For details, see the following sections:

- Divergence of Cache Behavior from the Standards
- Removing Cookies from a Response
- Inserting HTTP Headers at Response Time
- Ignoring Cache-Control and Pragma Headers in Requests
- Polling the Origin Server Every Time a Request Is Received
- PET and Client-Specific Content
- PET and Authentication, Authorization, and Auditing

Divergence of Cache Behavior from the Standards

Updated: 2013-10-28

By default, the integrated cache conforms to the following standards:

- RFC 2616, “Hypertext Transfer Protocol HTTP/1.1”
- The caching behaviors described in RFC 2617, “HTTP Authentication: Basic and Digest Access Authentication”
- The caching behavior described in RFC 2965, “HTTP State Management Mechanism”

The built-in policies and the Default content group attributes ensure conformance with most of these standards.

The default integrated cache behavior diverges from the specifications as follows:

- There is limited support for the Vary header.
  By default, any response containing a Vary header is considered to be non-cacheable unless it is compressed. A compressed response contains Content-Encoding: gzip, Content-Encoding: deflate, or Content-Encoding: pack200-gzip and is cacheable even if it contains the Vary: Accept-Encoding header.

- The integrated cache ignores the values of the headers Cache-Control: no-cache and Cache-Control: private.
  For example, a response that contains Cache-Control: no-cache=”Set-Cookie” is treated as if the response contained Cache-Control: no-cache. By default, the response is not cached.

- An image (Content-Type = image/*) is always considered cacheable even if an image response contains Set-Cookie or Set-Cookie2 headers, or if an image request contains a Cookie header.
  The integrated cache removes Set-Cookie and Set-Cookie2 headers from a response before caching it. This diverges from RFC 2965. You can configure RFC-compliant behavior as follows:

  ```
  add cache policy rfc_compliant_images_policy -rule "http.res.header.set-cookie2.exists || http.res.header.set-cookie.exists" -action NOCACHE
  bind cache global rfc_compliant_images_policy -priority 100 -type REQ_OVERRIDE
  ```
- The following Cache-Control headers in a request force an RFC-compliant cache to reload a cached response from the origin server:

  ```
  Cache-control: max-age=0
  Cache-control: no-cache
  ```
  To guard against Denial of Service attacks, this behavior is not the default. For more information, see "Inserting a Cache-Control Header" section.

- By default, the caching module considers a response to be cacheable unless a response header states otherwise.
  To make this behavior RFC 2616 compliant, set -weakPosRelExpiry and -weakNegResExpiry to 0 for all content groups.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Removing Cookies from a Response

Updated: 2014-08-12

Cookies are often personalized for a user, and typically should not be cached. The Remove Response Cookies parameter removes Set-Cookie and Set-Cookie2 headers before caching a response. By default, the Remove Response Cookies option for a content group prevents caching of responses with Set-Cookie or Set-Cookie2 headers.

Note that when images are cached, the built-in behavior is to remove the Set-Cookie and Set-Cookie2 headers before caching, no matter how the content group is configured.

Note: Citrix recommends that you accept the default Remove Response Cookies for every content group that stores embedded responses, for example, images.

To configure Remove Response Cookies for a content group by using the command line interface

At the command prompt, type:

set cache contentgroup <name> -removeCookies YES

To configure Remove Response Cookies for a content group by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Content Groups, and select the content group.
2. On Others tab, in the Settings group, select Remove response cookies option.

Inserting HTTP Headers at Response Time

Updated: 2014-08-12

The integrated cache can insert HTTP headers in responses that result from cache hits. The Citrix® NetScaler® appliance does not alter headers in responses that result from cache misses.

The following table describes headers that you can insert in a response.

Table 1. Different HTTP Headers You Can Insert in a Response That Is Served from the Cache

Header	Specifies
Age	Provides the age of the response in seconds, calculated from the time the response was generated at the origin server. By default, the cache inserts an Age header for every response that is served from the cache.
Via	Lists protocols and recipients between the start and end points for a request or a response. The NetScaler appliance inserts a Via header in every response that it serves from the cache. The default value of the inserted header is “NS-CACHE-9.2:last octet of the NetScaler IP address.” For more information, see "Configuring Global Attributes for Caching."
ETag	The cache supports response validation using Last-Modified and ETag headers to determine if a response is stale. The cache inserts an ETag in a response only if it caches the response and the origin server has not inserted its own ETag header. The ETag value is an arbitrary unique number. The ETag value for a response changes if it is refreshed from the origin server, but it stays the same if the server sends a 304 (object not updated) response. Origin servers typically do not generate validators for dynamic content because dynamic content is considered non-cacheable. You can override this behavior. With ETag header insertion, the cache is permitted to not serve full responses. Instead, the user agent is required to cache the dynamic response sent by the integrated cache the first time. To force a user agent to cache a

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Header	Specifies
	response, you configure the integrated cache to insert an ETag header and replace the origin-provided Cache-Control header.
Cache-Control	The NetScaler appliance typically does not modify cacheability headers in responses that is serves from the origin server. If the origin server sends a response that is labeled as non-cacheable, the client treats the response as non-cacheable even if the NetScaler appliance caches the response. To cache dynamic responses in a user agent, you can replace Cache-Control headers from the origin server. This applies only to user agents and other intervening caches. They do not affect the integrated cache. For more information, see "Inserting a Cache-Control Header" section.

Inserting an Age, Via, or ETag Header

The following procedures describe how to insert Age, Via, and ETag headers.

To insert an Age, Via, or Etag header by using the command line interface

At the command prompt, type:

set cache contentgroup <name> -insertVia YES -insertAge YES -insertETag YES
To insert an Age, Via, or Etag header by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Content Groups, and select the content group.
2. On Others tab, in the HTTP Header Insertions group, select the Via, Age, or ETag options, as appropriate.
   The values for the other header types are calculated automatically. Note that you configure the Via value in the main settings for the cache.

Inserting a Cache-Control Header

When the integrated cache replaces a Cache-Control header that the origin server inserted, it also replaces the Expires header. The new Expires header contains an expiration time in the past. This ensures that HTTP/1.0 clients and caches (that do not understand the Cache-Control header) do not cache the content.

To insert a Cache-Control header by using the command line interface

At the command prompt, type:

set cache contentgroup <name> -cacheControl <value>
To insert a Cache-Control header by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Content Groups, and
   1. Click Expiry Method tab, clear the heuristic and default expiry settings and set the relevant value in Expire content after text box.
   2. Click Others tab and type the header you want to insert in the Cache-Control text box. Alternatively, click Configure to set the Cache-Control directives in cached responses.

Ignoring Cache-Control and Pragma Headers in Requests

Updated: 2014-08-12

By default, the caching module processes Cache-Control and Pragma headers. The following tokens in Cache-Control headers are processed as described in RFC 2616.

- max-age
- max-stale
- only-if-cached
- no-cache

A Pragma: no-cache header in a request is treated in the same way as a Cache-Control: no-cache header.

If you configure the caching module to ignore Cache-Control and Pragma headers, a request that contains a Cache-Control: No-Cache header

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

causes the NetScaler appliance to retrieve the response from the origin server, but the cached response is not updated. If the caching module processes Cache-Control and Pragma headers, the cached response is refreshed.

The following table summarizes the implications of various settings for these headers and the Ignore Browser's Reload Request setting.

Table 2. Outcome of Settings for Ignoring Reload Requests, Cache-Control, and Pragma Headers

Setting for Ignore Cache-Control and Pragma Headers	Setting for Ignore Browser's Reload Request	Outcome
Yes	Yes or No	Ignore the Cache-Control and Pragma headers from the client, including the Cache-Control: no-cache directive.
No	Yes	The Cache-Control: no-cache header produces a cache miss, but a response that is already in the cache is not refreshed.
No	No	A request that contains a Cache-Control: no-cache header causes a cache miss and the stored response is refreshed.

To ignore Cache-Control and Pragma headers in a request by using the command line interface

At the command prompt, type:

set cache contentgroup <name> -ignoreReqCachingHdrs YES

To ignore browser reload requests by using the command line interface

At the command prompt, type:

set cache contentgroup <name> -ignoreReloadReq NO
Note that by default, the -ignoreReloadReq parameter is set to YES.

To ignore Cache-Control and Pragma headers in a request by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Content Groups, and select the content group.
2. On Others tab, in the Settings group, select Ignore Cache-control and Pragma Headers in Requests option.

Example of a Policy to Ignore Cache-Control Headers

In the following example, you configure a request-time override policy to cache responses that contain Content-type: image/* regardless of the Cache-Control header in the response.

To configure a request-time override policy to cache all responses with image/*

1. Flush the cache using the Invalidate All option.

   For more information, see "Flushing Responses in a Content Group."

2. Configure a new cache policy, and direct the policy to a particular content group. For more information, see "Configuring a Policy in the Integrated Cache."
3. Ensure the content group that the policy uses is configured to ignore Cache-Control headers, as described in "Ignoring Cache-Control and Pragma Headers in Requests."
4. Bind the policy to the request-time override policy bank.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For more information, see "Globally Binding an Integrated Caching Policy."

Polling the Origin Server Every Time a Request Is Received

Updated: 2014-08-12

You can configure the NetScaler appliance to always consult the origin server before serving a stored response. This is known as Poll Every Time (PET). When the NetScaler appliance consults the origin server and the PET response has not expired, a full response from the origin server does not overwrite cached content. This property is useful when serving client-specific content.

After a PET response expires, the NetScaler appliance refreshes it when the first full response arrives from the origin server.

The Poll Every Time (PET) function works as follows:

- For a cached response that has validators in the form of an ETag or a Last-Modified header, if the response expires it is automatically marked PET and cached.
- You can configure PET for a content group.
  If you configure a content group as PET, every response in the content group is marked PET. The PET content group can store responses that do not have validators. Responses that are automatically marked PET are always expired. Responses that belong to a PET content group can expire after a delay, based on how you configure the content group.

Two types of requests are affected by polling:

- **Conditional Requests**: A client issues a conditional request to ensure that the response that it has is the most recent copy.
  A user-agent request for a cached PET response is always converted to a conditional request and sent to the origin server. A conditional request has validators in If-Modified-Since or If-None-Match headers. The If-Modified-Since header contains the time from the Last-Modified header. An If-None-Match header contains the response's ETag header value.

  If the client's copy of the response is fresh, the origin server replies with 304 Not Modified. If the copy is stale, a conditional response generates a 200 OK that contains the entire response.

- **Non-Conditional Requests**: A non-conditional request can only generate a 200 OK that contains the entire response.

The following table summarizes response types based on the origin server's response

Table 3. How Responses Are Affected by Poll Every Time

Origin Server Response	Action
Send the full response	The origin server sends the response as-is to the client. If the cached response has expired, it is refreshed.
304 Not Modified	The following header values in the 304 response are merged with the cached response and the cached response is served to the client: • Date • Expires • Age • Cache-Control header Max-Age and S-Maxage tokens
401 Unauthorized 400 Bad Request 405 Method Not Allowed 406 Not Acceptable	The origin's response is served as-is to the client. The cached response is not changed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Origin Server Response	Action
407 Proxy Authentication Required	
Any other error response, for example, 404 Not Found	The origin's response is served as-is to the client. The cached response is removed.

Note: The Poll Every Time parameter treats the affected responses as non-storable.

To configure poll every time by using the command line interface

At the command prompt, type:

add cache contentgroup <contentGroupName> -pollEveryTime YES

To configure poll every time by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Content Groups, and select the content group.
2. On Others tab, in the Settings group, select Poll every time (validate cached content with origin for every request) option.

PET and Client-Specific Content

The PET function can ensure that content is customized for a client. For example, a Web site that serves content in multiple languages examines the Accept-Language request header to select the language for the content that it is serving. For a multi-language Web site where English is the predominant language, all English language content can be cached in a PET content group. This ensures that every request goes to the origin server to determine the language for the response. If the response is English, and the content has not changed, the origin server can serve a 304 Not Modified to the cache.

The following example shows commands to cache English responses in a PET content group, configure a named expression that identifies English responses in the cache, and configure a policy that uses this content group and named expression. Bold is used for emphasis:

```
add cache contentgroup EnglishLanguageGroup -pollEveryTime YES
add expression containsENExpression –rule "http.res.header(\"Content-Language\").contains(\"en\")"
add cache policy englishPolicy -rule containsENExpression -action CACHE -storeInGroup englishLanguageGroup
bind cache policy englishPolicy -priority 100 -precedeDefRules NO
```

PET and Authentication, Authorization, and Auditing

Outlook Web Access (OWA) is a good example of dynamically generated content that benefits from PET. All mail responses (*.EML objects) have an ETag validator that enables them to be stored as PET responses.

Every request for a mail response travels to the origin server, even if the response is cached. The origin server determines whether the requestor is authenticated and authorized. It also verifies that the response exists in the origin server. If all results are positive, the origin server sends a 304 Not Modified response.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Integrated Cache as a Forward Proxy

Aug 08, 2014

The integrated cache can service as a forward proxy device that passes requests to other NetScaler appliances or to other types of cache servers. You configure the integrated cache as a forward proxy by identifying the IP addresses of the cache server or servers. After configuring the forward proxy, the NetScaler appliance sends requests that contain the configured IP address on to the cache server instead of involving the integrated cache.

To configure the NetScaler as a forward cache proxy by using the command line interface

At the command prompt, type:

```
add cache forwardProxy <IPAddress> <port>
```

To configure the NetScaler as a forward cache proxy by using the configuration utility

1. Navigate to Optimization > Integrated Caching > Forward Proxy, and add a forward proxy by specifying the IP address and port number.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Default Settings for the Integrated Cache

Aug 26, 2013

The Citrix NetScaler integrated cache feature provides built-in policies with default settings as well as initial settings for the Default content group. The information in this section defines the parameters for the built-in policies and Default content group.

Default Caching Policies

Updated: 2013-08-26

The integrated cache has built-in policies. The NetScaler appliance evaluates the policies in a particular order, as discussed in the following sections.

You can override these built-in policies with a user-defined policy that is bound to a request-time override or response-time override policy bank.

Note that if you configured policies prior to release 9.0 and specified the -precedeDefRules parameter when binding the policies, they are automatically assigned to override-time bind points during migration.

Viewing the Default Policies

The built-in policy names start with an underscore (_). You can view the built-in policies from the command line and the administrative console using the show cache policy command.

Default Request Policies

You can override the following built-in request time policies by configuring new policies and binding them to the request-time override processing point. In the following policies, note that the MAY_NOCACHE action stipulates that the transaction is cached only when there is a user-configured or built-in CACHE directive at response time.

The following policies are bound to the _reqBuiltinDefaults policy label. They are listed in priority order.

1. Do not cache a response for a request that uses any method other than GET.
   The policy name is **_nonGetReq**. The following is the policy rule:

   !HTTP.REQ.METHOD.eq(GET)

2. Set a NOCACHE action for a request with header value that contains If-Match or If-Unmodified-Since.
   The policy name is **_advancedConditionalReq**. The following is the policy rule:

   HTTP.REQ.HEADER("If-Match").EXISTS || HTTP.REQ.HEADER("If-Unmodified-Since").EXISTS

3. Set a MAY_NOCACHE action for a request with the following header values: Cookie, Authorization, Proxy-authorization or a request which contains the NTLM or Negotiate header.
   The policy name is **_personalizedReq**. The following is the policy rule:

   HTTP.REQ.HEADER("Cookie").EXISTS || HTTP.REQ.HEADER("Authorization").EXISTS || HTTP.REQ.HEADER("Proxy-Authorization").EXISTS || HTTP.REQ.IS_NTLM_OR_NEGOTIATE

Default Response Policies

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You can override the following default response-time policies by configuring new policies and binding them to the response-time override processing point.

The following policies are bound to the **_resBuiltinDefaults** policy label and are evaluated in the order in which they are listed:

1. Do not cache HTTP responses unless they are of type 200, 304, 307, 203 or if the types are between 400 and 499 or between 300 and 302.
   The policy name is **_uncacheableStatusRes**. The following is the policy rule:

   !((HTTP.RES.STATUS.EQ(200)) || (HTTP.RES.STATUS.EQ(304)) || (HTTP.RES.STATUS.BETWEEN(400,499)) || (HTTP.RES.STATUS.BETWEEN(300, 302)) || (HTTP.RES.STATUS.EQ(307))|| (HTTP.RES.STATUS.EQ(203)))

2. Do not cache an HTTP response if it has a Vary header with a value of anything other than Accept-Encoding.
   The compression module inserts the Vary: Accept_Encoding header. The name of this expression is **_uncacheableVaryRes**. The following is the policy rule:

   ((HTTP.RES.HEADER("Vary").EXISTS) && ((HTTP.RES.HEADER("Vary").INSTANCE(1).LENGTH > 0) || (!HTTP.RES.HEADER("Vary").STRIP_END_WS.SET_TEXT_MODE(IGNORECASE).eq("Accept-Encoding"))))

3. Do not cache a response if its Cache-Control header value is No-Cache, No-Store, or Private, or if the Cache-Control header is not valid.
   The policy name is **_uncacheableCacheControlRes**. The following is the policy rule:

   ((HTTP.RES.CACHE_CONTROL.IS_PRIVATE) || (HTTP.RES.CACHE_CONTROL.IS_NO_CACHE) || (HTTP.RES.CACHE_CONTROL.IS_NO_STORE) || (HTTP.RES.CACHE_CONTROL.IS_INVALID))

4. Cache responses if the Cache-Control header has one of the following values: Public, Must-Revalidate, Proxy-Revalidate, Max-Age, S-Maxage.
   The policy name is **_cacheableCacheControlRes**. The following is the policy rule:

   ((HTTP.RES.CACHE_CONTROL.IS_PUBLIC) || (HTTP.RES.CACHE_CONTROL.IS_MAX_AGE) || (HTTP.RES.CACHE_CONTROL.IS_MUST_REVALIDATE) || (HTTP.RES.CACHE_CONTROL.IS_PROXY_REVALIDATE) || (HTTP.RES.CACHE_CONTROL.IS_S_MAXAGE))

5. Do not cache responses that contain a Pragma header.
   The name of the policy is **_uncacheablePragmaRes**. The following is the policy rule:

   HTTP.RES.HEADER("Pragma").EXISTS

6. Cache responses that contain an Expires header.
   The name of the policy is **_cacheableExpiryRes**. The following is the policy rule:

   HTTP.RES.HEADER("Expires").EXISTS

7. If the response contains a Content-Type header with a value of Image, remove any cookies in the header and cache it.
   The name of the policy is **_imageRes**. The following is the policy rule:

   HTTP.RES.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).STARTSWITH("image/")

   You could configure the following content group to work with this policy:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add cache contentgroup nocookie_group -removeCookies YES

8. Do not cache a response that contains a Set-Cookie header.
   The name of the policy is **_ personalizedRes**. The following is the policy rule:

   HTTP.RES.HEADER("Set-Cookie").EXISTS || HTTP.RES.HEADER("Set-Cookie2").EXISTS

Restrictions on Default Policies

You cannot override the following built-in request time policies with user-defined policies.

These policies are listed in priority order.

1. Do not cache any responses if the corresponding HTTP request lacks a GET or POST method.
2. Do not cache any responses for a request if the HTTP request URL length plus host name exceeds 1744 byes.
3. Do not cache a response for a request that contains an If-Match header.
4. Do not cache a request that contains an If-Unmodified-Since header.
   Note that this is different from the If-Modified-Since header.
5. Do not cache a response if the server does not set an expiry header.

You cannot override the following built-in response time policies. These policies are evaluated in the order in which they are listed:

1. Do not cache responses that have an HTTP response status code of 201, 202, 204, 205, or 206.
2. Do not cache responses that have an HTTP response status code of 4xx, with the exceptions of status codes 403, 404, and 410.
3. Do not cache responses if the response type is FIN terminated, or the response does not have one of the following attributes: Content-Length, or Transfer-Encoding: Chunked.
4. Do not cache the response if the caching module cannot parse its Cache-Control header.

Initial Settings for the Default Content Group

When you first enable integrated caching, the NetScaler appliance provides one predefined content group named the Default content group. The following table shows the settings for this group.

Table 1. Predefined Settings for the Default Content Group

Parameter	Description	Default Value
Hit parameters	The hit parameters contain the parameter names that are significant for generating a response. In parameterized hit selection, NetScaler appliance matches the URL stem byte-for-byte, matches normalized values of the hit parameters, and matches the target service information.	none
Invalidation Parameters	These parameters mark a cached object as obsolete during parameterized selection. Specific objects, or all objects in a content group, are selected if the values of the	none

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter	Description	Default Value
	invalidation parameters in the object and in the request are same after normalization. The invalidation parameters are a subset of the hit parameters.	
Poll Every Time	Poll every time for the objects in this content group.	NO
Ignore reload request	Specifies whether a request can force the system to reload a cached object from the origin. To guard against Denial of Service attacks, you must set this flag to YES. To get RFC-compliant behavior you should set it to NO.	YES
Remove Response Cookies	If this option is disabled for a content group, and if the response contains cookies, the cookies are stored and served with every cache hit. By default, the remove cookies option is enabled for a content group, to prevent the integrated cache from storing any responses with Set-Cookie or Set-Cookie2 headers unless the response is an image.	YES
Prefetch	The Prefetch option refreshes an object when it is about to expire. This ensures that the object remains stale or inactive (and therefore it cannot be served) for a shorter duration of time.	YES
Prefetch period	This duration in seconds during which prefetch should be attempted, immediately before the object's calculated expiry time.	heuristic
Maximum outstanding prefetches	The number of items that can be subjected to a prefetch at a time.	4294967295
Flashcache	Determines whether to enable queuing of client requests and simultaneous distribution of responses to all clients in the queue.	NO
Expire at last byte	Determines whether to expire a cached response immediately after serving it.	NO
Insert Via header	Defines a string to be inserted in a Via header. By default, a Via header is inserted in all responses served from a content group. The Via header is not inserted for responses that are served by the origin server.	YES
Insert Age header	The Age header contains information about the age of the object in seconds as calculated by the integrated cache.	YES
Insert ETag header	With ETag header insertion, the integrated cache does not serve full responses on repeat requests. This is done by forcing the user agent to cache the dynamic response	YES

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Parameter	Description	Default Value
	sent by the cache the first time.	
Cache-control header	You can enable caching of dynamic objects in the user agent by replacing the Cache-Control headers that are inserted by the origin server. You must configure the new Cache-Control header to be inserted in the content group.	NONE
Quick abort size	If the size of an object that is being downloaded is less than or equal to the quick abort value, and a client aborts during the download, the cache stops downloading the response. If the object is larger than the quick abort size, the cache continues to download the response.	4194303 KBytes (maximum)
Minimum Response Size	You can control memory use by setting a minimum response size. Cached objects must be larger than the minimum response size.	0 KBytes
Maximum Response Size	You can control memory use by setting a maximum response size. Cached objects must be smaller than the maximum response size.	80 KBytes
Memory usage limit	Sets the maximum amount of memory that the cache can use. The effective limit is based on the available memory of the NetScaler appliance. The minimum value is 0 and the maximum value is unlimited.	UNLIMITED
Ignore caching headers in request	Disregards Cache-Control and Pragma headers in HTTP requests.	YES
MinHits configured	Number of hits that are required to qualify a response for storage in this content group.	0
Always evaluate policies		NO
Pinned	By default, when the cache is full the NetScaler appliance replaces the least recently used response first. The NetScaler appliance does not apply this behavior to content groups that are marked as pinned.	NO
Lazy DNS resolution	If set to YES, DNS resolution is performed for responses only if the destination IP address in the request does not match the destination IP address of the cached response.	YES

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Jul 22, 2013

If the integrated cache feature does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for Troubleshooting

Updated: 2013-07-22

For best results, use the following resources to troubleshoot an integrated cache issue on a NetScaler appliance:

- The relevant trace files
- The ns.conf file
- The RFC 2616 document
- A copy of the object, if possible

In addition to the above resources, the following tools expedite troubleshooting:

- The iehttpheaders or a similar utility
- The Wireshark application customized for the NetScaler trace files

Troubleshooting Integrated Caching Issues

Updated: 2013-08-02

The following are effective steps to troubleshoot the objects that are not cached:

1. **Verify that the Integrated Caching feature is enabled.**
   Run the following command to verify the feature is enabled:

   ```
   show ns feature
   ```

   Following is sample output of the above command:

   ```
   show ns feature
       Feature status:
                   Web Logging: OFF
              Surge Protection: OFF
               Load Balancing: ON
             Content Switching: ON
             Cache Redirection: OFF
                Sure Connect: OFF
            Compression Control: OFF
              Priority Queuing: OFF
               SSL Offloading: OFF
         Global Server Load Balancing: OFF
            Http DoS Protection: OFF
              Dynamic Routing: OFF
             Content Filtering: OFF
            Integrated Caching: ON
                   SSL VPN: OFF
   ```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
OSPF Routing: OFF
RIP Routing: OFF
BGP Routing: OFF
Done
```

The entry highlighted in boldface (for reference) in the above output indicates that the integrated caching feature is enabled. If the feature is not enabled, run the following command to enable it:

enable ns feature IC

2. **Make sure that sufficient memory is available on the NetScaler appliance.**
   Depending on the size of the object to be cached, memory available to store the cacheable object might be insufficient. You can set the memory limit for the integrated cache either globally or for individual content groups.

   Run the following command to verify the memory allocated to integrated cache globally:

   show cache parameter

   Following is sample output of the above command:

```
show cache parameter
    Integrated cache global configuration:
    Memory usage limit: 256 MBytes
    Via header: NS-CACHE-6.1: 101
    Verify cached object using: HOSTNAME_AND_IP
    Max POST body size to accumulate: 32768
    Current outstanding prefetches: 0
    Max outstanding prefetches: 4294967294
    Treat NOCACHE policies as BYPASS policies: YES
Done
```

   The entry highlighted in boldface (for reference) in the preceding output indicates the amount of memory allocated to the integrated cache globally.

   Run the following command to verify the memory allocated to an individual content group:

   show cache contentGroup <Content_Group_Name>

   Following is sample output of the above command:

```
show cache contentgroup content1
    Name: content1
    Heuristic expiry time parameter: 10 percent
    Weak relative expiry time - Positive responses: 3600 secs
    Weak relative expiry time - Negative responses: 600 secs
    Hit parameters: NONE
    Invalidation Parameters: NONE
    Invalidation restricted to host: NO
    Ignore parameter value case: NO
    Match Request Cookies: NO
    Poll Every Time: NO
    Ignore reload request: YES
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Remove Response Cookies: YES
    Prefetch: YES
    Prefetch period: heuristic
    Current outstanding prefetches: 0
    Max outstanding prefetches: 4294967294
    Flashcache: NO
    Expire at last byte: NO
    Insert Via header: YES
    Insert Age header: YES
    Insert ETag header: YES
    Cache-control header: NONE
    Quick abort size: 4194303 KBytes (MAXIMUM)
    Minimum Response Size: 0 KBytes
    Maximum Response Size: 80 KBytes
    Memory usage: 0 Bytes
    Memory usage limit: 64 MBytes
    Ignore caching headers in request: YES
    Non-304 hits: 0
    304 hits: 0
    Cached objects: 0
    Number of times expired/flushed: 1
    MinHits configured: 0
    Always evaluate policies: NO
    Pinned: NO
Done
```

The entry highlighted in boldface (for reference) in the above output indicates the amount of memory allocated to the content group.

3. **Verify that cacheable object is small enough to be stored within the configured memory limits.**

   Complete the following procedure to make space for the object to be cached:

   - Run the following command to flush the cache for the content group to which the object belongs:
     flush cache contentGroup <Content_Group_Name>
   - Verify that the object is cached. If the object is cached successfully, increase the memory allocated for the content group. Otherwise, run the following command to flush the cache globally:
     flush cache contentGroup ALL
   - Verify that the object is cached. If the object is cached successfully, consider increasing the global memory limit. If the object is still not cached, something else is causing the failure to cache the object.

   The memory allocated to the integrated cache depends on the NetScaler appliance model. You can allocate approximately half the available memory to the integrated cache. Similarly, the maximum about to memory you can allocate for a content group cannot be more than the memory allocated for global cache.

   To increase the global memory limit for the integrated cache, run the following command:

   set cache parameter -memLimit <Integer>

   To increase the memory limit for a content group, run the following command:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set cache contentgroup <contentgroup name> -memLimit <Integer>
```

4. **Verify that the cache policy is bound to an appropriate bind point, an appropriate priority is set for the policy, and an appropriateprecedeDefRulesswitch is configured.**
   You must activate a caching policy by binding it globally. To verify that the policy is active, run the following command:

   ```
   show cache global
   ```

   Following is sample output of the above command:

   ```
   show cache global
   1)    Name: red_pol   State: ACTIVE   Priority: 1
         Rule: URL CONTAINS red
         Action: NOCACHE
         Precede default HTTP rules: YES
         Hits : 10
    Done
   ```

   In the output, verify the following settings:
   - **The policy is bound**: The output should contain all the active cache policies. If the cache policy for the object to be cached is not listed in the output, the policy is not yet bound. Run the following command to bind the policy globally:
     bind cache global <Policy_Name> -priority <Integer> [-precedeDefRules YES|NO]
   - **The policy is Active**: If the policy is bound, verify that the state of the policy is displayed as Active. The entry indicating that the policy in the preceding output is active is the first highlighted entry in the sample output of the show cache global command, above. The policy is active if it is bound globally and an appropriate priority is set. Otherwise, the status of the policy is showed as Passive.
   - **An appropriate priority is assigned to the policy**: The first highlighted entry in the sample output above displays the priority of the policy. If the priority is not set, you can use the bind command to set the priority of the policy. Note that the higher the priority number, the lower the priority. The priority assigned to the policy enables the NetScaler appliance to determine the order in which the policy should be evaluated.
     If evaluation of a particular policy fails, increase the priority of the policy so that it is evaluated before other policies. Caching policies, due to their high granularity, can be very complicated to configure. Therefore, two policies might be contradictory. As a result, only the higher-priority policy takes effect.
   - **The precedeDefRules switch** setting is correct: The second highlighted entry in the sample output of the show cache global command, above, indicates the precedeDefRules switch setting. This setting enables the NetScaler appliance to determine whether the policy should be evaluated before the default built-in policies, which implement the standard HTTP caching behavior, such as basing caching decisions on HTTP header fields (for example, the If-Modified-Since and no-cache fields). You can set this switch when binding the policy.
     For certain types of HTTP(S) transactions, you might have to make sure that the policy precedes default HTTP rules, to force objects to be cached. Especially if requests include header fields, such as If-Modified-Since, or responses contain the No-Cache header field, you might have to make sure that the cache policy overrides the default in order for objects from these transactions to be cached. Force the policy to override default HTTP rules by rebinding the cache policy with the -precedeDefRules YES switch.

5. **Verify the size of the object to be cached.**
   You can configure a content group with minimum, which by default is 0 KB, and maximum, which by default is 80 KB, response sizes for the objects to be cached. The object does not get cached if its size is not within the configured range. Additionally, verify that the cache expiry times are set to an appropriate value. For example, check for a very small

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

time limit, such as one second.

Run the following command from the command line interface of the appliance to display the size limits and expiry times for a specific content group:

```
show cache contentGroup <Content_Group_Name>
```

Following is an example of this command's output:

```
show cache contentGroup content1
    Name: content1
    Heuristic expiry time parameter: 10 percent
    Weak relative expiry time - Positive responses: 3600 secs
    Weak relative expiry time - Negative responses: 600 secs
    Hit parameters: NONE
    Invalidation Parameters: NONE
    Invalidation restricted to host: NO
    Ignore parameter value case: NO
    Match Request Cookies: NO
    Poll Every Time: NO
    Ignore reload request: YES
    Remove Response Cookies: YES
    Prefetch: YES
    Prefetch period: heuristic
    Current outstanding prefetches: 0
    Max outstanding prefetches: 4294967294
    Flashcache: NO
    Expire at last byte: NO
    Insert Via header: YES
    Insert Age header: YES
    Insert ETag header: YES
    Cache-control header: NONE
    Quick abort size: 4194303 KBytes (MAXIMUM)
    Minimum Response Size: 0 KBytes
    Maximum Response Size: 80 KBytes
    Memory usage: 0 Bytes
    Memory usage limit: UNLIMITED
    Ignore caching headers in request: YES
    Non-304 hits: 0
    304 hits: 0
    Cached objects: 0
    Number of times expired/flushed: 0
    MinHits configured: 0
    Always evaluate policies: NO
    Pinned: NO
 Done
```

In addition to the above steps for troubleshooting integrated caching issues, you can consider using the following troubleshooting techniques:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Depending on the configuration of a policy, there are virtually an unlimited number of reasons for the policy not getting evaluated. If you have completed the preceding steps to troubleshoot the issue, consider completing the following procedure to troubleshoot the issue further:
  1. Flush the cache.
  2. Verify the value of the hit parameter for the policy by running the following command:
     show cache global
     1) Name: home_pol_1 State: ACTIVE Priority: 99
        Rule: URL CONTAINS home
        Action: NOCACHE
        Precede default HTTP rules: NO
        Hits : 29
  3. Send an HTTP request for the related object from a Web browser.
  4. Run the show cache global command again and verify that the value for the hit parameter has incremented.

  Depending on the policy receiving hits or not, you can determine whether the issue is due to the policy have not been configured correctly or to a more global cache setting.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Front End Optimization

Oct 10, 2015

> **Note**
>
> Front-end optimization is available if you have an Enterprise or Platinum NetScaler license and are running NetScaler release 10.5 or later.

The HTTP protocols that underlie web applications were originally developed to support transmission and rendering of simple web pages. New technologies such as JavaScript and cascading style sheets (CSS), and new media types such as Flash videos and graphics-rich images, place heavy demands on front-end performance, that is, on performance at the browser level.

The NetScaler front end optimization (FEO) feature addresses such issues and reduces the load time and render time of web pages by:

- Reducing the number of requests required for rendering each page.
- Reducing the number of bytes in page responses.
- Simplifying and optimizing the content served to the client browser.

You can customize your FEO configuration to provide the best results for your users. NetScaler ADCs support numerous web content optimizations for both desktop and mobile users. The following tables describe the front-end optimizations provided by the FEO feature, and the operations performed on different types of files.

Optimizations Performed by the FEO Feature

Web Optimization	Problem	What NetScaler FEO feature does	Benefits
Inlining	Client browsers often send multiple requests to servers for loading external CSS, images, and JavaScript associated with the web page.	CSS inline JavaScript inline CSS combine	Loading the external CSS, images, and JavaScript inline with the HTML files improves page-rendering time. This optimization is beneficial for content that will be viewed only once, and for mobile devices that have limited cache sizes.
Minification	Data fetched from servers includes inessential characters such as white spaces, comments, and newline characters. The time that browsers spend in processing such data creates website latency.	CSS minification JavaScript minification	Minified files consume less bandwidth and avoid the latency caused by special processing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Web Optimization	Problem	What NetScaler FEO feature does	Benefits
Image optimization	Mobile browsers often have slow connection speeds and limited cache memory. Downloading images on mobile clients consumes more bandwidth, processing time, and cache space, resulting in web site latency.	Removal of HTML comments JPEG optimization CSS image inlining Image shrink-to attributes GIF to PNG conversion HTML image inlining WebP image conversion JPEG, GIF, PNG to JPEG-XR image conversion	Reduces the image to the size indicated in image tag by NetScaler, enabling client browsers to load images faster.
Repositioning	Inefficient processing of external CSS, images, and JavaScript increases page-load time.	Image lazy loading CSS move to Head JavaScript move to end	Repositions HTML elements, to reduce rendering time for web pages and enable client browsers to load the objects faster.
Connection Management	Many browsers set limits on the number of simultaneous connections that can be established to a single domain. This can cause browsers to download webpage resources one at a time, resulting in higher browsers time.	Domain sharding	Overcomes the connection limitation, which improves page-rendering time by enabling client browsers to download more resources in parallel.

Web Optimizations performed on different file types

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following table lists the web optimizations that the NetScaler ADC performs on CSS, Images, JavaScript, and HTML:

Object	Optimization
CSS	• Minify linked CSS files. • Combine multiple linked CSS files that are present within the <head> tag into a single CSS file. • Convert linked CSS files to inline CSS files. • Convert CSS import rule to linked CSS. **Note**: This optimization works if you have not defined the scope and media attributes for the <import> tag and when the <import> tag immediately follows the <style> tag. • Within a linked CSS file, convert linked images to inline images. • Move a CSS present within the <body> tag of an HTML page to the <head> tag. **Note**: The <head> tag must already be present within the HTML script.
Images	• Optimize JPEG images by removing extraneous bytes. • Reduce image size by weakening the image quality to a value specified in FEO parameters. • Image shrink to attributes. • Convert linked images to inline images. **Note**: For animated images in GIF format, only this optimization is supported. • Convert non-animated images in GIF format to PNG format. • Reduce the image size to that specified on the web page, if the size specified on the web page is smaller • Convert images from GIF, PNG, JPEG format to WebP format • Convert images in JPEG format to JPER-XR format • Lazy loading.
JavaScript	• Minify linked JavaScript • Convert linked JavaScript to inline JavaScript.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	• Move JavaScript present in the <body> tag to the end of the <body> tag. **Note**: The size of the <body> tag must be lesser than 64 Kbytes. • Extend cache expiry period.
HTML	• Removal of Comments from HTML

How Front End Optimization Works

After the NetScaler ADC receives the response from the server:

1. Parses the contents of the page, creates an entry in the cache (wherever applicable), and applies the FEO policy. For example, a NetScaler ADC can apply the following optimization rules:
   - Remove white spaces or comments present within a CSS or JavaScript.
   - Combine one or more CSS files to one file.
   - Convert GIF image format to PNG format.
2. Rewrites the embedded objects and saves the optimized content in the cache, with a different signature than the one used for the initial cache entry.
3. For subsequent requests, fetches the optimized objects from the cache, not from the server, and forwards the responses to the client.

*

Remove extraneous information such as white spaces and comments.
*

Remove extraneous information such as white spaces and comments.
**

The period during which the browser can use the cached resource without checking to see if fresh content is available on the server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Front End Optimization

Aug 25, 2016

Optionally, you can change the values of the front end optimization global settings. Otherwise, begin by creating actions that specify the optimization rules to be applied to the embedded objects.

After configuring actions, create policies, each with a rule specifying a type of request for which to optimize the response, and associate the actions with the policies.

Note: The NetScaler ADC evaluates front end optimization policies at request time only, not at response time.
To put the policies into effect, bind them to bind points. You can bind a policy globally, so that it applies to all traffic that flows through the NetScaler ADC, or you can bind the policy to a load balancing or content switching virtual server of type HTTP or SSL. When you bind a policy, assign it a priority. A lower priority number indicates a higher value. The NetScaler ADC applies the policies in the order of their priorities.

Prerequisites for Front End Optimization

Front end optimization requires the NetScaler integrated caching feature to be enabled. Additionally, you must perform the following integrated caching configurations:

- Allocate cache memory.
- Set the maximum response size and memory limit for a default cache content group.

For more information on configuring integrated caching, see Integrated Caching.

Note: The term Integrated Cache can be interchangeably used with AppCache; note that from a functionality point of view, both terms mean the same.

To configure front end optimization by using the command line interface

At the command prompt, do the following:

1. Enable the front end optimization feature.
   enable ns feature FEO

2. Create one or more front end optimization actions.
   add feo action <name> [-imgShrinkToAttrib] [-imgGifToPng] ...

   Example: To add a front end optimization action for converting images in GIF format to PNG format and to extend cache expiry period:
   add feo action allact -imgGifToPng -pageExtendCache
3. [Optional] Specify non-default values for front end optimization global settings.
   set feo parameter [-cacheMaxage <integer>] [-JpegQualityPercent <integer>] [-cssInlineThresSize <integer>] [-inlineJsThresSize <integer>] [-inlineImgThresSize <integer>]

   Example: To specify the cache maximum expiry period:
   set feo parameter -cacheMaxage 10
4. Create one or more front end optimization policy.
   add feo policy <name> <rule> <action>

   Example: To add a front end optimization policy and associate it with the above specified *allact* action:
   >add feo policy pol1 TRUE allact
   >add feo policy pol1 "(HTTP.REQ.URL.CONTAINS(\"testsite\"))" allact1
5. Bind the policy to a load balancing or content switching virtual server, or bind it globally.
   bind lb vserver <name> -policyName <string> -priority <num>

   bind cs vserver <name> -policyName <string> -priority <num>

   bind feo global <policyName> <priority> -type <type> <gotoPriorityExpression>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example: To apply the front end optimization policy to a virtual server named "abc":
> bind lb vserver abc -policyName pol1 -priority 1 -type NONE
Example: To apply the front end optimization policy for all the traffic reaching the ADC:
> bind feo global pol1 100 -type REQ_DEFAULT

6. Save the configuration.
   save ns config

Configuring front end optimization by using the configuration utility

1. Enable the front end optimization feature using the command line interface.
2. Create a front end optimization action.
   Navigate to Optimization > Front End Optimization > Actions, click Add and create a front end optimization action by specifying the relevant details.
3. [Optional] Specify the front end optimization global settings.
   Navigate to Optimization > Front End Optimization, and on the right-pane, under Settings, click Change Front End Optimization settings and specify the front end optimization global settings.
4. Create a front end optimization policy.
   Navigate to Optimization > Front End Optimization > Policies, click Add and create a front end optimization policy by specifying the relevant details.
5. Bind the policy to a load balancing or content switching virtual server.
   1. Navigate to Optimization > Front End Optimization > Policies.
   2. Select a front end optimization policy and click Policy Manager.
   3. Under Front End Optimzation Policy Manager, bind the front end optimization policy to a load balancing or content switching virtual server.

Verifying Front End Optimization Configuration

Updated: 2015-01-12

The dashboard utility displays summary and detailed statistics in tabular and graphic formats. You can view the FEO statistics to evaluate your FEO configuration.

Optionally, you can also display statistics for an FEO policy, including the number of hits that the policy counter increments during policy based FEO.

Note: For more information about statistics and charts, see the Dashboard help on the Citrix NetScaler appliance.

To View FEO statistics by using the command line interface

At the command prompt, type the following commands to display a summary of FEO statistics, FEO policy hits and details, and detailed FEO statistics, respectively:

- stat feo
  Note: The **stat feo policy** command displays statistics only for advanced FEO policies.
- show feo policy <name>
- stat feo -detail

To view FEO statistics on the Dashboard

© 1999-2017 Citrix Systems, Inc. All rights reserved.

In the Dashboard utility, you can:

- Select **Front End Optimization** to display a summary of FEO statistics.
- Click the **Graphical View** tab to display the rate of requests processed by the FEO feature.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Sample Optimization

Apr 22, 2014

The following table lists some examples of content optimization actions that are applied on HTML content and the embedded objects within the HTML content.

Optimization rule	Sample
Collapse white spaces within an HTML page	Before: <title>Hello, world! </title>
	After: <title>Hello, world!</title>
Combine CSS	Before: <link rel="stylesheet" type="text/css" href="**sheet/abc.css**"> <link rel="stylesheet" type="text/css" href="**sheet/xyz.css**">
	After: <link rel="stylesheet" type="text/css" href="**sheet/abc.css+xyz.css**">
Inline CSS	Before <html> <head> <link rel="sheet" href="abc.css"/> </head> <body> Hi! </body> </html> Note: abc.css contains .Alice {location: Australia;} .Tom {location: Asia;}
	After <html> <head> <style> .Alice {location: Australia;} .Tom {location: Asia;} </style> </head> <body> Hi!

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Optimization rule	Sample
	</body> **Sample** </html>
Move CSS to head	Before: <html> <head> </head> <body> <script src="abc.js" type="text/javascript"></script> <div class="monday tuesday> Hi! <style type="text/css"> .foo { day: wednesday; } </style> <link rel="stylesheet" type="text/css" href="styles/all_styles.css"> </body> </html>
	After: <html> <head> <style type="text/css"> .foo { day: wednesday; } </style> </head> <body> <script src="abc.js" type="text/javascript"></script> <div class="monday tuesday> Hi! <link rel="stylesheet" type="text/css" href="styles/all_styles.css"> </body> </html>

Minify JavaScript	Before:
	/* Remove this comment */
	document.write("abc " + state);
	state += 1; // Update this.

	After: document.write("abc "+state);state+=1;
Convert linked JavaScript to inline JavaScript	Before <html> <head> <script type="text/javascript" src="abc.js"></script> </head> <body>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Optimization rule	Sample
	 Hi! </body> </html> Note: abc.js contains /* contents of abc JavaScript file */
	After <html> <head> <script type="text/javascript"> /* contents of abc JavaScript file */ </script> </head> <body> Hi! </body> </html>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Content Accelerator

Nov 27, 2015

> **Important**
>
> The content accelerator feature is no longer supported on the NetScaler appliance.

Content accelerator is a NetScaler feature that you can use in a Citrix ByteMobile T1100 deployment, to store data on a Citrix ByteMobile T2100 appliance. For more information about Citrix ByteMobile, see How ByteMobile Works.
Storing data on a T2100 appliance saves bandwidth and provides faster response times, because the NetScaler does not have to connect to the server for repeated requests of the same data.

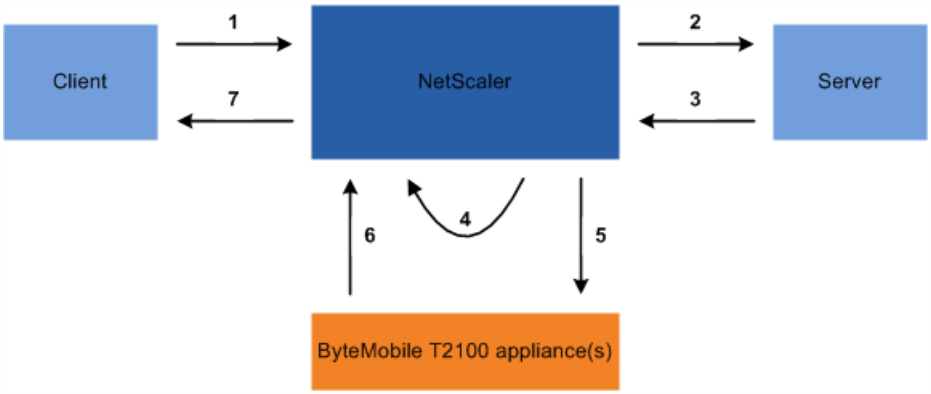
Note: Content accelerator works with a Citrix ByteMobile platinum license. Contact customer support for more information and for obtaining the license.

How Content Accelerator Works

Updated: 2015-05-20

When a load balancing or content switching virtual server receives a client request, the NetScaler appliance evaluates a content accelerator policy that you have bound to the virtual server. The policy filters the requests to identify the ones to which to apply the content accelerator feature.

Note: For HTTP requests, the content accelerator feature can serve partial content in response to single byte-range requests.
The following figure illustrates the operations that the appliance performs when a client request arrives at a virtual server configured to use the content accelerator feature:



The process flow is as follows:

1. Client sends request.
2. NetScaler forwards the request to the server.
3. Server responds with the predefined size of the response (specified by the accumResSize parameter of the add ca action command).
4. NetScaler computes a hash of the response sent by the server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. NetScaler looks up the hash on the T2100 appliance.

6. A successful lookup indicates that the data is available and the T2100 appliance sends the data to the NetScaler.

   Note:
   - When a lookup does not succeed, the NetScaler fetches all of the requested data from the server, and simultaneously serves the data to the client and updates the data on the T2100 appliance.
   - The T2100 appliance can be configured to specify the number of requests after which to cache the data.

7. NetScaler sends the response to the client.

Configuring Content Accelerator

Updated: 2013-12-16

Before configuring the content accelerator feature, you must enable it on the NetScaler appliance.

You can configure the content accelerator feature to use one or multiple T2100 appliances. You must add each T2100 appliance as a service and bind these services to a load balancing virtual server that is dedicated to distributing the load between the configured T2100 appliances.

You must also configure a content accelerator action to lookup the data on the T2100 appliance. The action must also specify the T2100 load balancing virtual server and the size of data (in KB) to be fetched from the server to calculate the hash.

The action must be bound to a content accelerator policy that defines the traffic on which to perform content acceleration. The content accelerator policy must be bound to a content switching or load balancing virtual server that receives client traffic. Alternatively, you can bind the policy globally to be applicable to all virtual servers.

Configuring content accelerator by using the command line interface

At the command prompt, do the following:

1. Enable the content accelerator feature.
   enable ns feature ca

2. Identify the T2100 appliances and add each as a service on the NetScaler appliance.
   add service <name> <IPAddress> <serviceType> <port>

   Example:
   > add service T2100-A 10.102.29.61 HTTP 30
   > add service T2100-B 10.102.29.62 HTTP 40
   > add service T2100-C 10.102.29.63 HTTP 50
   Note: The services must be of type HTTP only.

3. Create a load balancing virtual server for the T2100 appliances. Specify the token load balancing method and the rule shown in the following syntax.
   add lb vserver <name> <serviceType> <IPAddress> <port> -lbMethod TOKEN -rule "http.req.url.after_str(\"/lookup/\") alt http.req.url.path.SKIP(1).PREFIX(64)"

   Example:
   > add lb vserver T2100-lbvserver HTTP 10.102.29.64 99 -lbMethod TOKEN -rule "http.req.url.after_str(\"/lookup/\") alt

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

http.req.url.path.SKIP(1).PREFIX(64)"

4. Bind the T2100 services to the load balancing virtual server that you created for them.
   bind lb vserver <name> <serviceName>

   Example:
   > bind lb vserver T2100-lbvserver T2100-A
   > bind lb vserver T2100-lbvserver T2100-B
   > bind lb vserver T2100-lbvserver T2100-C
5. Define a content accelerator action.
   add ca action <name> -accumResSize <KBytes> -lbvserver <string> -type lookup

   Example:
   > add ca action ca_action1 -type lookup -lbvserver T2100-lbvserver -accumResSize 60
6. Define a content accelerator policy.
   add ca policy <name> -rule <expression> -action <name>

   Example: To create a content accelerator policy that caches all video formats.
   > add ca policy ca_mp4_pol -rule ns_video -action ca_action1
   where ns_video is a built-in expression.

7. Bind the content accelerator policy to either a virtual server that receives traffic or globally to the NetScaler system.
   bind lb vserver <name> -policyName <string>

   bind cs vserver <name> -policyName <string>

   bind ca global -policyName <string> -priority <num> -type <type>

   Example: To apply the content accelerator policy to a virtual server named "traf_rec"
   > bind lb vserver traf_rec -policyName ca_mp4_pol
   Example: To apply the content accelerator policy for all traffic reaching the NetScaler.
   > bind ca global -policyName ca_mp4_pol -priority 100 -type RES_DEFAULT
8. Save the configuration.
   save ns config

Configuring content accelerator by using the configuration utility

1. Navigate to System > Settings > Configure Advanced Features and select Content Accelerator.
2. Create a service for each of the T2100 appliances.
   1. Navigate to Traffic Management > Load Balancing > Services.
   2. Click Add and specify the relevant details. In the Server field, make sure you specify the IP address of the T2100 appliance. In the Protocol field select HTTP.
3. Create a virtual server and bind the T2100 services to it.
   1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
   2. Click Add and specify the relevant details.
   3. In the Method and Persistence tab, specify the Method as Token.
   4. In the Policies tab, specify the rule as http.req.url.after_str(\"/lookup/\") alt http.req.url.path.SKIP(1).PREFIX(64).
   5. In the Services tab, select the T2100 services that you want to bind to the virtual server.
4. Create a content accelerator action.
   1. Navigate to Optimization > Content Accelerator > Actions.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

   2. Specify the relevant details.
5. Create a content accelerator policy.
   1. Navigate to Optimization > Content Accelerator > Policies.
   2. Click Add, specify the policy rule, and associate the content accelerator action.
6. Bind the content accelerator policy globally or to a virtual server.
   1. Navigate to Optimization > Content Accelerator.
   2. Under the Content Accelerator Policy Manager [REQUEST] or Content Accelerator Policy Manager [RESPONSE] sections, bind the content accelerator policy globally or to a virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SPDY (Speedy)

May 20, 2015

Note: Supported from NetScaler 10.1 onwards.

SPDY is an open networking experimental protocol developed by Google to reduce the time taken by a client to load a web page in a browser. An application layer protocol, SPDY changes the way in which HTTP requests and responses are handled. SPDY offers the following advantages compared to a regular HTTP transaction:

- Multiplexed requests and responses—In a single SPDY session, multiple requests from the client can be sent over a single TCP connection to the server. This reduces the number of TCP connections and also optimizes usage of each TCP connection.
- Request prioritization—When requesting services from the server, a client can assign a priority to each request.
- Header Compression—SPDY compresses the HTTP request and response headers, saving bandwidth and reducing latency.
- Server push—The server can send data to the client before the client requests it.
- Security—SPDY is secure by design, because SSL is required for SPDY connections.

NetScaler supports the SPDY/2 and SPDY/3 (from NetScaler 10.5 onwards) versions.

Note: SPDY support depends on the browser version being used.

If you use a NetScaler appliance as a SPDY gateway for your servers, the servers do not have to support SPDY. The NetScaler appliance accepts the incoming SPDY requests, converts them, and sends them to the servers as HTTP requests. It also converts the HTTP responses and sends them to the clients as SPDY responses. While the key value of SPDY is reduced bandwidth consumption and faster communication with clients, an additional benefit of the NetScaler solution is that you avoid the time consuming task of upgrading your web servers and applications to support SPDY.

To use a NetScaler appliance as a SPDY gateway, you must enable SPDY on the appliance.

This document includes the following details:

- How SPDY Works over SSL
- Configuring SPDY on the NetScaler Appliance
- Troubleshooting for SPDY

SPDY Requirements

Both ends of a SPDY connection must support the same version of SPDY. In addition, the clients must meet the following requirements:

- Support ZLIB compression and accept compressed data.
- Support the Next Protocol Negotiation (NPN) TLS extension, because NPN is used in the TLS handshake.

How SPDY Works over SSL

Updated: 2014-03-13

If SPDY is enabled, when the NetScaler appliance sees TLS ALPN extension with list of supported protocols in the Client Hello message, it responds with either SPDY/3 or SPDY/2 in the ALPN extension in its Server Hello.

NetScaler can also negotiate SPDY over NPN. When NetScaler sees an empty NPN extension in the Client Hello message, it responds with a list of the protocols that it supports. If SPDY is enabled on the NetScaler appliance, the appliance advertises HTTP/1.1 and SPDY/2 protocols. The client selects one protocol from this list and negotiates the protocol with

© 1999-2017 Citrix Systems, Inc. All rights reserved.

the server. Because sending the negotiated protocol in plain text would raise security issues, the client sends the Change Cipher Spec notification which defines the details of the encryption for the session, followed by the Next Protocol message, which contains the encrypted protocol that the client has chosen. The client then sends the Finished message. The NetScaler appliance decrypts the Next Protocol message, and then sends a Finished message.

A session is then established, and application data can be exchanged.

Note: The NPN extension is not supported on a NetScaler FIPS appliance, and with TLS protocol versions 1.1 and 1.2.

Configuring SPDY on the NetScaler Appliance

Updated: 2014-09-15

By default, SPDY is disabled on the NetScaler appliance. After you enable SPDY, the appliance advertises SPDY/2 and/or SPDY/3 along with HTTP/1.1 during an SSL handshake. To enable SPDY on the NetScaler appliance, you must enable SPDY in the HTTP profile bound to the SSL virtual server.

To configure SPDY by using the command line interface

At the command prompt, do the following:

1. Enable SPDY on a HTTP profile.
   set ns httpProfile <profileName> -SPDY <options>

   Example
   > set ns httpProfile profile1 -SPDY ENABLED
2. Bind the HTTP profile to a SSL virtual server.
   set lb vserver <ssl-vserver-name> -httpProfileName <httpProfile-with-spdy>

   Example
   > set lb vserver SPDY_LB -httpProfileName profile1
   Note: To apply SPDY globally, enable SPDY on the global HTTP profile (nshttp_default_profile).

You can view the statistics by using the following command:

stat protocol http -detail

To configure SPDY by using the configuration utility

1. Navigate to System > Profiles, and in the HTTP Profiles tab, update the profile on which you want to enable SPDY.
2. Navigate to Traffic Management > Load Balancing > Virtual Servers, and associate the HTTP profile to the appropriate SSL virtual server.

Troubleshooting for SPDY

If SPDY sessions are not enabled even after performing the required steps, check the following conditions.

- If the client is using a Chrome browser, SPDY might not work in some scenarios because Chrome sometimes does not initiate TLS handshake.
- If there is a forward-proxy between the client and the NetScaler appliance, and the forward-proxy doesn't support SPDY, SPDY sessions might not be enabled.
- NetScaler does not support NPN over TLS 1.1/1.2. To use SPDY, the client should disable TLS1.1/1.2 in the browser.
- Similarly, if the client wants to use SPDY, SSL2/3 must be disabled on the browser.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Media Classification

Jun 22, 2015

Understanding the type of traffic in the network helps network administrators to manage bandwidth consumption for optimal network performance. The media classification mode monitors and displays the statistics of media traffic going through the NetScaler appliance.

With this mode enabled, a network administrators can collect stats showing the amount of data accessed, and the types of devices from which the media files have been accessed. The NetScaler appliance also supports byte-range requests in this mode.

Currently the NetScaler appliance can monitor and display statistics for the following media file types:

Media File Name	Media File Type
Microsoft Smooth Streaming	Video
Apple Live Streaming	Video
Audio Data Transport Stream (ADTS)	Audio
Advanced Audio Coding (AAC)	Audio
Flash Video (FLV)	Audio and Video
3GP	Audio and Video

The appliance can display stats for the following devices:

Device Platform	Device Type
iOS	iPad and iPod
Android	Mobiles and tablets
Laptop or Desktop	Windows laptop and desktop computers
Others	Other mobile devices (mobiles and tablets)

The network administrators can check the following stats counters to know the amount of data accessed through the NetScaler appliance for various media traffic types.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Media File Name	Stats Counter
Microsoft Smooth Streaming	• **mcmssmthstrmvid**—This counter records the total number of Microsoft Smooth Streaming videos served by the NetScaler appliance. • **Mcmssmthstrvidpl**—This counter records the total number of Microsoft Smooth Streaming video playlists served by the NetScaler appliance. • **Mcmssmthstrmvidbytes**—This counter records the total number of data bytes served for Microsoft Smooth Streaming media traffic on the NetScaler appliance. • **Mcmssmthstrmplvidbytespl**—This counter records the total number of Microsoft Smooth Streaming playlist bytes served by the NetScaler appliance.
Apple Live Streaming	• **mccaplelivestrmngvid**—This counter records the total number of Apple Live Streaming videos served by the NetScaler appliance. • **Mccapplelivestrmngvidpl**—This counter records the total number of Apple Live Streaming video playlists served by the NetScaler appliance. • **Mcapplelivestreamingvidbytes**—This counter records the total number of data bytes served for Apple Live Streaming media traffic on the NetScaler appliance. • **Mcapplelivestreamingplaylistvidbytespl**—This counter records the total number of Apple Live Playlist bytes served by the NetScaler appliance.
Audio Data Transport Stream (ADTS)	• **mcadtsaudio**—This counter records the total number of ADTS audio clips served by the NetScaler appliance. • **Mcadtsaudiobytes**—This counter records the total number of data bytes served for ADTS media traffic on the NetScaler appliance.
Advanced Audio Coding (AAC)	• **Mcaacaudio**—This counter records the total number of AAC audio clips served by the NetScaler appliance. • **Mcaacaudiobytes**—This counter records the total number of data bytes served for AAC media traffic on the NetScaler appliance.
Flash Video (FLV)	• **Mcflvvid**—This counter records the total number of flash videos served by the NetScaler appliance. • **Mcflvvidbytes**—This counter records the total number of data bytes served for flash videos on the NetScaler appliance.
3GP	• **mc3gpvidbytes**—This counter records the total number of data bytes served for 3GP media traffic on the NetScaler appliance.

The NetScaler appliance detects media file types by their signatures in the *initial body bytes* of the responses. For example, the initial body bytes for an mp4 file has the following signature in the response:

....ftypmp42 ....isommp42....moov...lmvhd.....c.!.c.!..

The NetScaler appliance detects the client device type by the *user agent string* that the client device includes in the HTTP

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

GET request. For example, a windows phone using a UC browser will have the following user agent string in the HTTP GET request:

User-Agent: **UCWEB**/2.0 (**Windows**; U; wds 8.10; en-US; HTC; 8X by HTC) U2/1.0.0

Enabling Media Classification

By default, media classification is disabled on the NetScaler appliance. You have to enable the mode before using it.

To enable media classification by using the command line interface

At the command prompt, type:

enable ns mode Mediaclassification

To enable media classification by using the configuration utility

- Enable media classification on NetScaler appliance
  Navigate to **System** > **Settings** > **Configure Modes** and select **Media Classification**.
- View media traffic statistics on the NetScaler appliance
  Navigate to **Optimization** and click **Media Classification** to view the media traffic statistics.

Verifying Media Classification Statistics

You can view the media traffic statistics in the dashboard utility or using the command line interface. The dashboard utility displays summary and detailed statistics in a tabular and graphic format.

Note: For more information about statistics and charts, see the Dashboard help on your Citrix NetScaler appliance.

To View media classification statistics by using the command line interface

At the command prompt, type one of the following commands to display a summary of media classification statistics, display detailed statistics, or clear the display:

- stat Mediaclassification
- stat Mediaclassification -detail
- stat Mediaclassification -clearstats

To view Media Classification statistics on the Dashboard

In the Dashboard utility, you can display the following types of media classification statistics:

- Select **Media Classification** to display a summary of the media traffic statistics.
- To display detailed media traffic statistics, click the **Details**.
- To clear the media traffic statistics, click **Clear**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Security

May 03, 2013

The following topics cover configuration and installation information for NetScaler security features. Most of these features are policy based.

Authentication, Authorization, Auditing (AAA)	Keeps unauthorized users out of the network, denies users access to tasks for which they are not authorized, and tracks the resources used during user sessions.
Application Firewall	Prevents security breaches, data loss, and possible unauthorized modifications to web sites that access sensitive business or customer information.
Content Filtering	Blocks inappropriate HTML requests, preventing the requests from reaching the Web servers.
HTTP Denial-of-Service Protection	Prevents hackers from attacking your Web site with large numbers of HTTP requests.
Priority Queuing	Detects high-priority connections and allows those connections to proceed ahead of other connections, guaranteeing unimpeded access to those users.
SureConnect	Services all incoming connections with either the requested content or a custom Web page that displays information about a delay in the request being serviced.
Surge Protection	Detects any rapid rise in connection attempts and adjusts the rate at which connections are allowed to proceed to the server, preventing server overload.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AAA Application Traffic

Sep 18, 2013

Many companies restrict web site access to valid users only, and control the level of access permitted to each user. The AAA feature allows a site administrator to manage access controls with the NetScaler appliance instead of managing these controls separately for each application. Doing authentication on the appliance also permits sharing this information across all web sites within the same domain that are protected by the appliance.

The AAA feature supports authentication, authorization, and auditing for all application traffic. To use AAA, you must configure authentication virtual servers to handle the authentication process and traffic management virtual servers to handle the traffic to web applications that require authentication. You also configure your DNS to assign FQDNs to each virtual server. After configuring the virtual servers, you configure a user account for each user that will authenticate via the NetScaler appliance, and optionally you create groups and assign user accounts to groups. After creating user accounts and groups, you configure policies that tell the appliance how to authenticate users, which resources to allow users to access, and how to log user sessions. To put the policies into effect, you bind each policy globally, to a specific virtual server, or to the appropriate user accounts or groups. After configuring your policies, you customize user sessions by configuring session settings and binding your session policies to the traffic management virtual server. Finally, if your intranet uses client certs, you set up the client certificate configuration.

Before configuring AAA, you should be familiar with and understand how to configure load balancing, content switching, and SSL on the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How AAA Works

Oct 21, 2015

AAA provides security for a distributed internet environment by allowing any client with the proper credentials to connect securely to protected application servers from anywhere on the Internet. This feature incorporates the three security features of authentication, authorization, and auditing. Authentication enables the NetScaler ADC to verify the client's credentials, either locally or with a third-party authentication server, and allow only approved users to access protected servers. Authorization enables the ADC to verify which content on a protected server it should allow each user to access. Auditing enables the ADC to keep a record of each user's activity on a protected server.

To understand how AAA works in a distributed environment, consider an organization with an intranet that its employees access in the office, at home, and when traveling. The content on the intranet is confidential and requires secure access. Any user who wants to access the intranet must have a valid user name and password. To meet these requirements, the ADC does the following:

- Redirects the user to the login page if the user accesses the intranet without having logged in.
- Collects the user's credentials, delivers them to the authentication server, and caches them in a directory that is accessible through LDAP.
- Verifies that the user is authorized to access specific intranet content before delivering the user's request to the application server.
- Maintains a session timeout after which users must authenticate again to regain access to the intranet. (You can configure the timeout.)
- Logs the user accesses, including invalid login attempts, in an audit log.

Authentication requires that several entities: the client, the NetScaler appliance, the external authentication server if one is used, and the application server, respond to each other when prompted by performing a complex series of tasks in the correct order. If you are using an external authentication server, this process can be broken down into the following fifteen steps.

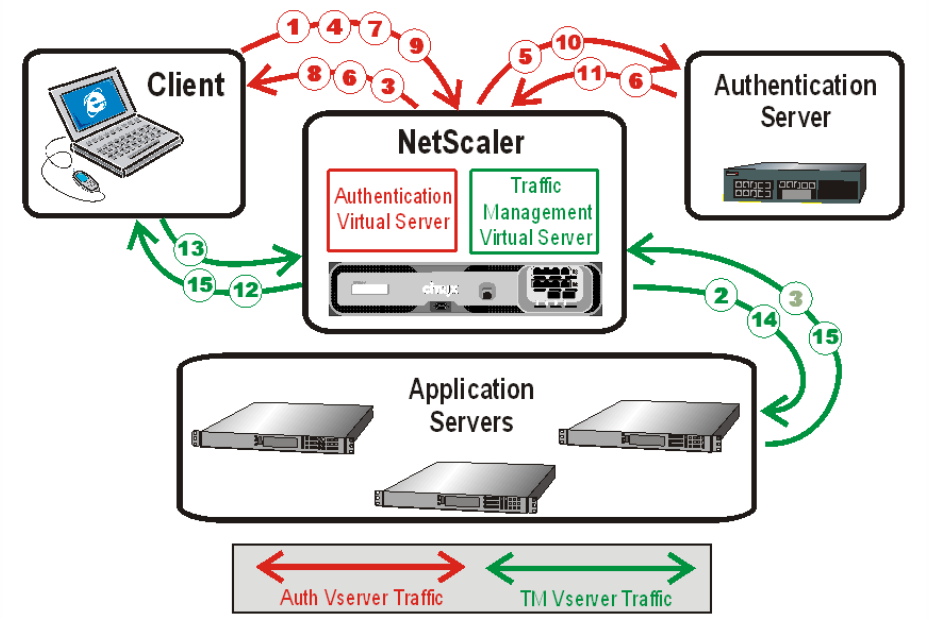
1. The client sends a GET request for a URL on the application server.
2. The NetScaler appliance's traffic management virtual server redirects the request to the application server.
3. The application server determines that the client has not been authenticated, and therefore sends an HTTP 200 OK response via the TM vserver to the client. The response contains a hidden script that causes the client to issue a POST request for /cgi/tm.
4. The client sends a POST request for /cgi/tm.
5. The NetScaler appliance's authentication virtual server redirects the request to the authentication server.
6. The authentication server creates an authentication session, sets and caches a cookie that consists of the initial URL and the domain of the traffic management virtual server, and then sends an HTTP 302 response via the authentication virtual server, redirecting the client to /vpn/index.html.
7. The client sends a GET request for /vpn/index.html.
8. The authentication virtual server redirects the client to the authentication server login page.
9. The client sends a GET request for the login page, enters credentials, and then sends a POST request with the credentials back to the login page.
10. The authentication virtual server redirects the POST request to the authentication server.
11. If the credentials are correct, the authentication server tells the authentication virtual server to log the client in and redirect the client to the URL that was in the initial GET request.
12. The authentication virtual server logs the client in and sends an HTTP 302 response that redirects the client to the initially requested URL.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

13. The client sends a GET request for their initial URL.
14. The traffic management virtual server redirects the GET request to the application server.
15. The application server responds via the traffic management virtual server with the initial URL.

If you use local authentication, the process is similar, but the authentication virtual server handles all authentication tasks instead of forwarding connections to an external authentication server. The following figure illustrates the authentication process.

Figure 1. Authentication Process Traffic Flow



When an authenticated client requests a resource, the ADC, before sending the request to the application server, checks the user and group policies associated with the client account, to verify that the client is authorized to access that resource. The ADC handles all authorization on protected application servers. You do not need to do any special configuration of your protected application servers.

AAA-TM handles password changes for users by using the protocol-specific method for the authentication server. For most protocols, neither the user nor the administrator needs to do anything different than they would without AAA-TM. Even when an LDAP authentication server is in use, and that server is part of a distributed network of LDAP servers with a single designated domain administration server, password changes are usually handled seamlessly. When an authenticated client of an LDAP server changes his or her password, the client sends a credential modify request to AAA-TM, which forwards it to the LDAP server. If the user's LDAP server is also the domain administration server, that server responds appropriately and AAA-TM then performs the requested password change. Otherwise, the LDAP server sends AAA-TM an LDAP_REFERRAL response to the domain administration server. AAA-TM follows the referral to the indicated domain administration server, authenticates to that server, and performs the password change on that server.

When configuring AAA-TM with an LDAP authentication server, the system administrator must keep the following conditions and limitations in mind:

- AAA-TM assumes that the domain administration server in the referral accepts the same bind credentials as the original server.
- AAA-TM only follows LDAP referrals for password change operations. In other cases AAA-TM refuses to follow the referral.
- AAA-TM only follows one level of LDAP referrals. If the second LDAP server also returns a referral, AAA-TM refuses to

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

follow the second referral.

The ADC supports auditing of all states and status information, so you can see the details of what each user did while logged on, in chronological order. To provide this information, the appliance logs each event, as it occurs, either to a designated audit log file on the appliance or to a syslog server. Auditing requires configuring the appliance and any syslog server that you use.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling AAA

Oct 30, 2013

To use the AAA - Application Traffic feature, you must enable it. You can configure AAA entities—such as the authentication and traffic management virtual servers—before you enable the AAA feature, but the entities will not function until the feature is enabled.

To enable AAA by using the command line interface

At the command prompt, type the following commands to enable AAA and verify the configuration:

- enable ns feature AAA
- show ns feature

Example

```
> enable feature AAA
 Done

> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging            WL               OFF
 2)   Surge Protection       SP               ON
 .
 .
 .
 15)  AAA                    AAA              ON
 .
 .
 .
 23)  HTML Injection         HTMLInjection    ON
 24)  NetScaler Push         push             OFF
 Done
```

To enable AAA by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Authentication, Authorization and Auditing check box.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting Up an Authentication Virtual Server

Oct 30, 2015

All authentication requests are redirected by the traffic management virtual server (load balancing or content switching) to the authentication virtual sever. This virtual server processes the associated authentication policies and accordingly provides access to the application.

To set up an authentication virtual server by using the NetScaler CLI

1. Enable the AAA feature.

   ns-cli-prompt> **enable ns feature** AAA

2. Configure an authentication virtual server. It must be of type SSL and make sure to bind SSL certificate-key pair to the virtual server.

   ns-cli-prompt> **add authentication vserver <name> SSL <ipaddress> <port>**

   ns-cli-prompt> **bind ssl certkey <auth-vserver-name> <certkey>**

3. Specify the FQDN of the domain for the authentication virtual server.

   ns-cli-prompt> **set authentication vserver <name> -authenticationDomain** <FQDN>

4. Associate the authentication virtual server to the relevant traffic management virtual server.

   Note: The FQDN of the traffic management virtual server must be in the same domain as the FQDN of the authentication virtual server for the domain session cookie to function correctly.

   On the traffic management virtual server:

   - Enable authentication.
   - Specify the FQDN of the authentication virtual server as the authentication host of the traffic management virtual server.
   - [Optional] Specify the authentication domain on the traffic management virtual server.

   Note: If you do not configure the authentication domain, the appliance assigns an FQDN that consists of the FQDN of the authentication virtual server without the hostname portion. For example, if domain name of the authentication virtual server is **tm.xyz.bar.com**, the appliance assigns **xyz.bar.com** as the authentication domain.

   For load balancing:
   ns-cli-prompt> **set lb vserver <name> -authentication** ON **-authenticationhost** <FQDN>
   [**-authenticationdomain** <authdomain>]

   For content switching:
   ns-cli-prompt> **set lb vserver <name> -authentication** ON **-authenticationhost** <FQDN>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

[-**authenticationdomain** <authdomain>]

5. Verify that both the virtual servers are UP and configure correctly.

   ns-cli-prompt> **show authentication vserver** <name>

To set up an authentication virtual server by using the NetScaler GUI

1. Enable the AAA feature.

   Navigate to **System > Settings**, click **Configure Basic features**, and enable **Authentication, Authorization and Auditing**.

2. Configure the authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and configure as required (refer to the configurations provided in the CLI procedure provided above).

3. Configure the traffic management virtual server for authentication.

   For load balancing:
   Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and configure the virtual server as required (refer to the configurations provided in the CLI procedure provided above).

   For content switching:
   Navigate to **Traffic Management > Content Switching > Virtual Servers**, and configure the virtual server as required (refer to the configurations provided in the CLI procedure provided above).

4. Verify the authentication setup.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and check the details of the relevant authentication virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Authentication Virtual Server

Aug 13, 2014

To configure AAA, first configure an authentication virtual server to handle authentication traffic. Next, bind an SSL certificate-key pair to the virtual server to enable it to handle SSL connections. For additional information about configuring SSL and creating a certificate-key pair, see the *Citrix NetScaler Traffic Management Guide* at "Traffic Management."

To configure an authentication virtual server by using the command line interface

To configure an authentication virtual server and verify the configuration, at the command prompt type the following commands in the order shown:

- add authentication vserver <name> ssl <ipaddress>
- show authentication vserver <name>
- bind ssl certkey <certkeyName>
- show authentication vserver <name>
- set authentication vserver <name> -authenticationDomain <FQDN>
- show authentication vserver <name>

Example

```
> add authentication vserver Auth-Vserver-2 SSL 10.102.29.77 443
 Done
> show authentication vserver Auth-Vserver-2
      Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
      State: DOWN[Certkey not bound]
      Client Idle Timeout: 180 sec
      Down state flush: DISABLED
      Disable Primary Vserver On Down : DISABLED
      Authentication : ON
      Current AAA Users: 0
 Done
> bind ssl certkey Auth-Vserver-2 Auth-Cert-1
 Done
> show authentication vserver Auth-Vserver-2
      Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
      State: UP
      Client Idle Timeout: 180 sec
      Down state flush: DISABLED
      Disable Primary Vserver On Down : DISABLED
      Authentication : ON
      Current AAA Users: 0
      Authentication Domain: myCompany.employee.com
 Done
> set authentication vserver Auth-Vserver-2 -AuthenticationDomain myCompany.employee.com
 Done
> show authentication vserver Auth-Vserver-2
      Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
      State: DOWN[Certkey not bound]
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Client Idle Timeout: 180 sec
    Down state flush: DISABLED
    Disable Primary Vserver On Down : DISABLED
    Authentication : ON
    Current AAA Users: 0
    Authentication Domain: myCompany.employee.com
Done
```

To configure an authentication virtual server by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Virtual Servers.
2. In the details pane, do one of the following:
   - To create a new authentication virtual server, click Add.
   - To modify an existing authentication virtual server, select the virtual server, and then click Edit.

   The Configuration dialog opens with the Basic Settings area expanded.
3. Specify values for the parameters as follows (asterisk indicates a required parameter):
   - Name*—name (Cannot be changed for a previously created virtual server)
   - IP Address*—ipaddress
   - Domain*—authenticationDomain
   - Failed login timeout—failedLoginTimeout (Seconds allowed before login fails and user must start login process again.)
   - Max login attempts—maxLoginAttempts (Number of login attempts allowed before user is locked out)

   Note: The authentication virtual server uses only the SSL protocol and port 443, so those options are greyed out. Any options that are not mentioned are not relevant and should be ignored.
4. Click Continue to display the Certificates area.
5. In the Certificates area, configure any SSL certificates you want to use with this virtual server.
   - To configure a CA certificate, click the arrow on the right of CA Certificate to display the CA Cert Key dialog box, select the certificate you want to bind to this virtual server, and click Save.
   - To configure a server certificate, click the arrow on the right of Server Certificate, and follow the same process as for CA certificate.
6. Click Continue to display the Advanced Authentication Policies area.
7. If you want to bind an advanced authentication policy to the virtual server, click the arrow on the right side of the line to display the Authentication Policy dialog box, choose the policy that you want to bind to the server, set the priority, and then click OK.
8. Click Continue to display the Basic Authentication Policies area.
9. If you want to create a basic authentication policy and bind it to the virtual server, click the plus sign to display the Policies dialog box, and follow the prompts to configure the policy and bind it to this virtual server.
10. Click Continue to display the 401-Based Virtual Servers area.
11. In the 401-Based Virtual Servers area, configure any load balancing or content switching virtual servers that you want to bind to this virtual server.
    - To bind a load balancing virtual server, click the arrow to the right of LB virtual server to display the LB Virtual Servers dialog box, and follow the prompts.
    - To bind a content switching virtual server, click the arrow to the right of CS virtual server to display the CS Virtual Servers dialog box, and follow the same process as to bind an LB virtual server.
12. If you want to create or configure a group, in the Groups area click the arrow to display the Groups dialog box, and follow the prompts.
13. Review your settings, and when you are finished, click Done. The dialog box closes. If you created a new authentication virtual server, it now appears in the Configuration window list.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Traffic Management Virtual Server

Aug 19, 2014

After you have created and configured your authentication virtual server, you next create or configure a traffic management virtual server and associate your authentication virtual sever with it. You can use either a load balancing or content switching virtual server for a traffic management virtual server. For more information about creating and configuring either type of virtual server, see the *Citrix NetScaler Traffic Management Guide* at Traffic Management.

Note: The FQDN of the traffic management virtual server must be in the same domain as the FQDN of the authentication virtual server for the domain session cookie to function correctly.

You configure a traffic management virtual server for AAA by enabling authentication and then assigning the FQDN of the authentication server to the traffic management virtual server. You can also configure the authentication domain on the traffic management virtual server at this time. If you do not configure this option, the NetScaler appliance assigns the traffic management virtual server an FQDN that consists of the FQDN of the authentication virtual server without the hostname portion. For example, if domain name of the authentication vserver is tm.xyz.bar.com, the appliance assigns xyz.bar.com. as the authentication domain.

To configure a TM virtual server for AAA by using the command line interface

At the command prompt, type one of the following sets of commands to configure a TM virtual server and verify the configuration:

- set lb vserver <name> –authentication ON -authenticationhost <FQDN> [-authenticationdomain <authdomain>]
- show lb vserver <name>
- set cs vserver <name> –authentication ON -authenticationhost <FQDN> [-authenticationdomain <authdomain>]
- show cs vserver <name>

Example

```
> set lb vserver vs-cont-sw -Authentication ON -AuthenticationHost mywiki.index.com
 Done
> show lb vserver vs-cont-sw
      vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
      State: DOWN
      Last state change was at Wed Aug 19 10:03:15 2009 (+410 ms)
      Time since last state change: 5 days, 20:00:40.290
      Effective State: DOWN
      Client Idle Timeout: 9000 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      No. of Bound Services :  0 (Total)      0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Connection Failover: DISABLED
      Authentication: ON      Host: mywiki.index.com
 Done
```

To configure a TM virtual server for AAA by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. In the navigation pane, do one of the following.
   - Navigate to Traffic Management > Load Balancing > Virtual Servers.
   - Navigate to Traffic Management > Content Switching > Virtual Servers
     The AAA configuration process for either type of virtual server is identical.
   - In the details pane, select the virtual server on which you want to enable authentication, and then click Edit.
   - In the Domain text box, type the authentication domain.
   - In the Advanced menu on the right, select Authentication.
   - Choose either Form Based Authentication or 401 Based Authentication., and fill in the Authentication information.
     - For Form Based Authentication, enter the Authentication FQDN (the fully-qualified domain name of the authentication server), the Authentication VServer (the IP address of the authentication virtual server), and the Authentication Profile (the profile to use for authentication).
     - For 401 Based Authentication, enter the Authentication VServer and the Authentication Profile only.
   - Click OK. A message appears in the status bar, stating that the vserver has been configured successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring DNS

Sep 24, 2013

For the domain session cookie used in the authentication process to function correctly, you must configure DNS to assign both the authentication and the traffic management virtual servers to FQDNs in the same domain. For information about how to the configure DNS address records, see the *Citrix NetScalerTraffic Management Guide* at ".Traffic Management"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Verifying Your Setup for AAA

Sep 18, 2013

After you configure authentication and traffic management virtual servers and before you create user accounts, you should verify that both virtual servers are configured correctly and are in the UP state.

To verify authentication virtual server setup by using the command line interface

At the command prompt, type the following command:

show authentication vserver <name>

Example

```
> show authentication vserver Auth-Vserver-2
    Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
    State: UP
    Client Idle Timeout: 180 sec
    Down state flush: DISABLED
    Disable Primary Vserver On Down : DISABLED
    Authentication : ON
    Current AAA Users: 0
    Authentication Domain: myCompany.employee.com
 Done
```

To verify your AAA virtual server setup by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Virtual Servers.
2. Review the information in the AAA Virtual Servers pane to verify that your configuration is correct and your authentication virtual server is accepting traffic. You can select a specific virtual server to view detailed information in the details pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating an Authentication Profile

Nov 03, 2015

When you want the same authentication settings to be used by multiple traffic management virtual servers, you can create an authentication profile which specifies the authentication virtual server, the authentication host, the authentication domain, and authentication level.

This authentication profile can be associated with the relevant traffic management virtual servers.

To configure an authentication profile by using the NetScaler CLI

1. Create the authentication profile and set the required parameters.

   For example, to create a profile with an authentication virtual server named "authVS".

   ns-cli-prompt> **add authentication authnProfile** authProfile1 -authnVsName authVS -authenticationHost authnVS.example.com -authenticationDomain example.com -authenticationLevel

   Note: The authentication weight or level will depend on the virtual server to which the traffic will bound. A session that is created by authenticating against traffic management virtual server at given level cannot be used to access traffic management virtual server at a higher level.

2. Bind the authentication profile to the relevant traffic management virtual servers.

   For example, to bind authProfile1 to a load balancing virtual server named "vserver1".

   ns-cli-prompt> **set lb vserver** vserver1 -authnProfile authProfile1

To configure an authentication profile by using the NetScaler GUI

In the **Configuration** tab, navigate to **Security > AAA - Application Traffic > Authentication Profile**, and configure the authentication profile as required.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Users and Groups

Aug 19, 2014

After configuring the AAA basic setup, you create users and groups. You first create a user account for each person who will authenticate via the NetScaler appliance. If you are using local authentication controlled by the NetScaler appliance itself, you create local user accounts and assign passwords to each of those accounts.

You also create user accounts on the NetScaler appliance if you are using an external authentication server. In this case, however, each user account must exactly match an account for that user on the external authentication server, and you do not assign passwords to the user accounts that you create on the NetScaler. The external authentication server manages the passwords for users that authenticate with the external authentication server.

If you are using an external authentication server, you can still create local user accounts on the NetScaler appliance if, for example, you want to allow temporary users (such as visitors) to log in but do not want to create entries for those users on the authentication server. You assign a password to each local user account, just as you would if you were using local authentication for all user accounts.

Each user account must be bound to policies for authentication and authorization. To simplify this task, you can create one or more groups and assign user accounts to them. You can then bind policies to groups instead of individual user accounts.

To create a local AAA user account by using the command line interface

At the command prompt, type the following commands to create a local AAA user account and verify the configuration:

- add aaa user <username> [–password <password>]
- show aaa user

Example

```
> add aaa user user-2 -password emptybag
 Done
> show aaa user
1)	UserName: user-1
2)	UserName: user-2
 Done
```

To change the password for an existing AAA local user account by using the command line interface

At the command prompt, type the following command and, when prompted, type the new password:
set aaa user <username>

Example

```
> set aaa user user-2
Enter password:
Done
```

To configure AAA local users by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Users
2. In the details pane, do one of the following:
   - To create a new user account, click Add.
   - To modify an existing user account, select the user account, and then click Open.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Create AAA User dialog box, in the User Name text box, type a name for the user.
4. If creating a locally authenticated user account, clear the External Authentication check box and provide a local password that the user will use to log on.
5. Click Create or OK, and then click Close. A message appears in the status bar, stating that the user has been configured successfully.

To create AAA local groups and add users to them by using the command line interface

At the command prompt, type the following commands. Type the first command one time, and type the second command once for each user:

- add aaa group <groupname>
- show aaa group

  Example

  ```
  > add aaa group group-2
  Done
  > show aaa group
  1)    GroupName: group-1
  2)    GroupName: group-2
  Done
  ```

- bind aaa group <groupname> -username <username>

  Example

  ```
  > bind aaa group group-2 -username user-2
   Done
  > show aaa group group-2
        GroupName: group-2

              UserName: user-2
   Done
  ```

To remove users from an AAA group by using the command line interface

At the command prompt, unbind users from the group by typing the following command once for each user account that is bound to the group:

unbind aaa group <groupname> -username <username>

Example

```
> unbind aaa group group-hr -username user-hr-1
 Done
```

To remove an AAA group by using the command line interface

First remove all users from the group. Then, at the command prompt, type the following command to remove an AAA group and verify the configuration:

- rm aaa group <groupname>
- show aaa group

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

```
> rm aaa group group-hr
 Done
> show aaa group
1)	GroupName: group-1
2)	GroupName: group-finance
 Done
```

To configure AAA local groups and add users to them by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Groups
2. In the details pane, do one of the following:
   - To create a new group, click Add.
   - To modify an existing group, select the group, and then click Edit.
3. If you are creating a new group, in the Create AAA Group dialog box, in the Group Name text box, type a name for the group.
4. In the Advanced area to the right, click AAA Users.
   1. To add a user to the group, select the user, and then click Add.
   2. To remove a user from the group, select the user, and then click Remove.
   3. To create a new user account and add it to the group, click the Plus icon, and then follow the instructions in “To configure AAA local users by using the configuration utility.”
5. Click Create or OK. The group that you created appears in the AAA Groups page.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring AAA Policies

Sep 18, 2013

After you set up your users and groups, you next configure authentication policies, authorization policies, and audit policies to define which users are allowed to access your intranet, which resources each user or group is allowed to access, and what level of detail AAA will preserve in the audit logs. An authentication policy defines the type of authentication to apply when a user attempts to log on. If external authentication is used, the policy also specifies the external authentication server. Authorization policies specify the network resources that users and groups can access after they log on. Auditing policies define the audit log type and location.

You must bind each policy to put it into effect. You bind authentication policies to authentication virtual servers, authorization policies to one or more user accounts or groups, and auditing policies both globally and to one or more user accounts or groups.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler operating system, policy priorities work in reverse order: the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000. The AAA feature implements only the first of each type of policy that a request matches, not any additional policies of that type that a request might also match, so policy priority is important for getting the results you intend.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you bind the policies. You can then add additional policies at any time without having to reassign the priority of an existing policy.

For additional information about binding policies on the NetScaler, see the *Citrix NetScaler Traffic Management Guide* at "Traffic Management."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Authentication Policies

Aug 19, 2014

The NetScaler ADC can authenticate users with local user accounts or by using an external authentication server. The appliance supports the following authentication types:

LOCAL

Authenticates to the NetScaler by using a password, without reference to an external authentication server. User data is stored locally on the NetScaler appliance.

RADIUS

Authenticate to an external Radius server.

LDAP

Authenticates to an external LDAP authentication server.

TACACS

Authenticates to an external Terminal Access Controller Access-Control System (TACACS) authentication server.

After a user authenticates to a TACACS server, the NetScaler ADC connects to the same TACACS server for all subsequent authorizations. When a primary TACACS server is unavailable, this feature prevents delays while the ADC waits for the first TACACS server to time out before resending the authorization request to the second TACACS server.

Note: When authenticating through a TACACS server, AAA-TM logs only successfully executed TACACS commands, to prevent the logs from showing TACACS commands that were entered by users who were not authorized to execute them.

CERT

Authenticates to the NetScaler appliance by using a client certificate, without reference to an external authentication server.

NEGOTIATE

Authenticates to a Kerberos authentication server. If there is an error in Kerberos authentication, NetScaler uses NTLM authentication.

SAML

Authenticates to a server that supports the Security Assertion Markup Language (SAML).

SAMLIDP

Configures the NetScaler ADC to serve as a Security Assertion Markup Language (SAML) Identity Provider (IdP).

WEB

Authenticates to a web server, providing the credentials that the web server requires in an HTTP request and analyzing the web server response to determine that user authentication was successful.

An authentication policy is comprised of an expression and an action. Authentication policies use NetScaler expressions.

After creating an authentication action and an authentication policy, bind it to an authentication virtual server and assign a priority to it. When binding it, also designate it as either a primary or a secondary policy. Primary policies are evaluated before secondary policies. In configurations that use both types of policy, primary policies are normally more specific policies while secondary policies are normally more general policies intended to handle authentication for any user accounts that do not meet the more specific criteria.

To add an authentication action by using the command line interface

If you do not use LOCAL authentication, you need to add an explicit authentication action. To do this, at the command prompt, type the following command:

```
add authentication tacacsAction <name> -serverip <IP> [-serverPort <port>] [-authTimeout <positive_integer>] [ ... ]
```

Example

```
> add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -serverport 1812 -authtimeout 15 -tacacsSecret "minotaur" -authorization OFF -accounting ON -auditFailedCmds
```

To configure an authentication action by using the command line interface

To configure an existing authentication action, at the command prompt, type the following command:

```
set authentication tacacsAction <name> -serverip <IP> [-serverPort <port>] [-authTimeout <positive_integer>] [ ... ]
```

Example

```
> set authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -serverport 1812 -authtimeout 15 -tacacsSecret "minotaur" -authorization OFF -accounting ON -auditFailedCmds (
```

To remove an authentication action by using the command line interface

To remove an existing RADIUS action, at the command prompt, type the following command:

```
rm authentication radiusAction <name>
```

Example

```
> rm authentication tacacsaction Authn-Act-1  Done
```

To configure an authentication server by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to Security > AAA - Application Traffic > Policies > Authentication.
2. In the details pane, on the Servers tab, do one of the following:
   - To create a new authentication server, click Add.
   - To modify an existing authentication server, select the server, and then click Open.
3. In the Create Authentication Server or Configure Authentication Server dialog box, type or select values for the parameters.
   - Name*—radiusActionName (Cannot be changed for a previously configured action)
   - Authentication Type*—authtype (Set to RADIUS, cannot be changed)
   - IP Address*—serverip <IP>
   - IPV6*—Select the checkbox if the server IP is an IPv6 IP. (No command line equivalent.)
   - Port*—serverPort
   - Time-out (seconds)*—authTimeout
4. Click Create or OK, and then click Close. The policy that you created appears in the Authentication Policies and Servers page.

To create and bind an authentication policy by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands in the order shown to create and bind an authentication policy and verify the configuration:

- add authentication negotiatePolicy <name> <rule> <reqAction>
- show authentication localPolicy <name>
- bind authentication vserver <name> -policy <policyname> [-priority <priority>] [-secondary]]
- show authentication vserver <name>

Example

```
> add authentication localPolicy Authn-Pol-1 ns_true
Done
> show authentication localPolicy
1)	Name: Authn-Pol-1	Rule: ns_true	Request action: LOCAL
Done
> bind authentication vs
```

To modify an existing authentication policy by using the command line interface

At the command prompt, type the following commands to modify an existing authentication policy:

set authentication localPolicy <name> <rule> [-reqaction <action>]

Example

```
> set authentication localPolicy Authn-Pol-1 'ns_true'
Done
```

To remove an authentication policy by using the command line interface

At the command prompt, type the following command to remove an authentication policy:

rm authentication localPolicy <name>

Example

```
> rm authentication localPolicy Authn-Pol-1
Done
```

To configure and bind authentication policies by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Policies > Authentication, and then select the type of policy that you want to create.
2. In the details pane, on the Policies tab, do one of the following:
   - To create a new policy, click Add.
   - To modify an existing policy, select the action, and then click Edit.
3. In the Create Authentication Policy or Configure Authentication Policy dialog, type or select values for the parameters.
   - Name*—policyname (Cannot be changed for a previously configured action)
   - Authentication Type*—authtype
   - Server*—authVsName
   - Expression*—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression window, and then by typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
4. Click Create or OK. The policy that you created appears in the Policies page.
5. Click the Servers tab, and in the details pane do one of the following:
   - To use an existing server, select it, and then click .
   - To create a new server, click Add, and follow the instructions.
6. If you want to designate this policy as a secondary authentication policy, on the Authentication tab, click Secondary. If you want to designate this policy as a primary authentication policy, skip this step.
7. Click Insert Policy.
8. Choose the policy you want to bind to the authentication virtual server from the drop-down list.
9. In the Priority column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
10. Click OK. A message appears in the status bar, stating that the policy has been configured successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

LDAP Authentication Policies

Aug 13, 2014

As with other types of authentication policies, a Lightweight Directory Access Protocol (LDAP) authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. In addition to standard authentication functions, LDAP can search other active directory (AD) servers for user accounts for users that do not exist locally. This function is called referral support or referral chasing.

Normally you configure the NetScaler ADC to use the IP address of the authentication server during authentication. With LDAP authentication servers, you can also configure the ADC to use the FQDN of the LDAP server instead of its IP address to authenticate users. Using an FQDN can simplify an otherwise much more complex AAA configuration in environments where the authentication server might be at any of several IP addresses, but always uses a single FQDN. To configure authentication by using a server's FQDN instead of its IP address, you follow the normal configuration process except when creating the authentication action. When creating the action, you use the **serverName** parameter instead of the **serverIP** parameter, and substitute the server's FQDN for its IP address.

Before you decide whether to configure the ADC to use the IP or the FQDN of your LDAP server to authenticate users, consider that configuring AAA to authenticate to an FQDN instead of an IP address adds an extra step to the authentication process. Each time the ADC authenticates a user, it must resolve the FQDN. If a great many users attempt to authenticate simultaneously, the resulting DNS lookups might slow the authentication process.

LDAP referral support is disabled by default and cannot be enabled globally. It must be explicitly enabled for each LDAP action. You must also make sure that the AD server accepts the same binddn credentials that are used with the referring (GC) server. To enable referral support, you configure an LDAP action to follow referrals, and specify the maximum number of referrals to follow.

If referral support is enabled, and the NetScaler ADC receives an LDAP_REFERRAL response to a request, AAA follows the referral to the active directory (AD) server contained in the referral and performs the update on that server. First, AAA looks up the referral server in DNS, and connects to that server. If the referral policy requires SSL/TLS, it connects via SSL/TLS. It then binds to the new server with the binddn credentials that it used with the previous server, and performs the operation which generated the referral. This feature is transparent to the user.

Note: These instructions assume that you are already familiar with the LDAP protocol and have already configured your chosen LDAP authentication server.
For more information about setting up authentication policies in general, see Authentication Policies. For more information about NetScaler expressions, which are used in the policy rule, see Policies and Expressions.

To enable LDAP referral support by using the command line interface

At the command prompt, type the following commands:

- set authentication ldapAction <name> -followReferrals ON
- set authentication ldapAction <name> -maxLDAPReferrals <integer>

Example

```
> set authentication ldapAction ldapAction-1 -followReferrals ON
set authentication ldapAction ldapAction-1 -maxLDAPReferrals 2
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To enable LDAP referral support by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to Security > AAA - Application Traffic > Policies > LDAP.
2. In the details pane, on the Servers tab, select the LDAP server that you want to configure, and then click Edit.
3. In the Configure Authentication Server dialog, scroll down to the Referrals check box, and select it.
4. In the Maximum Referral Level text box, type the maximum number of referrals to allow.
5. Click OK, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Negotiate Authentication Policies

Nov 02, 2015

As with other types of authentication policies, a Negotiate authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy.

In addition to standard authentication functions, the Negotiate Action command can now extract user information from a keytab file instead of requiring you to enter that information manually. If a keytab has more than one SPN, AAA selects the correct SPN. You can configure this feature at the NetScaler command line, or by using the configuration utility.

Note: These instructions assume that you are already familiar with the LDAP protocol and have already configured your chosen LDAP authentication server.

To configure AAA to extract user information from a keytab file by using the command line interface

At the command prompt, type the appropriate command:

- add authentication negotiateAction <name> [-keytab <string>]
- set authentication negotiateAction <name> [-keytab <string>]

Example

```
> set authentication negotiateAction negotiateAction-1 -keytab keytab-1
```

To configure AAA to extract user information from a keytab file by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to Security > AAA - Application Traffic > Policies > Negotiate.
2. In the details pane, on the Servers tab, do one of the following:
   - If you want to create a new Negotiate action, click Add.
   - If you want to modify an existing Negotiate action, in the data pane select the action, and then click Edit.
3. If you are creating a new Negotiate action, in the Name text box, type a name for your new action. The name can be from one to 127 characters in length and can consist of upper- and lowercase letters, numbers, and the hyphen (-) and underscore (_) characters. If you are modifying an existing Negotiate action, skip this step. The name is read-only; you cannot change it.
4. Under Negotiate, if the Use Keytab file check box is not already checked, check it.
5. In the Keytab file path text box, type the full path and filename of the keytab file that you want to use.
6. In the Default authentication group text box, type the authentication group that you want to set as default for this user.
7. Click Create or OK to save your changes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

RADIUS Authentication Policies

Aug 21, 2014

As with other types of authentication policies, a Remote Authentication Dial In User Service (RADIUS) authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. However, setting up a RADIUS authentication policy has certain special requirements that are described below.

Normally you configure the NetScaler ADC to use the IP address of the authentication server during authentication. With RADIUS authentication servers, you can now configure the ADC to use the FQDN of the RADIUS server instead of its IP address to authenticate users. Using an FQDN can simplify an otherwise much more complex AAA configuration in environments where the authentication server might be at any of several IP addresses, but always uses a single FQDN. To configure authentication by using a server's FQDN instead of its IP address, you follow the normal configuration process except when creating the authentication action. When creating the action, you substitute the **serverName** parameter for the **serverIP** parameter.

Before you decide whether to configure the ADC to use the IP or the FQDN of your RADIUS server to authenticate users, consider that configuring AAA to authenticate to an FQDN instead of an IP address adds an extra step to the authentication process. Each time the ADC authenticates a user, it must resolve the FQDN. If a great many users attempt to authenticate simultaneously, the resulting DNS lookups might slow the authentication process.

Note: These instructions assume that you are already familiar with the RADIUS protocol and have already configured your chosen RADIUS authentication server.
For more information about setting up authentication policies in general, see Authentication Policies. For more information about NetScaler expressions, which are used in the policy rule, see Policies and Expressions.

To add an authentication action for a RADIUS server by using the command line interface

If you authenticate to a RADIUS server, you need to add an explicit authentication action. To do this, at the command prompt, type the following command:

add authentication radiusAction <name> [-serverip <IP> | -serverName] <FQDN>] [-serverPort <port>] [-authTimeout <positive_integer>] {-radKey } [-radNASip (ENABLED | DISABLED)] [-radNASid <string>] [-radVendorID <positive_integer>] [-radAttributeType <positive_integer>] [-radGroupsPrefix <string>] [-radGroupSeparator <string>] [-passEncoding <passEncoding>] [-ipVendorID <positive_integer>] [-ipAttributeType <positive_integer>] [-accounting (ON | OFF)] [-pwdVendorID <positive_integer> [-pwdAttributeType <positive_integer>]] [-defaultAuthenticationGroup <string>] [-callingstationid (ENABLED | DISABLED)]

Example

The following example adds a RADIUS authentication action named **Authn-Act-1**, with the server IP **10.218.24.65**, the server port **1812**, the authentication timeout **15** minutes, the radius key **WareTheLorax**, NAS IP disabled, and NAS ID **NAS1**.

```
> add authentication radiusaction Authn-Act-1 -serverip 10.218.24.65 -serverport 1812
        -authtimeout 15 -radkey WareTheLorax -radNASip DISABLED -radNASid NAS1
Done
```

The following example adds the same RADIUS authentication action, but using the server FQDN **rad01.example.com** instead of the IP.

```
> add authentication radiusaction Authn-Act-1 -serverName rad01.example.com -serverport 1812
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
-authtimeout 15 -radkey WareTheLorax -radNASip DISABLED -radNASid NAS1
Done
```

To configure an authentication action for an external RADIUS server by using the command line

To configure an existing RADIUS action, at the NetScaler command prompt, type the following command:

set authentication radiusAction <name> [-serverip <IP> | -serverName] <FQDN>] [-serverPort <port>] [-authTimeout <positive_integer>] {-radKey } [-radNASip (ENABLED | DISABLED)] [-radNASid <string>] [-radVendorID <positive_integer>] [-radAttributeType <positive_integer>] [-radGroupsPrefix <string>] [-radGroupSeparator <string>] [-passEncoding <passEncoding>] [-ipVendorID <positive_integer>] [-ipAttributeType <positive_integer>] [-accounting (ON | OFF)] [-pwdVendorID <positive_integer> [-pwdAttributeType <positive_integer>]] [-defaultAuthenticationGroup <string>] [-callingstationid (ENABLED | DISABLED)]

To remove an authentication action for an external RADIUS server by using the command line interface

To remove an existing RADIUS action, at the command prompt, type the following command:

rm authentication radiusAction <name>

Example

```
> rm authentication radiusaction Authn-Act-1
Done
```

To configure a RADIUS server by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to Security > AAA - Application Traffic > Policies > Authentication > Radius
2. In the details pane, on the Servers tab, do one of the following:
   - To create a new RADIUS server, click Add.
   - To modify an existing RADIUS server, select the server, and then click Edit.
3. In the Create Authentication RADIUS Server or Configure Authentication RADIUS Server dialog, type or select values for the parameters. To fill out parameters that appear beneath Send Calling Station ID, expand Details.
   - Name*—radiusActionName (Cannot be changed for a previously configured action)
   - Authentication Type*—authtype (Set to RADIUS, cannot be changed)
   - Server Name / IP Address*—Choose either Server Name or Server IP
     - Server Name*—serverName <FQDN>
     - IP Address*—serverIp <IP> If the server is assigned an IPv6 IP address, select the IPv6 check box.
   - Port*—serverPort
   - Time-out (seconds)*—authTimeout
   - Secret Key*—radKey (RADIUS shared secret.)
   - Confirm Secret Key*—Type the RADIUS shared secret a second time. (No command line equivalent.)
   - Send Calling Station ID—callingstationid
   - Group Vendor Identifier—radVendorID
   - Group Attribute Type—radAttributeType
   - IP Address Vendor Identifier—ipVendorID
   - pwdVendorID—pwdVendorID
   - Password Encoding—passEncoding
   - Default Authentication Group—defaultAuthenticationGroup
   - NAS ID—radNASid
   - Enable NAS IP address extraction—radNASip
   - Group Prefix—radGroupsPrefix

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- Group Separator—radGroupSeparator
- IP Address Attribute Type—ipAttributeType
- Password Attribute Type—pwdAttributeType
- Accounting—accounting

4. Click Create or OK. The policy that you created appears in the Servers page.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SAML Authentication Policies

Jul 23, 2015

For information on NetScaler as a SAML SP and IdP, see SAML Authentication.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SAML IDP Authentication Policies

Jul 23, 2015

For information on NetScaler as a SAML SP and IdP, see SAML Authentication.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Web Authentication Policies

Aug 19, 2014

AAA-TM is now able to authenticate a user to a web server, providing the credentials that the web server requires in an HTTP request and analyzing the web server response to determine that user authentication was successful. As with other types of authentication policies, a Web authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy.

To set up web-based authentication with a specific web server, first you create a web authentication action. Since authentication to web servers does not use a rigid format, you must specify exactly which information the web server requires and in which format when creating the action. To do this, you create an expression in NetScaler default syntax that contains the following items:

- **Server IP**—The IP address of the authentication Web server.
- **Server Port**—The port of the authentication Web server.
- **Authentication Rule**—An expression in NetScaler default syntax that contains the user's credentials in the format that the Web server expects.
- **Scheme**—HTTP (for unencrypted web authentication) or HTTPS (for encrypted web authentication).
- **Success Rule**—An expression in NetScaler default syntax that matches the web server response string that signifies that the user authenticated successfully.

For all other parameters, follow the normal rules for the add authentication action command.

Next you create a policy associated with that action. The policy is similar to an LDAP policy, and like LDAP policies uses NetScaler classic syntax.

Note: These instructions assume that you are already familiar with the authentication requirements of the web server(s) to which you want to authenticate, and have already configured the web authentication server.

To configure a Web authentication action by using the command line interface

To create a web authentication action at the command line, at the command line type the following command:

add authentication webAuthAction <name> -serverIP <ip_addr|ipv6_addr|*> -serverPort <port|*> [-fullReqExpr <string>] -scheme (http | https) -successRule <expression> [-defaultAuthenticationGroup <string>] [-Attribute1 <string>] [-Attribute2 <string>] [-Attribute3 <string>] [-Attribute4 <string>] [-Attribute5 <string>] [-Attribute6 <string>] [-Attribute7 <string>] [-Attribute8 <string>] [-Attribute9 <string>] [-Attribute10 <string>] [-Attribute11 <string>] [-Attribute12 <string>] [-Attribute13 <string>] [-Attribute14 <string>] [-Attribute15 <string>] [-Attribute16 <string>]

Example

```
> add authentication webAuthAction webauth1 -serverIP 10.214.56.31 -serverPort 80 -
```

To configure a Web authentication action by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to Security > AAA - Application Traffic > Policies > LDAP.
2. In the details pane, on the Servers tab, do one of the following:
   - If you want to create a new web authentication action, click Add.
   - If you want to modify an existing web authentication action, in the data pane select the action, and then click Edit.
3. If you are creating a new web authentication action, in the Create Authentication Web server dialog box, Name text

© 1999-2017 Citrix Systems, Inc. All rights reserved.

box, type a name for the new web authentication action. The name can be from one to 127 characters in length, and can consist of upper- and lowercase letters, numbers, and the hyphen (-) and underscore (_) characters. If you are modifying an existing web authentication action, skip this step. The name is read-only; you cannot change it.

4. In the Web Server IP Address text box, type the IPv4 or IPv6 IP address of the authentication web server. If the address is an IPv6 IP address, select the IPv6 check box first.
5. In the Port text box, type the port number on which the web server accepts connections.
6. Select HTTP or HTTPS in the Protocol drop-down list.
7. In the HTTP Request Expression text area, type a PCRE-format regular expression that creates the web server request that contains the user's credentials in the exact format expected by the authentication web server.
8. In the Expression to validate the Authentication text area, type a NetScaler default syntax expression that describes the information in the web server response that indicates that user authentication was successful.
9. Fill out the remaining fields as described in the general authentication action documentation.
10. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Advanced Authentication Policies

Jun 11, 2014

If you know exactly how you want an authentication policy to be configured, you can use the advanced authentication policy dialog to create the policy quickly.

To configure an advanced authentication policy by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies, and then select Policy.
2. In the details pane do one of the following:
   - To create a new policy, click Add.
   - To modify an existing policy, select the policy, and then click Edit.
3. In the Create Authentication Policy or Configure Authentication Policy dialog box, type or select values for the parameters.
   - Name*—The policy name. Cannot be changed for a previously configured policy.
   - Action Type*—The policy type: Cert, Negotiate, LDAP, RADIUS, SAML, SAMLIDP, TACACS, or WEBAUTH.
   - Action*—The authentication action (profile) to associate with the policy. You can choose an existing authentication action, or click the plus and create a new action of the proper type.
   - Log Action—The audit action to associate with the policy. You can choose an existing audit action, or click the plus and create a new action.
   - Expression*—The rule that selects connections to which you want to apply the action that you specified. The rule can be simple ("true" selects all traffic) or complex. You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression window, and then by typing your expression directly into the expression text area, or by clicking Add to open Add Expression dialog box and using the drop-down lists in it to construct your expression.)
   - Comment—You can type a comment that describes the type of traffic that this authentication policy will apply to. Optional.
4. Click Create or OK, and then click Close. If you created a policy, that policy appears in the Authentication Policies and Servers page.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Authorizing User Access to Application Resources

Nov 23, 2015

You can control the resources that an authenticated user can access within an application.

To do this, associate an authorization policy to each of the users, either individually or by associating the policy to a group of users. The authorization policy must specify the following:

- **Rule.** The resource to which access must be authorized. This can be specified by using basic or advanced expressions.
- **Action.** Whether access to the resource must be allowed or denied.

For examples, see Sample Authorization Configurations.

By default, access to all resources within an application is **DENIED** to all users. However, you can change this default authorization action to **ALLOW** access to all users (by setting the session parameters in session profile or by setting the global session parameters).

> ## Warning
>
> For optimum security, Citrix recommends that you do not to change the default authorization action from DENY to ALLOW. Instead, it is advised to create specific authorization policies for users who need access to specific resources.

To configure authorization by using the NetScaler CLI

1. Configure the authorization policy.

   ns-cli-prompt> **add authorization policy <name> <rule> <action>**

2. Associate the policy with the appropriate user or group.

   - Bind the policy to a specific user.
     ns-cli-prompt> **bind aaa user <username> -policy** <policyname>

   - Bind the policy to a specific group.
     ns-cli-prompt> **bind aaa group <groupName> -policy** <policyname>

To configure authorization by using the NetScaler GUI (Configuration tab)

1. Create the authorization policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Authorization**, click **Add** and then define the policy as required.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Associate the policy with the appropriate user or group.

   Navigate to **Security > AAA - Application Traffic > Users** or **Groups**, and edit the relevant user or group to associate it with the authorization policy.

Sample Authorization Configurations

Here are some example configurations to authorize user access to some application resources. Note that these are CLI commands. You can do similar configurations using the GUI, although you must not enclose the expression within quotes (").

Example 1: Allow "user1" access to URLs that have the suffix "gif" COPY

```
> add authorization policy authzpol1 "HTTP.REQ.URL.SUFFIX.EQ(\"gif\")" ALLOW

> bind aaa user user1 -policy authzpol1
```

Example 2: Deny users of "group1" access to URLs that have the suffix "png" COPY

```
> add authorization policy authzpol2 "HTTP.REQ.URL.SUFFIX.EQ(\"png\")" DENY

> bind aaa group group1 -policy authzpol2
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Auditing Authenticated Sessions

Nov 23, 2015

You can configure the NetScaler appliance to keep a log of all the events that are triggered in an authenticated session. Using this information, you can audit state and status information, to see the history for users in chronological order.

To do this, define an audit policy that specifies the following:

- **Log type.** The logs can be stored remotely (syslog) or locally on the NetScaler appliance (nslog).
- **Rule.** The conditions on which the logs are stored.
- **Action.** Details of the log server and other details for creating the log entries.

This audit policy can be configured at different levels: user-level, group-level, AAA virtual server, and global system level. The policies configured at the user-level have the highest priority.

> ## Note
>
> This topic details steps for using syslog. Make necessary changes to use nslog.

To configure syslog auditing by using the NetScaler CLI

1. Configure the audit server with the relevant log settings.

   ns-cli-prompt> **add audit syslogAction <name> <serverIP> ...**

2. Configure the audit policy by associating the audit server.

   ns-cli-prompt> **add audit syslogPolicy <name> <rule> <action>**

3. Associate the audit policy with one of the following entities:

   - Bind the policy to a specific user.

     ns-cli-prompt> **bind aaa user <userName> -policy** <policyname> ...

   - Bind the policy to a specific group.

     ns-cli-prompt> **bind aaa group <groupName> -policy** <policyname> ...

   - Bind the policy to a AAA virtual server.

     ns-cli-prompt> **bind authentication vserver <name> -policy** <policyname> ...

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Bind the policy globally to the NetScaler.

  ns-cli-prompt> **bind tm global** -**policyName** <policyname> ...

To configure syslog auditing by using the NetScaler GUI (Configuration tab)

1. Configure the audit server and policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Auditing > Syslog**, and configure the server and the policy in the relevant tabs.

2. Associate the policy with one of the following:

   - Bind the policy to a specific user.

     Navigate to **Security > AAA - Application Traffic > Users**, and associate the authorization policy with the relevant user.

   - Bind the policy to a specific group.

     Navigate to **Security > AAA - Application Traffic > Groups**, and associate the authorization policy with the relevant group.

   - Bind the policy to a AAA virtual server.

     Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the authorization policy with the relevant virtual server.

   - Bind the policy globally to the NetScaler.

     Navigate to **Security > AAA - Application Traffic > Policies > Auditing > Syslog** or **Nslog**, select the authorization policy, and click **Action > Global Bindings** to bind the policy globally.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Session Settings

Apr 18, 2013

After you configure your authentication, authorization, and auditing profiles, you configure session settings to customize your user sessions. The session settings are:

The session timeout.

Controls the period after which the user is automatically disconnected and must authenticate again to access your intranet.

The default authorization setting.

Determines whether the NetScaler appliance will by default allow or deny access to content for which there is no specific authorization policy.

The single sign-on setting.

Determines whether the NetScaler appliance will log users onto all web applications automatically after they authenticate, or will pass users to the web application logon page to authenticate for each application.

The credential index setting.

Determines whether the NetScaler appliance will use primary or the secondary authentication credentials for single signon.

To configure the session settings, you can take one of two approaches. If you want different settings for different user accounts or groups, you create a profile for each user account or group for which you want to configure custom sessions settings. You also create policies to select the connections to which to apply particular profiles, and you bind the policies to users or groups. You can also bind a policy to the authentication virtual server that handles the traffic to which you want to apply the profile.

If you want the same settings for all sessions, or if you want to customize the default settings for sessions that do not have specific profiles and policies configured, you can simply configure the global session settings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Session Profiles

Aug 19, 2014

To customize your user sessions, you first create a session profile. The session profile allows you to override global settings for any of the session parameters.

Note: The terms “session profile” and “session action” mean the same thing.

To create a session profile by using the command line interface

At the command prompt, type the following commands to create a session profile and verify the configuration:

- add tm sessionAction <name> [-sessTimeout <mins>] [-defaultAuthorizationAction (ALLOW | DENY)] [-SSO (ON | OFF)] [-ssoCredential (PRIMARY | SECONDARY)] [-ssoDomain <string>] [-httpOnlyCookie (YES | NO)] [-persistentCookie (ENABLED | DISABLED)] [-persistentCookieValidity <minutes>]
- show tm sessionAction <name>

Example

```
> add tm sessionAction session-profile -sessTimeout 30 -defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)     Name: session-profile
       Authorization action : ALLOW
       Session timeout: 30 minutes
 Done
```

To modify a session profile by using the command line interface

At the command prompt, type the following commands to modify a session profile and verify the configuration:

- set tm sessionAction <name> [-sessTimeout <mins>] [-defaultAuthorizationAction (ALLOW | DENY)] [-SSO (ON | OFF)] [-ssoCredential (PRIMARY | SECONDARY)] [-ssoDomain <string>] [-httpOnlyCookie (YES | NO)] [-persistentCookie (ENABLED | DISABLED)] [-persistentCookieValidity <minutes>]
- show tm sessionAction

Example

```
> set tm sessionAction session-profile -sessTimeout 30 -defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)     Name: session-profile
       Authorization action : ALLOW
       Session timeout: 30 minutes
 Done
```

To remove a session profile by using the command line interface

At the command prompt, type the following command to remove a session profile:

rm tm sessionAction <name>

To configure session profiles by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Session.
2. Navigate to Security > AAA - Application Traffic > Policies > Session.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the details pane, click the Profiles tab.
4. On the Profiles tab, do one of the following:
   - To create a new session profile, click Add.
   - To modify an existing session profile, select the profile, and then click Edit.
5. In the Create TM Session Profile or Configure TM Session Profile dialog, type or select values for the parameters.
   - Name*—actionname (Cannot be changed for a previously configured session action.)
   - Session Time-out—sesstimeout
   - Default Authorization Action—defaultAuthorizationAction
   - Single Signon to Web Applications—sso
   - Credential Index—ssocredential
   - Single Sign-on Domain—ssoDomain
   - HTTPOnly Cookie—httpOnlyCookie
   - Enable Persistent Cookie—persistentCookie
   - Persistent Cookie Validity—persistentCookieValidity
6. Click Create or OK. The session profile that you created appears in the Session Policies and Profiles pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Session Policies

Aug 19, 2014

After you create one or more session profiles, you create session policies and then bind the policies globally or to an authentication virtual server to put them into effect.

To create a session policy by using the command line interface

At the command prompt, type the following commands to create a session policy and verify the configuration:

- add tm sessionPolicy <name> <rule> <action>
- show tm sessionPolicy <name>

Example

```
> add tm sessionPolicy session-pol "URL == /*.gif" session-profile
 Done
> show tm sessionPolicy session-pol
1)     Name: session-pol       Rule: URL == '/*.gif'
       Action: session-profile
 Done
```

To modify a session policy by using the command line interface

At the command prompt, type the following commands to modify a session policy and verify the configuration:

- set tm sessionPolicy <name> [-rule <expression>] [-action <action>]
- show tm sessionPolicy <name>

Example

```
> set tm sessionPolicy session-pol "URL == /*.gif" session-profile
 Done
> show tm sessionPolicy session-pol
1)     Name: session-pol       Rule: URL == '/*.gif'
       Action: session-profile
 Done
```

To globally bind a session policy by using the command line interface

At the command prompt, type the following commands to globally bind a session policy and verify the configuration:

bind tm global -policyName <policyname> [-priority <priority>]

Example

```
> bind tm global -policyName session-pol
 Done

> show tm sessionPolicy session-pol
1)     Name: session-pol       Rule: URL == '/*.gif'
       Action: session-profile
       Policy is bound to following entities
       1) TM GLOBAL    PRIORITY : 0
 Done
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

To bind a session policy to an authentication virtual server by using the command line interface

At the command prompt, type the following command to bind a session policy to an authentication virtual and verify the configuration:
bind authentication vserver <name> -policy <policyname> [-priority <priority>]
Example

```
> bind authentication vserver auth-vserver-1 -policyName Session-Pol-1 -priority 1000
Done
```

To unbind a session policy from an authentication virtual server by using the command line interface

At the command prompt, type the following commands to unbind a session policy from an authentication virtual server and verify the configuration:
unbind authentication vserver <name> -policy <policyname>
Example

```
> unbind authentication vserver auth-vserver-1 -policyName Session-Pol-1
Done
```

To unbind a globally bound session policy by using the command line interface

At the command prompt, type the following commands to unbind a globally-bound session policy:
unbind tm global -policyName <policyname>
Example

```
> unbind tm global -policyName Session-Pol-1
Done
```

To remove a session policy by using the command line interface

First unbind the session policy from global, and then, at the command prompt, type the following commands to remove a session policy and verify the configuration:
rm tm sessionPolicy <name>
Example

```
> rm tm sessionPolicy Session-Pol-1
Done
```

To configure and bind session policies by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Session.
2. Navigate to Security > AAA - Application Traffic > Policies > Session.
3. In the details pane, on the Policies tab, do one of the following:
   - To create a new session policy, click Add.
   - To modify an existing session policy, select the policy, and then click Edit.
4. In the Create Session Policy or Configure Session Policy dialog, type or select values for the parameters.
   - Name*—policyname (Cannot be changed for a previously configured session policy.)
   - Request Profile*—actionname
   - Expression*—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression text area and then typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
5. Click Create or OK. The policy that you created appears in the details pane of the Session Policies and Profiles page.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

6. To globally bind a session policy, in the details pane, select Global Bindings from the Action drop-down list, and fill in the dialog.
   1. Select the name of the session policy you want to globally bind.
   2. Click OK.
7. To bind a session policy to an authentication virtual server, in the navigation pane, click Virtual Servers, and add that policy to the policies list.
   1. In the details pane, select the virtual server, and then click Edit.
   2. In the Advanced selections to the right of the detail area, click Policies.
   3. Select a policy, or click the plus icon to add a policy.
   4. In the Priority column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
   5. Click OK.

   A message appears in the status bar, stating that the policy has been configured successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Global Session Settings

Aug 13, 2014

In addition to or instead of creating session profiles and policies, you can configure global session settings. These settings control the session configuration when there is no explicit policy overriding them.

To configure the session settings by using the command line interface

At the command prompt, type the following commands to configure the global session settings and verify the configuration:

set tm sessionParameter [-sessTimeout <mins>] [-defaultAuthorizationAction (ALLOW | DENY)] [-SSO (ON | OFF)] [-ssoCredential (PRIMARY | SECONDARY)] [-ssoDomain <string>] [-httpOnlyCookie (YES | NO)] [-persistentCookie (ENABLED | DISABLED)] [-persistentCookieValidity <minutes>]

Example

```
> set tm sessionParameter -sessTimeout 30
  Done
> set tm sessionParameter -defaultAuthorizationAction DENY
  Done
> set tm sessionParameter -SSO ON
  Done
> set tm sessionParameter -ssoCredential PRIMARY
  Done
```

To configure the session settings by using the configuration utility

1. Navigate to Security > AAA - Application Traffic
2. In the details pane, under Settings, click Change global settings.
3. In the Global Session Settings dialog, type or select values for the parameters.
   - Session Time-out—sessTimeout
   - Default Authorization Action—defaultAuthorizationAction
   - Single Sign-on to Web Applications—sso
   - Credential Index—ssoCredential
   - Single Sign-on Domain—ssoDomain
   - HTTPOnly Cookie—httpOnlyCookie
   - Enable Persistent Cookie—persistentCookie
   - Persistent Cookie Validity (minutes)—persistentCookieValidity
   - Home Page—homepage
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Traffic Settings

Sep 24, 2013

If you use forms-based or SAML single sign-on (SSO) for your protected applications, you configure that feature in the Traffic settings. SSO enables your users to log on once to access all protected applications, rather than requiring them to log on separately to access each one.

Forms-based SSO allows you to use a web form of your own design as the sign-on method instead of a generic pop-up window. You can therefore put you company logo and other information you might want your users to see on the logon form. SAML SSO allows you to configure one NetScaler appliance or virtual appliance instance to authenticate to another NetScaler appliance on behalf of users who have authenticated with the first appliance.

To configure either type of SSO, you first create a forms or SAML SSO profile. Next, you create a traffic profile and link it to the SSO profile you created. Next, you create a policy, link it to the traffic profile. Finally, you bind the policy globally or to an authentication virtual server to put your configuration into effect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Traffic Profiles

Aug 19, 2014

After creating at least one forms or SAML sso profile, you must next create a traffic profile.

Note: In this feature, the terms “profile” and “action” mean the same thing.

To create a traffic profile by using the command line interface

At the command prompt, type:

add tm trafficAction <name> [-appTimeout <mins>] [-SSO (ON | OFF) [-formSSOAction <string>]] [-persistentCookie (ENABLED | DISABLED)] [-InitiateLogout (ON | OFF)]

Example

add tm trafficAction Traffic-Prof-1 –appTimeout 10 -SSO ON -formSSOAction SSO-Prof-1

To modify a session profile by using the command line interface

At the command prompt, type:

set tm trafficAction <name> [-appTimeout <mins>] [-SSO (ON | OFF) [-formSSOAction <string>]] [-persistentCookie (ENABLED | DISABLED)] [-InitiateLogout (ON | OFF)]

Example

set tm trafficAction Traffic-Prof-1 –appTimeout 10 -SSO ON -formSSOAction SSO-Prof-1

To remove a session profile by using the command line interface

At the command prompt, type:

rm tm trafficAction <name>

Example

rm tm trafficAction Traffic-Prof-1

To configure traffic profiles by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Traffic.
2. Navigate to Security > AAA - Application Traffic > Policies > Traffic.
3. In the details pane, click the Profiles tab.
4. On the Profiles tab, do one of the following:
   - To create a new traffic profile, click Add.
   - To modify an existing traffic profile, select the profile, and then click Edit.
5. In the Create Traffic Profile or Configure Traffic Profile dialog box, specify values for the parameters.
   - Name*—name (Cannot be changed for a previously configured session action.)
   - AppTimeout—appTimeout
   - Single Sign-On—SSO
   - Form SSO Action—formSSOAction
   - SAML SSO Action—samlSSOAction
   - Enable Persistent Cookie—persistentCookie
   - Initiate Logout—InitiateLogout
6. Click Create or OK. The traffic profile that you created appears in the Traffic Policies, Profiles, and either the Form SSO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Profiles or SAML SSO Profiles pane, as appropriate.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Traffic Policies

Aug 19, 2014

After you create one or more form SSO and traffic profiles, you create traffic policies and then bind the policies, either globally or to a traffic management virtual server, to put them into effect.

To create a traffic policy by using the command line interface

At the command prompt, type:

add tm trafficPolicy <name> <rule> <action>

Example

add tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS("login=true")" Traffic-Prof-1

To modify a traffic policy by using the command line interface

At the command prompt, type:

set tm trafficPolicy <name> <rule> <action>

Example

set tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS("login=true")" Traffic-Prof-1

To globally bind a traffic policy by using the command line interface

At the command prompt, type:

bind tm global -policyName <string> [-priority <priority>]

Example

bind tm global -policyName Traffic-Pol-1

To bind a traffic policy to a load balancing or content switching virtual server by using the command line interface

At the command prompt, type one of the following commands:

- bind lb vserver <name> -policy <policyName> [-priority <priority>]
- bind cs vserver <name> -policy <policyName> [-priority <priority>]

Example

bind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1 -priority 1000

To unbind a globally bound traffic policy by using the command line interface

At the command prompt, type:

unbind tm global -policyName <policyname>

Example

unbind tm global -policyName Traffic-Pol-1

To unbind a traffic policy from a load balancing or content switching virtual server by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type one of the following commands:

- unbind lb vserver <name> -policy <policyname>
- unbind cs vserver <name> -policy <policyname>

Example

```
unbind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1
```

To remove a traffic policy by using the command line interface

First unbind the session policy from global, and then, at the command prompt, type:

```
rm tm trafficPolicy <name>
```

Example

```
rm tm trafficPolicy Traffic-Pol-1
```

To configure and bind traffic policies by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Traffic.
2. Navigate to Security > AAA - Application Traffic > Policies > Traffic.
3. In the details pane, do one of the following:
   - To create a new session policy, click Add.
   - To modify an existing session policy, select the policy, and then click Edit.
4. In the Create Traffic Policy or Configure Traffic Policy dialog, specify values for the parameters.
   - Name*—policyName (Cannot be changed for a previously configured session policy.)
   - Profile*—actionName
   - Expression—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression text area and then typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
5. Click Create or OK. The policy that you created appears in the details pane of the Session Policies and Profiles page.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Form SSO Profiles

Jun 08, 2015

To enable and configure forms-based SSO, you first create an SSO profile.

Note:

- Forms-based single sign-on does not work if the form is customized to include Javascript.
- In this feature, the terms “profile” and “action” mean the same thing.

To create a form SSO profile by using the command line interface

At the command prompt, type:

- add tm formSSOAction <name> -actionURL <URL> -userField <string> -passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <string>] [-responsesize <positive_integer>] [-nvtype (STATIC | DYNAMIC)] [-submitMethod (GET | POST)
- show tm formSSOAction [<name>]

Example

```
add tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-nameValuePair "loginID passwd" -responsesize "9096"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nvtype STATIC -submitMethod GET
–sessTimeout 10 -defaultAuthorizationAction ALLOW
```

To modify a form SSO by using the command line interface

At the command prompt, type:

set tm formSSOAction <name> -actionURL <URL> -userField <string> -passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <string>] [-responsesize <positive_integer>] [-nvtype (STATIC | DYNAMIC)] [-submitMethod (GET | POST)

Example

```
set tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nameValuePair "loginID passwd" -responsesize "9096"
-nvtype STATIC -submitMethod GET
–sessTimeout 10 -defaultAuthorizationAction ALLOW
```

To remove a form SSO profile by using the command line interface

At the command prompt, type:

rm tm formSSOAction <name>

Example

```
rm tm sessionAction SSO-Prof-1
```

To configure form SSO profiles by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Security > AAA - Application Traffic > Policies > Traffic.
2. In the details pane, click the Form SSO Profiles tab.
3. On the Form SSO Profiles tab, do one of the following:
   - To create a new form SSO profile, click Add.
   - To modify an existing form SSO profile, select the profile, and then click Edit.
4. In the Create Form SSO Profile or Configure Form SSO Profile dialog, specify values for the parameters:
   - Name*—name (Cannot be changed for a previously configured session action.)
   - Action URL*—actionURL
   - User Name Field*—userField
   - Password Field*—passField
   - Expression*—ssoSuccessRule
   - Name Value Pair—nameValuePair
   - Response Size—responsesize
   - Extraction—nvtype
   - Submit Method—submitMethod
5. Click Create or OK, and then click Close. The form SSO profile that you created appears in the Traffic Policies, Profiles, and Form SSO Profiles pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SAML SSO Profiles

Jun 12, 2014

To enable and configure SAML-based SSO, you first create a SAML SSO profile.

To create a SAML SSO profile by using the command line interface

At the command prompt, type:

add tm samlSSOProfile <name> -samlSigningCertName <string> -assertionConsumerServiceURL <URL> -relaystateRule <expression> -sendPassword (ON | OFF) [-samlIssuerName <string>]

Example

add tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example, Inc." -assertionConsumerServiceURL "https://service.example.com" -relaystateRule "true" -sendPassword "ON" -samlIssuer

To modify a SAML SSO by using the command line interface

At the command prompt, type:

set tm samlSSOProfile <name> -samlSigningCertName <string> -assertionConsumerServiceURL <URL> -relaystateRule <expression> -sendPassword (ON | OFF) [-samlIssuerName <string>]

Example

set tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example, Inc." -assertionConsumerServiceURL "https://service.example.com" -relaystateRule "true" -sendPassword "ON" -samlIssuer

To remove a SAML SSO profile by using the command line interface

At the command prompt, type:

rm tm samlSSOProfile <name>

Example

rm tm sessionAction saml-SSO-Prof-1

To configure a SAML SSO profile by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Policies > Traffic.
2. In the details pane, click the SAML SSO Profiles tab.
3. On the SAML SSO Profiles tab, do one of the following:
   - To create a new SAML SSO profile, click Add.
   - To modify an existing SAML SSO profile, select the profile, and then click OpenEdit.
4. In the Create SAML SSO Profiles or the Configure SAML SSO Profiles dialog box, set the following parameters:
   - Name*
   - Signing Certificate Name*
   - ACS URL*
   - Relay State Rule*
   - Send Password
   - Issuer Name
5. Click Create or OK, and then click Close. The SAML SSO profile that you created appears in the Traffic Policies, Profiles, and SAML SSO Profiles pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Session Timeout for OWA 2010

Jun 06, 2014

You can now force OWA 2010 connections to time out after a specified period of inactivity. OWA sends repeated keepalive requests to the server to prevent timeouts. Keeping the connections open can interfere with single sign-on.

To force OWA 2010 to timeout after a specified period by using the command line interface

At the command prompt, type the following commands:

- add tm trafficAction <actname> [-forcedTimeout <forcedTimeout> -forcedTimeoutVal <mins>]
  For <actname>, substitute a name for your traffic policy. For <mins>, substitute the number of minutes after which to initiate a forced timeout. For <forcedTimeout>, substitute one of the following values:
  - START—Starts the timer for forced timeout if a timer has not already been started. If a running timer exists, has no effect.
  - STOP—Stops a running timer. If no running timer is found, has no effect.
  - RESET—Restarts a running timer. If no running timer is found, starts a timer just as if the START option had been used.
- add tm trafficPolicy <polname> <rule> <actname>
  For <polname>, substitute a name for your traffic policy. For <rule>, substitute a rule in NetScaler default syntax.
- bind lb vserver <vservername> –policyName <name> -priority <number>
  For <vservername>, substitute the name of the AAA traffic management virtual server. For <priority>, substitute an integer that designates the policy's priority.

Example

```
add tm trafficAction act-owa2010timeout -forcedTimeout RESET -forcedTimeoutVal 10
add tm trafficPolicy pol-owa2010timeout true act-owa2010timeout
bind lb vserver vs-owa2010 -policyName pol-owa2010timeout -priority 10
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Authenticating with Client Certificates

Aug 13, 2014

Web sites that contain sensitive content, such as online banking websites or websites with employee personal information, sometimes require client certificates for authentication. To configure AAA to authenticate users on the basis of client-side certificate attributes, you first enable client authentication on the traffic management virtual server and bind the root certificate to the authentication virtual server. Then, you implement one of two options. You can configure the default authentication type on the authentication virtual server as CERT, or you can create a certificate action that defines what the NetScaler ADC must do to authenticate users on the basis of a client certificate. In either case, your authentication server must support CRLs. You configure the ADC to extract the user name from the SubjectCN field or another specified field in the client certificate.

When the user tries to log on to an authentication virtual server for which an authentication policy is not configured, and a global cascade is not configured, the user name information is extracted from the specified field of the certificate. If the required field is extracted, the authentication succeeds. If the user does not provide a valid certificate during the SSL handshake, or if the user name extraction fails, authentication fails. After it validates the client certificate, the ADC presents a logon page to the user.

The following procedures assume that you have already created a functioning AAA configuration, and therefore they explain only how to enable authentication by using client certificates. These procedures also assume that you have obtained your root certificate and client certificates and have placed them on the ADC in the /nsconfig/ssl directory.

To configure the AAA client certificate parameters by using the command line interface

At the command prompt, type the following commands, in the order shown, to configure the certificate and verify the configuration:

- add ssl certKey <certkeyName> -cert <certFile> -key <keyFile> -password -inform <inform> -expiryMonitor <expiryMonitor> -notificationPeriod <notificationPeriod>
- bind ssl certKey <certkeyName> -vServer <certkeyName> -CA -crlCheck Mandatory
- show ssl certKey [<certkeyName>]
- set aaa parameter -defaultAuthType CERT
- show aaa parameter
- set aaa certParams -userNameField "Subject:CN"
- show aaa certParams

To configure the AAA client certificate parameters by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Virtual Servers.
2. In the details pane, select the virtual server that you want to configure to handle client certificate authentication, and then click Edit.
3. On the Configuration page, under Certificates, click the right arrow (>) to open the CA Cert Key installation dialog.
4. In theCA Cert Key dialog box, click Insert.
5. In the CA Cert Key - SSL Certificates dialog box, click Install.
6. In the Install Certificate dialog box, set the following parameters, whose names correspond to the CLI parameter names as shown:
   - Certificate-Key Pair Name*—certkeyName
   - Certificate File Name—certFile
   - Key File Name—keyFile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Certificate Format—inform
- Password—password
- Certificate Bundle—bundle
- Notify When Expires—expiryMonitor
- Notification Period—notificationPeriod

7. Click Install, and then click Close.
8. In the CA Cert Key dialog box, in the Certificate list, select the root certificate.
9. Click Save.
10. Click Back to return to the main configuration screen.
11. Navigate to Security > AAA - Application Traffic > Policies > Authentication > CERT.
12. In the details pane, select the policy you want to configure to handle client certificate authentication, and then click Edit.
13. In the Configure Authentication CERT Policy dialog, Server drop-down list, select the virtual server you just configured to handle client certificate authentication.
14. Click OK. A message appears in the status bar, stating that the configuration completed successfully.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Client Certificate Pass-Through

Jul 30, 2014

The NetScaler ADC can now be configured to pass client certificates through to protected applications that require client certificates for user authentication. The ADC first authenticates the user, then inserts the client certificate into the request and sends it to the application. This feature is configured by adding appropriate SSL policies.

The exact behavior of this feature when a user presents a client certificate depends upon the configuration of the VPN virtual server.

- If the VPN virtual server is configured to accept client certificates but not require them, the ADC inserts the certificate into the request and then forwards the request to the protected application.
- If the VPN virtual server has client certificate authentication disabled, the ADC renegotiatiates the authentication protocol and reauthenticates the user before it inserts the client certificate in the header and forwards the request to the protected application.
- If the VPN virtual server is configured to require client certificate authentication, the ADC uses the client certificate to authenticate the user, then inserts the certificate in the header and forwards the request to the protected application.

In all of these cases, you configure client certificate pass-through as follows.

To create and configure client certificate pass-through by using the command line interface

At the command prompt, type the following commands:

- add vpn vserver <name> SSL <IP> 443
  For <name>, substitute a name for the virtual server. The name must contain from one to 127 ASCII characters, beginning with a letter or underscore (_), and containing only letters, numbers, and the underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. For <IP>, substitute the IP address assigned to the virtual server.
- set ssl vserver <name> -clientAuth ENABLED -clientCert <clientcert>
  For <name>, substitute the name of the virtual server that you just created. For <clientCert>, substitute one of the following values:
  - disabled—disables client certificate authentication on the VPN virtual server.
  - mandatory—configures the VPN virtual server to require client certificates to authenticate.
  - optional—configures the VPN virtual server to allow client certificate authentication, but not to require it.
- bind vpn vserver <name> -policy local
  For <name>, substitute the name of the VPN virtual server that you created.
- bind vpn vserver <name> -policy cert
  For <name>, substitute the name of the VPN virtual server that you created.
- bind ssl vserver <name> -certkeyName <certkeyname>
  For <name>, substitute the name of the virtual server that you created. For <certkeyName>, substitute the client certificate key.
- bind ssl vserver <name> -certkeyName <cacertkeyname> -CA -ocspCheck Optional
  For <name>, substitute the name of the virtual server that you created. For <cacertkeyName>, substitute the CA certificate key.
- add ssl action <actname> -clientCert ENABLED -certHeader CLIENT-CERT
  For <actname>, substitute a name for the SSL action.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- add ssl policy <polname> -rule true -action <actname>
  For <polname>, substitute a name for your new SSL policy. For <actname>, substitute the name of the SSL action that you just created.
- bind ssl vserver <name> -policyName <polname> -priority 10
  For <name>, substitute the name of the VPN virtual server.

Example

```
add vpn vserver vs-certpassthru SSL 10.121.250.75 443
set ssl vserver vs-certpassthru -clientAuth ENABLED -clientCert optional
bind vpn vserver vs-certpassthru -policy local
bind vpn vserver vs-certpassthru -policy cert
bind ssl vserver vs-certpassthru -certkeyName mycertKey
bind ssl vserver vs-certpassthru -certkeyName mycertKey -CA -ocspCheck Optional
add ssl action act-certpassthru -clientCert ENABLED -certHeader CLIENT-CERT
add ssl policy pol-certpassthru -rule true -action act-certpassthru
bind ssl vserver vs-certpassthru -policyName pol-certpassthru -priority 10
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring AAA with Commonly Used Protocols

Mar 15, 2013

Configuring the NetScaler for Authentication, Authorization, and Auditing (AAA) needs a specific setup on the NetScaler and clients' browsers. The configuration varies with the protocol used for AAA.

For more information about configuring the NetScaler for Kerberos authentication, see Handling Authentication, Authorization and Auditing with Kerberos/NTLM.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Handling Authentication, Authorization and Auditing with Kerberos/NTLM

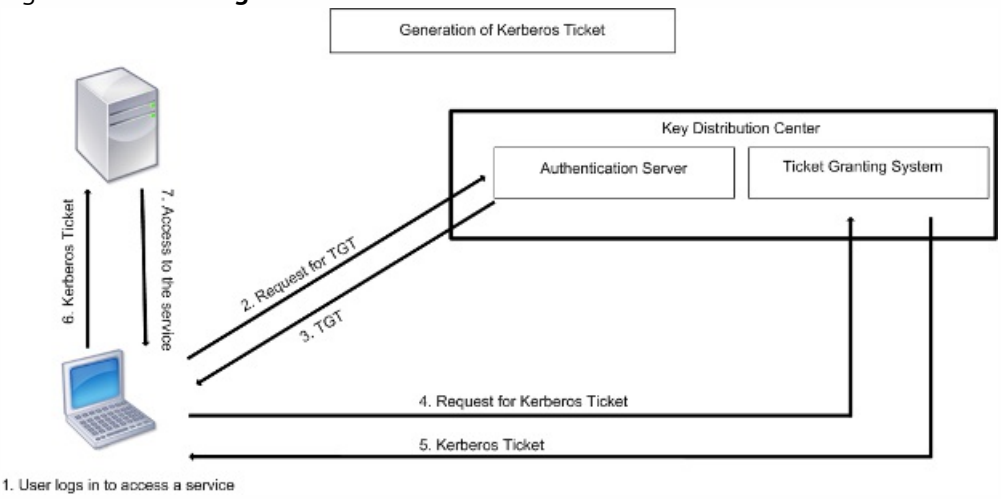
Mar 27, 2012

Kerberos, a computer network authentication protocol, provides secure communication over the Internet. Designed primarily for client-server applications, it provides for mutual authentication by which the client and server can each ensure the other's authenticity. Kerberos uses a trusted third party, referred to as Key Distribution Center (KDC). A KDC consists of an Authentication Server (AS), which authenticates a user, and a Ticket Granting Server (TGS).

Each entity on the network (client or server) has a secret key that is known only to itself and the KDC. The knowledge of this key implies authenticity of the entity. For communication between two entities on the network, the KDC generates a session key, referred to as the Kerberos ticket or service ticket. The client makes a request to the AS for credentials for a specific server. The client then receives a ticket, referred to as Ticket Granting Ticket (TGT). The client then contacts the TGS, using the TGT it received from the AS to prove its identity, and asks for a service. If the client is eligible for the service, the TGS issues a Kerberos ticket to the client. The client then contacts the server hosting the service (referred to as the service server), using the Kerberos ticket to prove that it is authorized to receive the service. The Kerberos ticket has a configurable lifetime. The client authenticates itself with the AS only once. If it contacts the physical server multiple times, it reuses the AS ticket.

The following figure shows the basic functioning of the Kerberos protocol.

Figure 1. **Functioning of Kerberos**



Kerberos authentication has the following advantages:

- Faster authentication. When a physical server gets a Kerberos ticket from a client, the server has enough information to authenticate the client directly. It does not have to contact a domain controller for client authentication, and therefore the authentication process is faster.
- Mutual authentication. When the KDC issues a Kerberos ticket to a client and the client uses the ticket to access a service, only authenticated servers can decrypt the Kerberos ticket. If the virtual server on the NetScaler is able to decrypt the Kerberos ticket, you can conclude that both the virtual server and client are authenticated. Thus, the authentication of the server happens along with the authentication of the client.
- Single sign-on between Windows and other operating systems that support Kerberos.

Kerberos authentication may have the following disadvantages:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Kerberos has strict time requirements; the clocks of the involved hosts must be synchronized with the Kerberos server clock to ensure that the authentication does not fail. You can mitigate this disadvantage by using the Network Time Protocol daemons to keep the host clocks synchronized. Kerberos tickets have an availability period, which you can configure.
- Kerberos needs the central server to be available continuously. When the Kerberos server is down, no one can log on. You can mitigate this risk by using multiple Kerberos servers and fallback authentication mechanisms.
- Because all the authentication is controlled by a centralized KDC, any compromise in this infrastructure, such as the user's password for a local workstation being stolen, can allow an attacker to impersonate any user. You can mitigate this risk to some extent by using only a desktop machine or laptop that you trust, or by enforcing preauthentication by means of a hardware-token.

To use Kerberos authentication, you must configure it on the NetScaler appliance and on each client.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How NetScaler Implements Kerberos for Client Authentication

Jun 12, 2014

Note: Kerberos/NTLM authentication is supported only in the NetScaler 9.3 nCore release or later, and it can be used only for AAA traffic management (AAA-TM) virtual servers.

NetScaler handles the components involved in Kerberos authentication in the following way:

Key Distribution Center (KDC)

In the Windows 2000 Server or later versions, the Domain Controller and KDC are part of the Windows Server. If the Windows Server is UP and running, it indicates that the Domain Controller and KDC are configured. The KDC is also the Active Directory server.

Note: All Kerberos interactions are validated with the Windows Kerberos Domain Controller.

Authentication Service and Protocol Negotiation

A NetScaler appliance supports Kerberos authentication on the AAA-TM authentication virtual servers. If the Kerberos authentication fails, the NetScaler uses the NTLM authentication.

By default, Windows 2000 Server and later Windows Server versions use Kerberos for AAA. If you create an authentication policy with NEGOTIATE as the authentication type, the NetScaler attempts to use the Kerberos protocol for AAA and if the client's browser fails to receive a Kerberos ticket, the NetScaler uses the NTLM authentication. This process is referred to as negotiation.

The client may fail to receive a Kerberos ticket in any of the following cases:

- Kerberos is not supported on the client.
- Kerberos is not enabled on the client.
- The client is in a domain other than that of the KDC.
- The Access Directory on the KDC is not accessible to the client.

For Kerberos/NTLM authentication, the NetScaler does not use the data that is present locally on the NetScaler appliance.

Authorization

The traffic management virtual server can be a load balancing virtual server or a content switching virtual server.

Auditing

The NetScaler appliance supports auditing of Kerberos authentication with the following audit logging:

- Complete audit trail of the traffic management end-user activity
- SYSLOG and high performance TCP logging
- Complete audit trail of system administrators
- All system events
- Scriptable log format

Supported Environment

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Kerberos authentication does not need any specific environment on the NetScaler. The client (browser) must provide support for Kerberos authentication.

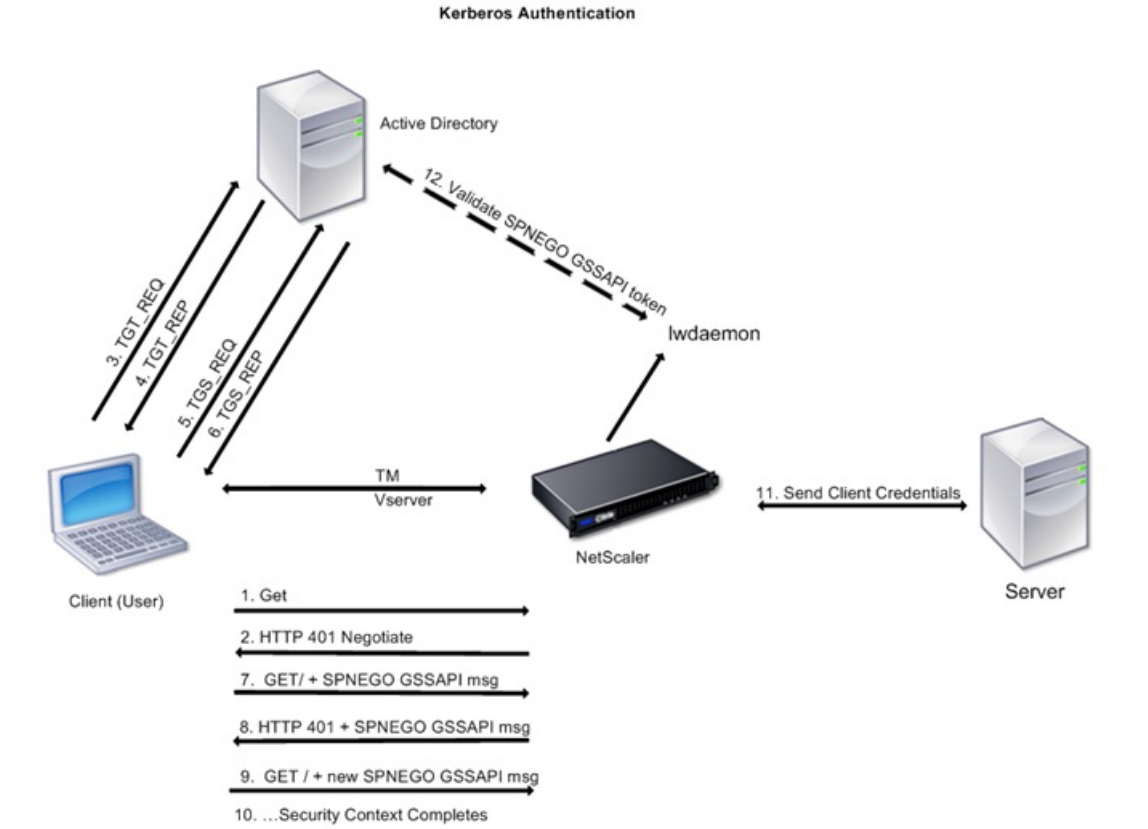
High Availability

In a high availability setup, only the active NetScaler joins the domain. In case of a failover, the NetScaler lwagent daemon joins the secondary NetScaler appliance to the domain. No specific configuration is required for this functionality.

Kerberos Authentication Process

The following figure shows a typical process for Kerberos authentication in the NetScaler environment.

Figure 1. Kerberos Authentication Process on NetScaler



The Kerberos authentication occurs in the following stages:

Client authenticates itself to the KDC.

1. The NetScaler appliance receives a request from a client.
2. The traffic management (load balancing or content switching) virtual server on the NetScaler sends a challenge to the client.
3. To respond to the challenge, the client gets a Kerberos ticket.
   - The client sends the Authentication Server of the KDC a request for a ticket-granting ticket (TGT) and receives the TGT. (See 3, 4 in the figure, Kerberos Authentication Process.)
   - The client sends the TGT to the Ticket Granting Server of the KDC and receives a Kerberos ticket. (See 5, 6 in the figure, Kerberos Authentication Process.)

Note: The above authentication process is not necessary if the client already has a Kerberos ticket whose lifetime has not expired. In addition, clients such as Web Services, .NET, or J2EE, which support SPNEGO, get a Kerberos ticket for the target server, create an SPNEGO token, and insert the token in the HTTP header when they send an HTTP request. They do not

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

go through the client authentication process.

Client requests a service.

1. The client sends the Kerberos ticket containing the SPNEGO token and the HTTP request to the traffic management virtual server on the NetScaler. The SPNEGO token has the necessary GSSAPI data.
2. The NetScaler establishes a security context between the client and the NetScaler. If the NetScaler cannot accept the data provided in the Kerberos ticket, the client is asked to get a different ticket. This cycle repeats till the GSSAPI data is acceptable and the security context is established. The traffic management virtual server on the NetScaler acts as an HTTP proxy between the client and the physical server.

NetScaler completes the authentication.

1. After the security context is complete, the traffic management virtual server validates the SPNEGO token.
2. From the valid SPNEGO token, the virtual server extracts the user ID and GSS credentials, and passes them to the authentication daemon.
3. A successful authentication completes the Kerberos authentication.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Kerberos Authentication on the NetScaler Appliance

Oct 26, 2015

This topic provides the detailed steps to configure Kerberos authentication on the NetScaler by using the CLI and the GUI.

Configuring Kerberos authentication on the NetScaler CLI

1. Enable the AAA feature to ensure the authentication of traffic on the appliance.

   ns-cli-prompt> **enable ns feature** AAA

2. Add the keytab file to the NetScaler appliance. A keytab file is necessary for decrypting the secret received from the client during Kerberos authentication. A single keytab file contains authentication details for all the services that are bound to the traffic management virtual server on the NetScaler.

   First generate the keytab file on the Active Directory server and then transfer it to the NetScaler appliance.

   1. Log on to the Active Directory server and add a user for Kerberos authentication. For example, to add a user named "Kerb-SVC-Account":

      net user Kerb-SVC-Account freebsd!@#456 /add

      Note: In the **User Properties** section, ensure that the "Change password at next logon option" is not selected and the "Password does not expire" option is selected.

   2. Map the HTTP service to the above user and export the keytab file. For example, run the following command on the Active Directory server:

      ktpass /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM /pass freebsd!@#456 /mapuser newacp\dummy /ptype KRB5_NT_PRINCIPAL

      Note: You can map more than one service if authentication is required for more than one service. If you want to map more services, repeat the above command for every service. You can give the same name or different names for the output file.

   3. Transfer the keytab file to the NetScaler by using the unix **ftp** command or any other file transfer utility of your choice.

   4. Log on to the NetScaler appliance, and run the **ktutil** utility to verify the keytab file. The keytab file has an entry for the HTTP service after it is imported.

      The **kutil** interactions are as follows:

      root@ns# ktutil

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
ktutil: rkt /var/keytabfile
ktutil: list

slot KVNO Principal
----------------------------------------------------------------------


ktutil: wkt /etc/ krb5.keytab
ktutil: list

slot KVNO Principal
---- ---- ----------------------------------------------------------------
1 2 HTTP/owa.newacp.com@NEWACP.COM

ktutil: quit
```

3. The NetScaler must obtain the IP address of the domain controller from the fully qualified domain name (FQDN). Therefore, Citrix recommends configuring the NetScaler with a DNS server.

   ns-cli-prompt> **add dns nameserver** <ip-address>

   Note: Alternatively, you can add static host entries or use any other means so that the NetScaler can resolve the FQDN name of the domain controller to an IP address.

4. Configure the authentication action and then associate it to an authentication policy.

   1. Configure the negotiate action.

      ns-cli-prompt> **add authentication negotiateAction** <name> -domain <domainName> -domainUser <domainUsername> -domainUserPasswd <domainUserPassword>

   2. Configure the negotiate policy and associate the negotiate action to this policy.

      ns-cli-prompt> **add authentication negotiatePolicy** <name> <rule> <reqAction>

5. Create an authentication virtual server and associate the negotiate policy with it.

   1. Create an authentication virtual server.

      ns-cli-prompt> **add authentication vserver** <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>

   2. Bind the negotiate policy to the authentication virtual server.

      ns-cli-prompt> **bind authentication vserver** <name> -policy <negotiatePolicyName>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

6. Associate the authentication virtual server with the traffic management (load balancing or content switching) virtual server.

   ns-cli-prompt> **set lb vserver** <name> -authn401 ON -authnVsName <string>

   Note: Similar configurations can also be done on the content switching virtual server.

7. Verify the configurations by doing the following:

   1. Access the traffic management virtual server, using the FQDN. For example, http://owa.newacp.com.

   2. View the details of the session on the NetScaler CLI.

      ns-cli-prompt> **show aaa session**

Configuring Kerberos authentication on the NetScaler GUI

1. Enable the AAA feature.

   Navigate to **System** > **Settings**, click **Configure Basic Features** and enable the AAA feature.

2. Add the keytab file as detailed in step 2 of the CLI procedure mentioned above.

3. Add a DNS server.

   Navigate to **Traffic Management** > **DNS** > **Name Servers**, and specify the IP address for the DNS server.

4. Configure the **Negotiate** action and policy.

   Navigate to **Security** > **AAA - Application Traffic** > **Policies** > **Authentication** > **Advanced Policies** > **Policy**, and create a policy with **Negotiate** as the action type.

5. Bind the negotiate policy to the authentication virtual server.

   Navigate to **Security** > **AAA - Application Traffic** > **Virtual Servers**, and associate the **Negotiate** policy with the authentication virtual server.

6. Associate the authentication virtual server with the traffic management (load balancing or content switching) virtual server.

   Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and specify the relevant authentication settings.

   Note: Similar configurations can also be done on the content switching virtual server.

7. Verify the configurations as detailed in step 7 of the CLI procedure mentioned above.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Kerberos Authentication on a Client

Oct 27, 2015

Kerberos support must be configured on the browser to use Kerberos for authentication. You can use any Kerberos-compliant browser. Instructions for configuring Kerberos support on Internet Explorer and Mozilla Firefox follow. For other browsers, see the documentation of the browser.

To configure Internet Explorer for Kerberos authentication

1. In the Tools menu select Internet Options.
2. On the Security tab, click Local Intranet, and then click Sites.
3. In the Local Intranet dialog box, make sure that the Automatically detect intranet network option is selected, and then click Advanced.
4. In the Local Intranet dialog box, add the web sites of the domains of the traffic management virtual server on the NetScaler. The specified sites become local intranet sites.
5. Click Close or OK to close the dialog boxes.

To configure Mozilla Firefox for Kerberos authentication

1. Make sure that you have Kerberos properly configured on your computer.
2. Type about:config in the URL bar.
3. In the filter text box, type network.negotiate.
4. Change network.negotiate-auth.delegation-uris to the domain that you want to add.
5. Change network.negotiate-auth.trusted-uris to the domain that you want to add.
   Note: If you are running Windows, you also need to enter sspi in the filter text box and change the network.auth.use-sspi option to False.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Offloading Kerberos Authentication from Physical Servers

Aug 24, 2016

The NetScaler appliance can offload authentication tasks from servers. Instead of the physical servers authenticating the requests from clients, the Netscaler authenticates all the client requests before it forwards them to any of the physical servers bound to it. The user authentication is based on Active Directory tokens.

There is no authentication between the NetScaler and the physical server, and the authentication offload is transparent to the end users. After the initial logon to a Windows computer, the end user does not have to enter any additional authentication information in a pop-up or on a logon page.

In the current NetScaler release, Kerberos authentication is available only for Authentication, Authorization, and Auditing (AAA) Traffic Management Virtual Servers. Kerberos authentication is not supported for SSL VPN in the NetScaler Gateway Enterprise Edition appliance or for NetScaler appliance management.

Kerberos authentication requires configuration on the NetScaler appliance and on client browsers.

To configure Kerberos authentication on the NetScaler appliance

1. Create a user account on Active Directory. When creating a user account, verify the following options in the User Properties section:
   - Make sure that you do not select the Change password at next logon option.
   - Be sure to select the Password does not expire option.
2. On the AD server, at the CLI command prompt, type:
   - ktpass -princ HTTP/kerberos.crete.lab.net@crete.lab.net -ptype KRB5_NT_PRINCIPAL -mapuser kerbuser@crete.lab.net -mapop set -pass Citrix1 -out C:\kerbtabfile.txt

   Note: Be sure to type the above command on a single line. The output of the above command is written into the C:\kerbtabfile.txt.file.
3. Upload the kerbtabfile.txt file to the /etc directory of the NetScaler appliance by using a Secure Copy (SCP) client.
4. Run the following command to add a DNS server to the NetScaler appliance.
   - add dns nameserver 1.2.3.4

   The NetScaler appliance cannot process Kerberos requests without the DNS server. Be sure to use the same DNS server that is used in the Microsoft Windows domain.

5. Switch to the shell prompt and run the following commands from the shell prompt:
   - ktutil # rkt /etc/kerbtabfile.txt
   - # wkt /etc/krb5.keytab
   - # list

     The list command displays the user account details that you created in the Active Directory. A sample screen of the output of the list command is shown below.

     Figure 1. Sample Output of the list Command

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
> shell
Copyright (c) 1992-2008 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992,
        The Regents of the University of California. All righ

root@ns# cd /etc
root@ns# ls -la *.txt
-rw-r--r--  1 root  wheel  82 Apr  4 00:43 kerbtabfile.txt
root@ns# ktutil
ktutil:  rkt /etc/kerbtabfile.txt
ktutil:  wkt /etc/krb5.keytab
ktutil:  list
slot KVNO Principal
---- ---- ---------------------------------------------------
   1    3 HTTP/kerberos.crete.example.com@crete.example.com
ktutil:  quit
root@ns#
```

6. Switch to the command line interface of NetScaler.
7. Run the following command to create a Kerberos authentication server:
   - add authentication negotiateAction KerberosServer -domain "crete.lab.net" -domainUser kerbuser -domainUserPasswd Citrix1
8. Run the following command to create a negotiation policy:
   - add authentication negotiatePolicy Kerberos-Policy "REQ.IP.DESTIP == 192.168.17.200" KerberosServer
9. Run the following command to create an authentication virtual server.
   - add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.example.com
10. Run the following command to bind the Kerberos policy to the authentication virtual server:
    - bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100
11. Run the following command to bind an SSL certificate to the authentication virtual server. You can use one of the test certificates, which you can install from the GUI NetScaler appliance. Run the following command to use the ServerTestCert sample certificate.
    - bind ssl vserver Kerb-Auth -certkeyName ServerTestCert
12. Create an HTTP load balancing virtual server with the IP address, 192.168.17.200.
    Ensure that you create a virtual server from the command line interface for NetScaler 9.3 releases if they are older than 9.3.47.8.
13. Run the following command to configure an authentication virtual server:
    - set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth
14. Enter the host name http://www.crete.example.com in the address bar of the Web browser.
    The Web browser displays an authentication dialog box because the Kerberos authentication is not set up in the browser.

    Note: Kerberos authentication requires a specific configuration on the client. Ensure that the client can resolve the hostname, which results in the Web browser connecting to an HTTP virtual server.
15. Configure Kerberos on the Web browser of the client computer.
    - For configuring on Internet Explorer, see Configuring Internet Explorer for Kerberos authentication.
    - For configuring on Mozilla Firefox, see Configuring Internet Explorer for Kerberos authentication.
16. Verify whether you can access the backend physical server without authentication.

To configure Internet Explorer for Kerberos authentication

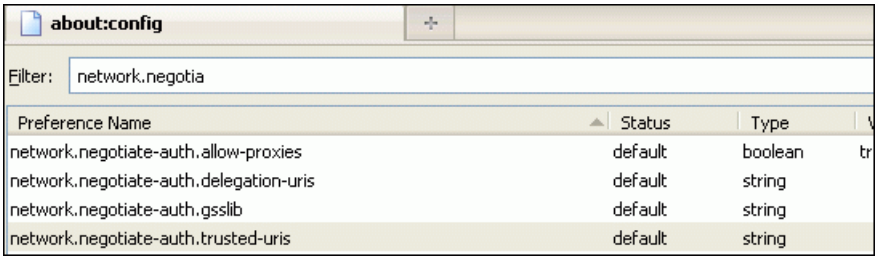
1. Select Internet Options from the Tools menu.
2. Activate the Security tab.
3. Select Local Intranet from the Select a zone to view change security settings section.
4. Click Sites.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Click Advanced.
6. Specify the URL, http://www.crete.example.com and click Add.
7. Restart Internet Explorer.

To configure Mozilla Firefox for Kerberos authentication

1. Enter about:config in the address bar of the browser.
2. Click the warning disclaimer.
3. Type Network.Negotiate-auth.trusted-uris in the Filter box.
4. Double click Network.Negotiate-auth.trusted-uris. A sample screen is shown below.



5. In the Enter String Value dialog box, specify www.crete.example.com.

6. Restart Firefox.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler Kerberos Single Sign-On

Oct 11, 2013

NetScaler appliances now support single sign-on (SSO) using the Kerberos 5 protocol. Users log on to a proxy, the Application Delivery Controller (ADC), which then provides access to protected resources.

The NetScaler Kerberos SSO implementation requires the user's password for SSO methods that rely on basic, NTLM, or forms-based authentication. The user's password is not required for Kerberos SSO, although if Kerberos SSO fails and the NetScaler appliance has the user's password, it uses the password to attempt NTLM SSO.

If the user's password is available, the KCD account is configured with a realm, and no delegated user information is present, the NetScaler Kerberos SSO engine impersonates the user to obtain access to authorized resources. Impersonation is also called unconstrained delegation.

The NetScaler Kerberos SSO engine can also be configured to use a delegated account to obtain access to protected resources on the user's behalf. This configuration requires delegated user credentials, a keytab, or a delegated user certificate and matching CA certificate. Configuration that uses a delegated account is called constrained delegation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

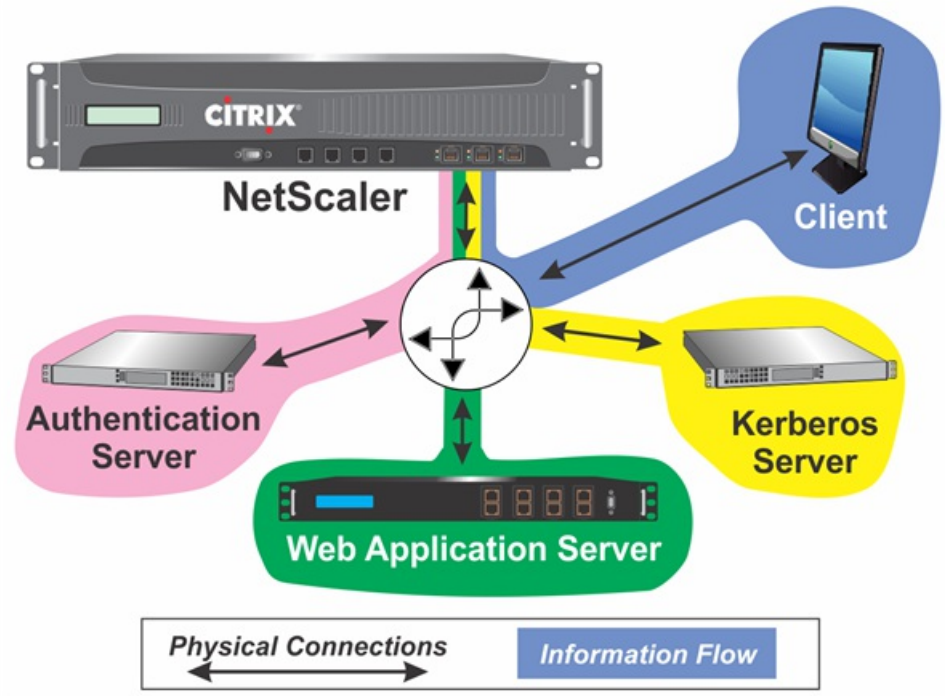
An Overview of NetScaler Kerberos SSO

Oct 14, 2013

To use the NetScaler Kerberos SSO feature does, users first authenticate with Kerberos or a supported third-party authentication server. Once authenticated, the user requests access to a protected web application. The web server responds with a request for proof that the user is authorized to access that web application. The user's browser contacts the Kerberos server, which verifies that the user is authorized to access that resource, and then provides the user's browser with a service ticket that provides proof. The browser resends the user's request to the web application server with the service ticket attached. The web application server verifies the service ticket, and then allows the user to access the application.

AAA-TM implements this process as shown in the following diagram. The diagram illustrates the flow of information through the NetScaler appliance and AAA-TM, on a secure network with LDAP authentication and Kerberos authorization. AAA-TM environments that use other types of authentication have essentially the same information flow, although they might differ in some details.

Figure 1. A Secure Network with LDAP and Kerberos



NetScaler AAA-TM authentication and authorization in a Kerberos environment requires that the following actions take place.

1. The client sends a request for a resource to the traffic management virtual server on the NetScaler appliance.
2. The traffic management virtual server passes the request to the authentication virtual server, which authenticates the client and then passes the request back to the traffic management virtual server.
3. The traffic management virtual server sends the client's request to the web application server.
4. The web application server responds to the traffic management virtual server with a 401 Unauthorized message that requests Kerberos authentication, with fallback to NTLM authentication if the client does not support Kerberos.
5. The traffic management virtual server contacts the Kerberos SSO daemon.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

6. The Kerberos SSO daemon contacts the Kerberos server and obtains a ticket-granting ticket (TGT) allowing it to request service tickets authorizing access to protected applications.
7. The Kerberos SSO daemon obtains a service ticket for the user and sends that ticket to the traffic management virtual server.
8. The traffic management virtual server attaches the ticket to the user's initial request and sends the modified request back to the web application server.
9. The web application server responds with a 200 OK message.

These steps are transparent to the client, which just sends a request and receives the requested resource.

Integration of NetScaler Kerberos SSO with Authentication Methods

All AAA-TM authentication mechanisms support NetScaler Kerberos SSO. AAA-TM supports the Kerberos SSO mechanism with the Kerberos, CAC (Smart Card) and SAML authentication mechanisms with any form of client authentication to the NetScaler appliance. It also supports the HTTP-Basic, HTTP-Digest, Forms-based, and NTLM (versions 1 and 2) SSO mechanisms if the client uses either HTTP-Basic or Forms-Based authentication to log on to the NetScaler appliance.

The following table shows each supported client-side authentication method, and the supported server-side authentication method for that client-side method.

Table 1. Supported Authentication Methods

	Basic/Digest/NTLM	Kerberos Constrained Delegation	User Impersonation
CAC (Smart Card): at SSL/TLS Layer		X	X
Forms-Based (LDAP/RADIUS/TACACS)	X	X	X
HTTP Basic (LDAP/RADIUS/TACACS)	X	X	X
Kerberos		X	
NTLM v1/v2		X	X
SAML		X	
SAML Two-Factor	X	X	X
Certificate Two-Factor	X	X	X

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting up NetScaler SSO

Oct 08, 2013

You can configure NetScaler SSO to work in one of two ways: by impersonation or by delegation. SSO by impersonation is a simpler configuration than SSO by delegation, and is therefore usually preferable when your configuration allows it. To configure NetScaler SSO by impersonation, you must have the user's user name and password.

To configure NetScaler SSO by delegation, you must have the delegated user's credentials in one of the following formats: the user's user name and password, the keytab configuration that includes the user name and an encrypted password, or the delegated user certificate and the matching CA certificate.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Prerequisites

Mar 30, 2015

Before you configure NetScaler SSO, you need to have your NetScaler appliance fully configured to manage traffic to and authentication for your web application servers. Therefore, you must configure either load balancing or content switching, and then AAA, for these web application servers. You should also verify routing between the appliance, your LDAP server, and your Kerberos server.

If your network is not already configured in this manner, perform the following configuration tasks:

- Configure a server and service for each web application server.
- Configure a traffic management virtual server to handle traffic to and from your web application server.

Following are brief instructions and examples for performing each of these tasks from the NetScaler command line. For further assistance, see Setting up AAA Virtual Servers and DNS.

To create a server and service by using the NetScaler command line

For NetScaler SSO to obtain a TGS (service ticket) for a service, either the FQDN assigned to the server entity on the NetScaler appliance must match the FQDN of the web application server, or the server entity name must match the NetBios name of the web application server. You can take either of the following approaches:

- Configure the NetScaler server entity by specifying the FQDN of the web application server.
- Configure the NetScaler server entity by specifying the IP address of the web application server, and assign the server entity the same name as the NetBios name of the web application server.

At the command prompt, type the following commands:

- add server name <serverFQDN>
- add service name serverName serviceType port

For the variables, substitute the following values:

- **serverName**—A name for the NetScaler appliance to use to refer to this server.
- **serverFQDN**—The FQDN of the server. If the server has no domain assigned to it, use the server's IP address and make sure that the server entity name matches the NetBios name of the web application server.
- **serviceName**—A name for the NetScaler appliance to use to refer to this service.
- **type**—The protocol used by the service, either HTTP or MSSQLSVC.
- **port**—The port on which the service listens. HTTP services normally listen on port 80. Secure HTTPS services normally listen on port 443.

Example

The following examples add server and service entries on the NetScaler appliance for the web application server was1.example.com. The first example uses the FQDN of the web application server; the second uses the IP address.

To add the server and service using the web application server FQDN, was1.example.com, you would type the following commands:

```
add server was1 was1.example.com
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add service was1service was1 HTTP 80
```

To add the server and service using the web application server IP and NetBios name, where the web application server IP is 10.237.64.87 and its NetBios name is WAS1, you would type the following commands:

```
add server WAS1 10.237.64.87
add service was1service WAS1 HTTP 8
```

To create a traffic management virtual server by using the NetScaler command line

The traffic management virtual server manages traffic between the client and the web application server. You can use either a load balancing or a content switching virtual server as the traffic management server. The SSO configuration is the same for either type.

To create a load balancing virtual server, at the command prompt, type the following command:

```
add lb vserver <vserverName> <type> <IP> <port>
```

For the variables, substitute the following values:

- **vserverName**—A name for the NetScaler appliance to use to refer to this virtual server.
- **type**—The protocol used by the service, either HTTP or MSSQLSVC.
- **IP**—The IP address assigned to the virtual server. This would normally be an IANA-reserved, non-public IP address on your LAN.
- **port**—The port on which the service listens. HTTP services normally listen on port 80. Secure HTTPS services normally listen on port 443.

Example

To add a load balancing virtual server called tmvserver1 to a configuration that manages HTTP traffic on port 80, assigning it a LAN IP address of 10.217.28.20 and then binding the load balancing virtual server to the wasservice1 service, you would type the following commands:

```
add lb vserver tmvserver1 HTTP 10.217.28.20 80
bind lb vserver tmvserv1 wasservice1
```

To create an authentication virtual server by using the NetScaler command line

The authentication virtual server manages authentication traffic between the client and the authentication (LDAP) server. To create an authentication virtual server, at the command prompt type the following commands:

- add authentication vserver <authvserverName> SSL <IP> 443
- set authentication vserver <authvservername> –authenticationdomain <domain>

For the variables, substitute the following values:

- **authvserverName** —A name for the NetScaler appliance to use to refer to this authentication virtual server. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the authentication virtual server is added by using the rename authentication vserver command.
- **IP**—The IP address assigned to the authentication virtual server. As with the traffic management virtual server, this address would normally be an IANA-reserved, non-public IP on your LAN.
- **domain**—The domain assigned to the virtual server. This would usually be the domain of your network. It is customary, though not required, to enter the domain in all capitals when configuring the authentication virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

To add an authentication virtual server called authverver1 to your configuration and assign it the LAN IP 10.217.28.21 and the domain EXAMPLE.COM, you would type the following commands:

```
add authentication vserver authvserver1 SSL 10.217.28.21 443
set authentication vserver authvserver1 –authenticationdomain EXAMPLE.COM
```

To configure a traffic management virtual server to use an authentication profile

The authentication virtual server can be configured to handle authentication for a single domain or for multiple domains. If it is configured to support authentication for multiple domains, you must also specify the domain for NetScaler SSO by creating an authentication profile, and then configuring the traffic management virtual server to use that authentication profile.

Note: The traffic management virtual server can be either a load balancing (lb) or content switching (cs) virtual server. The following instructions assume that you are using a load balancing virtual server. To configure a content switching virtual server, simply substitute set cs vserver for set lb vserver. The procedure is otherwise the same.
To create the authentication profile, and then configure the authentication profile on a traffic management virtual server, type the following commands:

- add authentication authnProfile <authnProfileName> {-authvserverName <string>} {-authenticationHost <string>} {-authenticationDomain <string>}
- set lb vserver <vserverName> -authnProfile <authnprofileName>

For the variables, substitute the following values:

- **authnprofileName**— A name for the authentication profile. Must begin with a letter, number, or the underscore character (_), and must consist of from one to thirty-one alphanumeric or hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters.
- **authvserverName**— The name of the authentication virtual server that this profile uses for authentication.
- **authenticationHost**— Host name of the authentication virtual server.
- **authenticationDomain**— Domain for which NetScaler SSO handles authentication. Required if the authentication virtual server performs authentication for more than one domain, so that the correct domain is included when the NetScaler appliance sets the traffic management virtual server cookie.

Example

To create an authentication profile named authnProfile1 for authentication of the example.com domain, and to configure the load balancing virtual server vserver1 to use the authentication profile authnProfile1, you would type the following commands:

```
add authentication authnProfile authnProfile1 -authnvsName authvsesrver1
    -authenticationHost authvsesrver1 -authenticationDomain example.com
set lb vserver vserver1 -authnProfile authnProfile1
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SSO

Oct 13, 2013

Configuring NetScaler SSO to authenticate by impersonation is simpler than configuring than SSO to authenticate by delegation, and is therefore usually preferable when your configuration allows it. You just create a KCD account. You can use the user's password.

If you do not have the user's password, you can configure NetScaler SSO to authenticate by delegation. Although somewhat more complex than configuring SSO to authenticate by impersonation, the delegation method provides flexibility in that a user's credentials might not be available to the NetScaler appliance in all circumstances.

For either impersonation or delegation, you must also enable integrated authentication on the web application server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Integrated Authentication on the Web Application Server

Oct 14, 2013

To set up NetScaler Kerberos SSO on each web application server that Kerberos SSO will manage, use the configuration interface on that server to configure the server to require authentication. Select Kerberos (negotiate) authentication by preference, with fallback to NTLM for clients that do not support Kerberos.

Following are instructions for configuring Microsoft Internet Information Server (IIS) to require authentication. If your web application server uses software other than IIS, consult the documentation for that web server software for instructions.

To configure Microsoft IIS to use integrated authentication

1. Log on to the IIS server and open Internet Information Services Manager.
2. Select the web site for which you want to enable integrated authentication. To enable integrated authentication for all IIS web servers managed by IISM, configure authentication settings for the Default Web Site. To enable integrated authentication for individual services (such as Exchange, Exadmin, ExchWeb, and Public), configure these authentication settings for each service individually.
3. Open the Properties dialog box for the default web site or for the individual service, and click the Directory Security tab.
4. Beside Authentication and Access Control, select Edit.
5. Disable anonymous access.
6. Enable Integrated Windows authentication (only). Enabling integrated Windows authentication should automatically set protocol negotiation for the web server to Negotiate, NTLM, which specifies Kerberos authentication with fallback to NTLM for non-Kerberos capable devices. If this option is not automatically selected, manually set protocol negotiation to Negotiate, NTLM.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting Up SSO by Impersonation

Mar 18, 2015

You can configure the KCD account for NetScaler SSO by impersonation. In this configuration, the NetScaler appliance obtains the user's username and passwordwhen the user authenticates to the authentication server and uses those credentials to impersonate the user to obtain a ticket-granting ticket (TGT). If the user's name is in UPN format, the appliance obtains the user's realm from UPN. Otherwise, it obtains the user's name and realm by extracting it from the SSO domain used during initial authentication, or from the session profile.

When configuring the KCD account, you must set the realm parameter to the realm of the service that the user is accessing. The same realm is also used as the user's realm if the user's realm cannot be obtained from authentication with the Netscaler appliance or from the session profile.

To create the KCD account for SSO by impersonation with a password

At the command prompt, type the following command:

add aaa kcdaccount <accountname> -realmStr <realm>
For the variables, substitute the following values:

- **accountname**—The KCD account name.
- **realm**—The domain assigned to NetScaler SSO.

Example:

To add a KCD account named kcdccount1, and use the keytab named kcdvserver.keytab, you would type the following command:

add aaa kcdAccount kcdaccount1 -keytab kcdvserver.keytab

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SSO by Delegation

Oct 13, 2013

To configure SSO by Delegation, you need to perform the following tasks:

- If you are configuring delegation by delegated user certificate, install the matching CA certificates on the NetScaler appliance and add them to the NetScaler configuration.
- Create the KCD account on the appliance. The appliance uses this account to obtain service tickets for your protected applications.
- Configure the Active Directory server.

Installing the Client CA Certificate on the NetScaler appliance

If you are configuring NetScaler SSO with a client certificate, you must copy the matching CA certificate for the client certificate domain (the client CA certificate) to the NetScaler appliance, and then install the CA certificate. To copy the client CA certificate, use the file transfer program of your choice to transfer the certificate and private-key file to the NetScaler appliance, and store the files in /nsconfig/ssl.

To install the client CA certificate on the NetScaler appliance

At the command prompt, type the following command:

add ssl certKey <certkeyName> -cert <cert> [(-key <key> [-password]) | -fipsKey <fipsKey>] [-inform (DER | PEM)] [-expiryMonitor (ENABLED | DISABLED | UNSET) [-notificationPeriod <positive_integer>]] [-bundle (YES | NO)]

For the variables, substitute the following values:

- **certkeyName**—A name for the client CA certificate. Must begin with an ASCII alphanumeric or underscore (_) character, and must consist of from one to thirty-one characters. Allowed characters include the ASCII alphanumerics, underscore, hash (#), period(.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the certificate-key pair is created. If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cert" or 'my cert').
- **cert**—Full path name and file name of the X509 certificate file used to form the certificate-key pair. The certificate file must be stored on the NetScaler appliance, in the /nsconfig/ssl/ directory.
- **key**—Full path name and file name of the file that contains the private key to the X509 certificate file. The key file must be stored on the NetScaler appliance in the /nsconfig/ssl/ directory.
- **password**—If a private key is specified, the passphrase used to encrypt the private key. Use this option to load encrypted private keys in PEM format.
- **fipsKey**—Name of the FIPS key that was created inside the Hardware Security Module (HSM) of a FIPS appliance, or a key that was imported into the HSM.
  Note: You can specify either a key or a fipsKey, but not both.
- **inform**—Format of the certificate and private-key files, either PEM or DER.
- **passplain**—Pass phrase used to encrypt the private key. Required when adding an encrypted private-key in PEM format.
- **expiryMonitor**—Configure the NetScaler appliance to issue an alert when the certificate is about to expire. Possible values: ENABLED, DISABLED, UNSET.
- **notificationPeriod**—If expiryMonitor is ENABLED, number of days before the certificate expires to issue an alert.
- **bundle**—Parse the certificate chain as a single file after linking the server certificate to its issuer's certificate within the file. Possible values: YES, NO.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

The following example adds the specified delegated user certificate customer-cert.pem to the NetScaler configuration along with the key customer-key.pem, and sets the password, certificate format, expiration monitor, and notification period.

To add the delegated user certificate, you would type the following commands:

```
add ssl certKey customer -cert "/nsconfig/ssl/customer-cert.pem"
    -key "/nsconfig/ssl/customer-key.pem" -password "dontUseDefaultPWs!"
    -inform PEM -expiryMonitor ENABLED [-notificationPeriod 14
```

Creating the KCD Account

If you are configuring NetScaler SSO by delegation, you can configure the KCD account to use the user's log-on name and password, to use the user's log-on name and keytab, or to use the user's client certificate. If you configure SSO with user name and password, the NetScaler appliance uses the delegated user account to obtain a Ticket Granting Ticket (TGT), and then uses the TGT to obtain service tickets for the specific services that each user requests. If you configure SSO with keytab file, the NetScaler appliance uses the delegated user account and keytab information. If you configure SSO with a delegated user certificate, the NetScaler appliance uses the delegated user certificate.

To create the KCD account for SSO by delegation with a password

At the command prompt, type the following commands:

```
add aaa kcdaccount <accountname> -delegatedUser root -kcdPassword <password> -realmStr <realm>
```

For the variables, substitute the following values:

- **accountname**—A name for the KCD account.
- **password**—A password for the KCD account.
- **realm**—The realm of the KCD account, usually the domain for which SSO is active.

Example (UPN Format)

To add a KCD account named kcdaccount1 to the NetScaler appliance configuration with a password of password1 and a realm of EXAMPLE.COM, specifying the delegated user account in UPN format (as root), you would type the following commands:

```
add aaa kcdaccount kcdaccount1 –delegatedUser root
-kcdPassword password1 -realmStr EXAMPLE.COM
```

Example (SPN Format)

To add a KCD account named kcdaccount1 to the NetScaler appliance configuration with a password of password1 and a realm of EXAMPLE.COM, specifying the delegated user account in SPN format, you would type the following commands:

```
add aaa kcdAccount kcdaccount1 -realmStr EXAMPLE.COM
-delegatedUser "host/kcdvserver.example.com" -kcdPassword password1
```

Creating the KCD account for SSO by delegation with a keytab

If you plan to use a keytab file for authentication, first create the keytab. You can create the keytab file manually by logging onto the AD server and using the ktpass utility, or you can use the NetScaler configuration utility to create a batch script, and then run that script on the AD server to generate the keytab file. Next, use FTP or another file transfer program to transfer the keytab file to the NetScaler appliance and place it in the /nsconfig/krb directory. Finally, configure the KCD account for NetScaler SSO by delegation and provide the path and file name of the keytab file to the NetScaler appliance.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To create the keytab file manually

Log on to the AD server command line and, at the command prompt, type the following command:

ktpass /princ <SPN> /ptype KRB5_NT_PRINCIPAL /mapuser <DOMAIN>\<username> /pass <password> -out <File_Path>
For the variables, substitute the following values:

- **SPN**—The service principal name for the KCD service account.
- **DOMAIN**—The domain of the Active Directory server.
- **username**—The KSA account username.
- **password**—The KSA account password.
- **path**— The full path name of the directory in which to store the keytab file after it is generated.

To use the NetScaler configuration utility to create a script to generate the keytab file.

1. Navigate to Security > AAA - Application Traffic
2. In the data pane, under Kerberos Constrained Delegation, click Batch file to generate Keytab.
3. In the Generate KCD (Kerberos Constrained Delegation) Keytab Script dialog box, set the following parameters:
   - **Domain User Name**—The KSA account username.
   - **Domain Password**—The KSA account password.
   - **Service Principal**—The service principal name for the KSA.
   - **Output File Name**—The full path and file name to which to save the keytab file on the AD server.
4. Clear the Create Domain User Account check box.
5. Click Generate Script.
6. Log on to the Active Directory server and open a command line window.
7. Copy the script from the Generated Script window and paste it directly into the Active Directory server command-line window. The keytab is generated and stored in the directory under the file name that you specified as **Output File Name**.
8. Use the file transfer utility of your choice to copy the keytab file from the Active Directory server to the NetScaler appliance and place it in the /nsconfig/krb directory.

To create the KCD account

At the command prompt, type the following command:

add aaa kcdaccount <accountname> –keytab <keytab>
Example:

To add a KCD account named kcdccount1, and use the keytab named kcdvserver.keytab, you would type the following commands:

add aaa kcdaccount kcdaccount1 –keytab kcdvserver.keytab

To create the KCD account for SSO by delegation with a delegated user cert

At the command prompt, type the following command:

add aaa kcdaccount <accountname> -realmStr <realm> -delegatedUser <user_name/SPN> -usercert <cert> -cacert <cacert>
For the variables, substitute the following values:

- **accountname**—A name for the KCD account.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- **realmStr**—The realm for the KCD account, usually the domain for which SSO is configured.
- **delegatedUser**—The delegated user name, in SPN format.
- **usercert**—The full path and name of the delegated user certificate file on the NetScaler appliance. The delegated user certificate must contain both the client certificate and the private key, and must be in PEM format. If you use smart card authentication, you might need to create a smart card certificate template to allow certificates to be imported with the private key.
- **cacert**—The full path to and name of the CA certificate file on the NetScaler appliance.

Example:

To add a KCD account named kcdccount1, and use the keytab named kcdvserver.keytab, you would type the following command:

```
add aaa kcdaccount kcdaccount1 -realmStr EXAMPLE.COM
    -delegatedUser "host/kcdvserver.example.com" -usercert /certs/usercert
    -cacert /cacerts/cacert
```

Setting up Active Directory for NetScaler SSO

When you configure SSO by delegation, in addition to creating the KCDAccount on the NetScaler appliance, you must also create a matching Kerberos Service Account (KSA) on your LDAP active directory server, and configure the server for SSO. To create the KSA, use the account creation process on the active directory server. To configure SSO on the active directory server, open the properties window for the KSA. In the Delegation tab, enable the following options: Trust this user for delegation to specified services only and Use any Authentication protocol. (The Kerberos only option does not work, because it does not enable protocol transition or constrained delegation.) Finally, add the services that NetScaler SSO will manage.

Note: If the Delegation tab is not visible in the KSA account properties dialog box, before you can configure the KSA as described, you must use the Microsoft setspn command-line tool to configure the active directory server so that the tab is visible.

To configure delegation for the Kerberos service account

1. In the LDAP account configuration dialog box for the Kerberos service account that you created, click the Delegation tab.
2. Choose "Trust this user for delegation to the specified services only".
3. Under "Trust this user for delegation to the specified services only," choose "Use any authentication protocol".
4. Under “Services to which this account can present delegated credentials,” click Add.
5. In the Add Services dialog box, click Users or Computers, choose the server that hosts the resources to be assigned to the service account, and then click OK.
   Note: Constrained delegation does not support services hosted in domains other than the domain assigned to the account, even though Kerberos might have a trust relationship with other domains
6. Back in the Add Services dialog box, in the Available Services list, choose the services assigned to the service account. NetScaler SSO supports the HTTP and MSSQLSVC services.
7. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Generating the KCD Keytab Script

Aug 13, 2014

The KCD Keytab Script dialog box generates the keytab script, which in turn generates the keytab file necessary to configure KCD on the NetScaler ADC.

To generate the KCD keytab script by using the configuration utility

1. Navigate to Security > AAA - Application Traffic
2. In the details pane, under Kerberos Constrained Delegation, click Batch file to generate keytab.
3. In the Generate KCD (Kerberos Constrained Delegation) Keytab Script dialog box, fill out the fields as described below.
   - **Domain User Name:** The name of the domain user.
   - **Domain Password:** The password for the domain user.
   - **Service Principal:** The service principal.
   - **Output File Name:** A filename for the KCD script file.
   - **Create Domain User Account:** Select this check box to create the specified domain user account.
4. Click Generate Script to generate the script. The script is generated, and appears in the Generated Script text box below the Generate Script button.
5. Copy the script, and save it as a file on your AD domain controller. You must now run this script on the domain controller to generate the keytab file, and then copy the keytab file to the /nsconfig/krb/ directory on the NetScaler appliance.
6. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SAML Authentication

Jul 21, 2015

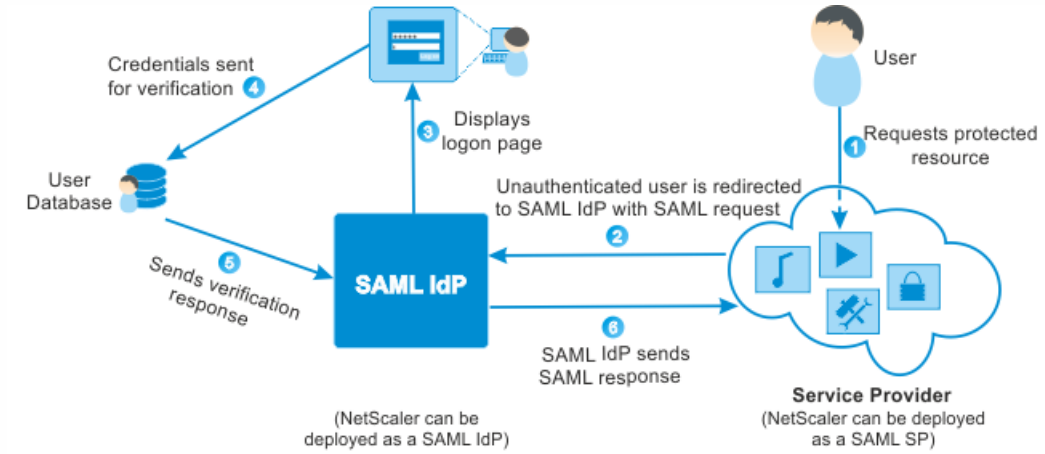
Security Assertion Markup Language (SAML) is an XML-based authentication mechanism that provides single sign-on capability and is defined by the OASIS Security Services Technical Committee.

Why SAML?

Consider a scenario in which a service provider (LargeProvider) hosts a number of applications for a customer (BigCompany). BigCompany has users that must seamlessly access these applications. In a traditional setup, LargeProvider would need to maintain a database of users of BigCompany. This raises some concerns for each of the following stakeholders:

- LargeProvider must ensure security of user data.
- BigCompany must validate the users and keep the user data up-to-date, not just in its own database, but also in the user database maintained by LargeProvider. For example, a user removed from the BigCompany database must also be removed from the LargeProvider database.
- A user has to log on individually to each of the hosted applications.

The SAML authentication mechanism provides an alternative approach. The following deployment diagram shows how SAML works.



The concerns raised by traditional authentication mechanisms are resolved as follows:

- LargeProvider does not have to maintain a database for BigCompany users. Freed from identity management, LargeProvider can concentrate on providing better services.
- BigCompany does not bear the burden of making sure the LargeProvider user database is kept in sync with its own user database.
- A user can log on once, to one application hosted on LargeProvider, and be automatically logged on to the other applications that are hosted there.

The NetScaler appliance can be deployed as a SAML Service Provider (SP) and a SAML Identity Provider (IdP). Read through the relevant topics to understand the configurations that must be performed on the NetScaler appliance.

The following table lists some articles that are specific to deployments where the NetScaler appliance is used as a SAML SP or a SAML IdP.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SAML SP	SAML IdP	Information Link
NetScaler	Citrix AppController Z3	http://support.citrix.com/article/CTX133820
NetScaler	CloudGateway	http://support.citrix.com/article/CTX133558
NetScaler	Microsoft AD FS 2.0	http://support.citrix.com/article/CTX133919
NetScaler	Shibboleth	http://support.citrix.com/article/CTX200271
NetScaler	Shibboleth (With SAML single logout configuration)	http://support.citrix.com/article/CTX200392
Siteminder	NetScaler	http://support.citrix.com/article/CTX200177
ShareFile	NetScaler	http://support.citrix.com/article/CTX200323

Some information on other specific deployments:

- NetScaler as SAML SP on FIPS Device
- Configuring Office365 for Single Sign-on with NetScaler as SAML IdP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler as a SAML SP

Jan 04, 2016

The SAML Service Provider (SP) is a SAML entity that is deployed by the service provider. When a user tries to access a protected application, the SP evaluates the client request. If the client is unauthenticated (does not have a valid NSC_TMAA or NSC_TMAS cookie), the SP redirects the request to the SAML Identity Provider (IdP).

The SP also validates SAML assertions that are received from the IdP.

When the NetScaler appliance is configured as an SP, all user requests are received by a traffic management virtual server (load balancing or content switching) that is associated with the relevant SAML action.

> ## Note
>
> A NetScaler appliance can be used as a SAML SP in a deployment where the SAML IdP is configured either on the appliance or on any external SAML IdP.

When used as a SAML SP, a NetScaler appliance:

- Can extract the user information (attributes) from the SAML token. This information can then be used in the policies that are configured on the NetScaler. For example, if you want to extract the GroupMember and emailaddress attributes, in the SAMLAction, specify the **Attribute2** parameter as GroupMember and the **Attribute3** parameter as emailaddress.

  Note: Default attributes such as username, password, and logout URL must not be extracted in attributes 1 to 16, because they as are implicitly parsed and stored in the session.

- Can extract attribute names of upto 127 bytes from an incoming SAML assertion. The previous limit was 63 bytes. Support introduced in NetScaler 11.0 Build 64.x.

- Supports post, redirect, and artifact bindings. Support for redirect and artifact bindings is introduced in NetScaler 11.0 Build 55.x.

  Note: Redirect binding should not be used for large amount of data, when the assertion after inflate or decoding is greater than 10K.

- Can decrypt assertions. Support introduced in NetScaler 11.0 Build 55.x.

- Can extract multi-valued attributes from a SAML assertion. These attributes are sent is nested XML tags such as:

  ```
  <AttributeValue>
     <AttributeValue>Value1</AttributeValue>
     <AttributeValue>Value2</AttributeValue
  </AttributeValue>
  ```

  When presented with above XML, the NetScaler appliance can extract both Value1 and Value2 as values of a given

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attribute, as opposed to the old firmware that extracts only Value1.

Note: Support introduced in NetScaler 11.0 Build 64.x.

- Can specify the validity of a SAML assertion.

  If the system time on NetScaler SAML IdP and the peer SAML SP is not in sync, the messages might get invalidated by either party. To avoid such cases, you can now configure the time duration for which the assertions will be valid.

  This duration, called the "skew time," specifies the number of minutes for which the message should be accepted. The skew time can be configured on the SAML SP and the SAML IdP.

  Note: Support introduced in NetScaler 11.0 Build 64.x.

To configure the NetScaler appliance as a SAML SP by using the command line interface

1. Configure a SAML SP action.

   Example: The following command adds a SAML action that redirects unauthenticated user requests.
   > **add authentication samlAction** SamlSPAct1 -samlIdPCertName nssp –samlRedirectUrl https://auth1.example.com

2. Configure the SAML policy.

   Example: The following command defines a SAML policy that applies the above defined SAML action to all traffic.
   > **add authentication samlPolicy** SamlSPPol1 ns_true SamlSPAct1

3. Bind the SAML policy to the authentication virtual server.

   Example: The following command binds the SAML policy to a authentication virtual server named "av_saml".
   > **bind authentication vserver** av_saml -policy SamlSPPol1

4. Bind the authentication virtual server to the appropriate traffic management virtual server.

   Example: The following command adds a load balancing virtual server named "lb1_ssl" and associates the authentication virtual server named "av_saml" to the load balancing virtual server.
   > **add lb vserver** lb1_ssl SSL 10.217.28.224 443 -persistenceType NONE -cltTimeout 180 -AuthenticationHost auth1.example.com -Authentication ON -authnVsName av_saml

To configure a NetScaler appliance as a SAML SP by using the graphical user interface

1. Configure the SAML action and policy.

   Navigate to **Security** > **AAA - Application Traffic** > **Policies** > **Authentication** > **Advanced Policies** > **Policy**, create a policy with SAML as the action type, and associate the required SAML action with the policy.

2. Associate the SAML policy with an authentication virtual server.

   Navigate to **Security** > **AAA - Application Traffic** > **Virtual Servers**, and associate the SAML policy with the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

authentication virtual server.

3. Associate the authentication server with the appropriate traffic management virtual server.

   Navigate to **Traffic Management** > **Load Balancing** (or **Content Switching**) > **Virtual Servers**, select the virtual server, and associate the authentication virtual server with it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler as a SAML IdP

Jan 04, 2016

The SAML IdP (Identity Provider) is a SAML entity that is deployed on the customer network. The IdP receives requests from the SAML SP and redirects users to a logon page, where they must enter their credentials. The IdP authenticates these credentials with the user directory (external authentication server, such as LDAP) and then generates a SAML assertion that is sent to the SP.

The SP validates the token, and the user is then granted access to the requested protected application.

When the NetScaler appliance is configured as an IdP, all requests are received by an authentication virtual server that is associated with the relevant SAML IdP profile.

> **Note**
>
> A NetScaler appliance can be used as a IdP in a deployment where the SAML SP is configured either on the appliance or on any external SAML SP.

When used as a SAML IdP, a NetScaler appliance:

- Supports all authentication methods that it supports for traditional logons.
- Digitally signs assertions. Support for the SHA256 algorithm is introduced in NetScaler 11.0 Build 55.x.
- Supports single-factor and two-factor authentication. SAML must not be configured as the secondary authentication mechanism.
- Can encrypt assertions by using the public key of the SAML SP. This is recommended when the assertion includes sensitive information. Support introduced in NetScaler 11.0 Build 55.x.
- Can be configured to accept only digitally signed requests from the SAML SP. Support introduced in NetScaler 11.0 Build 55.x.
- Can log on to the SAML IdP by using the following 401-based authentication mechanisms: Negotiate, NTLM, and Certificate. Support introduced in NetScaler 11.0 Build 55.x.
- Can be configured to send 16 attributes in addition to the NameId attribute. The attributes must be extracted from the appropriate authentication server. For each of them, you can specify the name, the expression, the format, and a friendly name in the SAML IdP profile. Support introduced in NetScaler 11.0 Build 55.x.
- If the NetScaler appliance is configured as a SAML IdP for multiple SAML SP, a user can gain access to applications on the different SPs without explicitly authenticating every time. The NetScaler appliance creates a session cookie for the first authentication, and every subsequent request uses this cookie for authentication. Support introduced in NetScaler 11.0 Build 55.x.
- Can send multi-valued attributes in a SAML assertion. Support introduced in NetScaler 11.0 Build 64.x.
- Supports post and redirect bindings. Support for redirect bindings is introduced in NetScaler 11.0 Build 64.x.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- Can specify the validity of a SAML assertion.

  If the system time on NetScaler SAML IdP and the peer SAML SP is not in sync, the messages might get invalidated by either party. To avoid such cases, you can now configure the time duration for which the assertions will be valid.

  This duration, called the "skew time," specifies the number of minutes for which the message should be accepted. The skew time can be configured on the SAML SP and the SAML IdP.

  Note: Support introduced in NetScaler 11.0 Build 64.x.

- Can be configured to serve assertions only to SAML SPs that are pre-configured on or trusted by the IdP. For this configuration, the SAML IdP must have the service provider ID (or issuer name) of the relevant SAML SPs. Support introduced in NetScaler 11.0 Build 64.x.

> **Note**
>
> Before proceeding, make sure that you have an authentication virtual server that is linked to an LDAP authentication server.

To configure a NetScaler appliance as a SAML IdP by using the command line interface

1. Configure a SAML IdP profile.

   Example: Adding NetScaler as an IdP with SiteMinder as the SP.
   > **add authentication samlIdPProfile** samlIDPProf1 -samlSPCertName siteminder-cert -encryptAssertion ON -samlIdPCertName ns-cert -assertionConsumerServiceURL http://sm-proxy.nsi-test.com:8080/affwebservices/public/saml2assertionconsumer -rejectUnsignedRequests ON -signatureAlg RSA-SHA256 -digestMethod SHA256

2. Configure the SAML authentication policy and associate the SAML IdP profile as the action of the policy.

   > **add authentication samlIdPPolicy** samlIDPPol1 -rule ns_true -action samlIDPProf1

3. Bind the policy to the authentication virtual server.

   > **bind authentication vserver** saml-auth-vserver -policy samlIDPPol1 -priority 100

To configure a NetScaler appliance as a SAML IdP by using the graphical user interface

1. Configure the SAML IdP profile and policy.

   Navigate to **Security** > **AAA - Application Traffic** > **Policies** > **Authentication** > **Advanced Policies** > **Policy**, and create a policy with SAML IdP as the action type, and associate the required SAML IdP profile with the policy.

2. Associate the SAML IdP policy with an authentication virtual server.

   Navigate to **Security** > **AAA - Application Traffic** > **Virtual Servers**, and associate the SAML IdP policy with the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authentication virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SAML Single Sign-on

Oct 13, 2015

To provide single sign-on capabilities across applications that are hosted on the service provider, you can configure SAML single sign-on on the SAML SP.

Configuring SAML single sign-on by using the command line interface

1. Configure the SAML SSO profile.

   Example: In the following command, https://nssp2.example.com is the load balancing virtual server that has a web link from the SharePoint portal. Nssp.example.com is the Traffic Management virtual server that is load balancing the SharePoint server.
   > **add tm samlSSOProfile** tm-saml-sso -samlSigningCertName nssp -assertionConsumerServiceURL "https://nssp2.example.com/cgi/samlauth" -relaystateRule "\"https://nssp2.example.com/samlsso.html\"" -sendPassword ON -samlIssuerName nssp.example.com

2. Associate the SAML SSO profile with the traffic action.

   Example: The following command enables SSO and binds the SAML SSO profile created above to a traffic action.
   > **add tm trafficAction** html_act -SSO ON -samlSSOProfile tm-saml-sso

3. Configure the traffic policy that specifies when the action must be executed.

   Example: The following command associates the traffic action with a traffic policy.
   > **add tm trafficPolicy** html_pol "HTTP.REQ.URL.CONTAINS(\"abc.html\")" html_act

4. Bind the traffic policy created above to a traffic management virtual server (load balancing or content switching). Alternatively, the traffic policy can be associated globally.

   Note: This traffic management virtual server must be associated with the relevant authentication virtual sever that is associated with the SAML action.

   > **bind lb vserver** lb1_ssl -policyName html_pol -priority 100 -gotoPriorityExpression END -type REQUEST

Configuring SAML single sign-on by using the graphical user interface

1. Define the SAML SSO profile, the traffic profile, and the traffic policy.

   Navigate to **Security** > **AAA - Application Traffic** > **Policies** > **Traffic**, select the appropriate tab, and configure the settings.

2. Bind the traffic policy to a traffic management virtual server or globally to the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

OAuth Authentication

Jul 29, 2015

The NetScaler AAA-TM feature now supports OAuth and OpenID-Connect mechanisms for authenticating and authorizing users to applications that are hosted on applications such as Google, Facebook, and Twitter.

> **Note**
>
> OAuth on NetScaler is currently qualified only for Google applications.

A major advantage is that user information is not sent to the hosted applications and therefore the risk of identity theft is considerably reduced.

In the NetScaler implementation, the application to be accessed is represented by the AAA-TM virtual server. So, to configure OAuth, you must configure an OAuth policy which which must then be associated with a AAA-TM virtual server.

To configure OAuth by using the command line interface

1. Define an OAuth action.
   > **add authentication OAuthAction** <name> -authorizationEndpoint <URL> -tokenEndpoint <URL> [-idtokenDecryptEndpoint <URL>] -clientID <string> -clientSecret <string> [-defaultAuthenticationGroup <string>] [-Attribute1 <string>] [-Attribute2 <string>] [-Attribute3 <string>] ...

2. Associate the action with an advanced authentication policy.
   > **add authentication Policy** <name> -rule <expression> -action <string>

Example

add authentication oauthAction a -authorizationEndpoint https://example.com/ -tokenEndpoint https://example.com/ -clientiD sadf -clientsecret df

> **Note**
>
> - Refer to the man page for information on the parameters.
> - Attributes (1 to 16) can be extracted in the OAuth response. Currently, these attributes are not evalauted. They are added for future reference.

To configure OAuth by using the graphical user interface

1. Configure the OAuth action and policy.
   Navigate to **Security** > **AAA - Application Traffic** > **Policies** > **Authentication** > **Advanced Policies** > **Policy**, and create a policy with OAuth as the action type, and associate the required OAuth action with the policy.

2. Associate the OAuth policy with an authentication virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Navigate to **Security** > **AAA - Application Traffic** > **Virtual Servers**, and associate the OAuth policy with the authentication virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Multi-Factor (nFactor) Authentication

Sep 24, 2015

> **Important**
>
> Supported from NetScaler 11.0 Build 62.x onwards.

Multi-factor authentication enhances the security of an application by requiring users to provide multiple proofs of identify to gain access. The NetScaler appliance provides an extensible and flexible approach to configuring multi-factor authentication. This approach is called *nFactor authentication*.

With nFactor authentication you can:

- Configure any number of authentication factors.
- Base the selection of the next factor on the result of executing the previous factor.
- Customize the login interface. For example, you can customize the label names, error messages, and help text.
- Extract user group information without doing authentication.
- Configure pass-through for an authentication factor. This means that no explicit login interaction is required for that factor.
- Configure the order in which different types of authentication are applied. Any of the authentication mechanisms that are supported on the NetScaler appliance can be configured as any factor of the nFactor authentication setup. These factors are executed in the order in which they are configured.
- Configure the NetScaler to proceed to an authentication factor that must be executed when authentication fails. To do so, you configure another authentication policy with the exact same condition, but with the next highest priority and with the action set to "NO_AUTH". You must also configure the next factor, which must specify the alternative authentication mechanism to apply.

For a nFactor benefit, see One Public IP for AAA-TM Deployments on NetScaler.

Sample deployments using nFactor authentication:

- Getting two passwords up-front, pass-through in next factor. Read
- Group extraction followed by certificate or LDAP authentication, based on group membership. Read
- SAML followed by LDAP or certificate authentication, based on attributes extracted during SAML. Read
- SAML in first factor, followed by group extraction, and then LDAP or certificate authentication, based on groups extracted. Read
- Prefilling user name from certificate. Read
- Certificate authentication followed by group extraction for 401 enabled traffic management virtual servers. Read
- Username and 2 passwords with group extraction in third factor. Read
- Certificate fallback to LDAP in same cascade; one virtual server for both certificate and LDAP authentication. Read
- LDAP in first factor and WebAuth in second factor. Read
- Domain drop down in first factor, then different policy evaluations based on group. Read

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How to Configure nFactor Authentication

Nov 23, 2015

The primary entity used for nFactor authentication is called a *login schema*.

A login schema specifies an authentication schema XML file that defines the manner in which the login form will be rendered. Considering the interaction that the user must have when logging in to the application, you can create a single file for multiple factors or different files for different factors. View sample XML file.

- **Single file for multiple factors.** User will be provided a single form in which to provide credentials for multiple authentication factors.

- **Different files for different factors.** User will be provided a different form for each authentication factor.

Next, you must associate the XML file(s) with login schema(s). You can also specify expressions to extract the user name and the password from the login form.

> ## Tip
>
> You can configure an authentication factor to be pass-through. This means that the user is not required to provide credentials explicitly and there is no login form for that factor. The credentials are either taken from the previous factor or the user name and/or password are dynamically extracted by using the username/password expressions that are configured for that login schema. You must set the login schema to "NOSCHEMA", instead of an XML file.

Now that the login schemas are configured, you must specify the manner in which they must be invoked. A login schema can be invoked by using either a login schema policy or an authentication policy label. The decision depends on the following:

- **Login schema policy.**
  - Specifies the condition on which the login form must be presented to the user.
  - Must be bound to an authentication virtual server.
  - In an authentication virtual server that has multiple login schema policies, the policy with the highest priority that evaluates to true is executed. That is, the login form associated with that policy is presented to the user.
  - The login schema policy is only used to present the first login form.

- **Authentication policy label.**
  - Specifies a collection of authentication policies for a particular factor. Each policy label corresponds to a single factor.
  - Specifies the login form that must be presented to the user.
  - Must be bound as the next factor of an authentication policy or of another authentication policy label.
  - Typically, a policy label includes authentication policies for a specific authentication mechanism. However, you can also have a policy label that has authentication policies for different authentication mechanisms.

To summarize, the configurations you must perform to set up nFactor authentication are as follows:

1. Create the authentication schema XML files.
2. Associate each XML file with a login schema.
3. Associate each login schema with a login schema policy or authentication policy label.
4. Bind login schema policy to an authentication virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Bind authentication policy label, as next factor, to an authentication policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

How nFactor Authentication Works

Aug 18, 2015

Imagine a user requesting access to an application that requires user credentials. As is the case in NetScaler deployments, the request arrives at the NetScaler through a traffic management virtual server (in this case, a load balancing virtual server). Since the user must provide authentication credentials, the load balancing virtual server redirects the request to the authentication virtual server, which does the following:

1. Checks to determine whether any login schema policies are associated with the authentication virtual server.

   - If yes, the user is presented the login form associated with the login schema policy with the highest priority that evaluates to true.

   - If no, the default login form is presented to the user.

   Note: The default login schema files are available in the */nsconfig/loginschema/LoginSchema/* directory of the NetScaler appliance. Citrix recommends that you copy these files to the */nsconfig/loginschema/* directory before using them, so that changes made to the files are preserved post reboot.

2. The authentication policies that are associated with the authentication virtual server are evaluated. For the policies that are evaluated to true, the actions are executed in order of priority until one of the actions succeeds.

3. The policy label that is associated as the next factor is invoked.

4. The authentication policies that are associated with the authentication policy label are evaluated. For the policies that are evaluated to true, the actions are executed in order of priority until one of the actions succeeds.

5. The policy label that is associated as the next factor is invoked.

6. Steps 4 and 5 are performed repetitively till all the configured next factors are executed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring nFactor Authentication

Nov 23, 2015

To understand the step-wise configurations for nFactor authentication, let us consider a 2-factor authentication deployment where the 1st factor is LDAP authentication and the 2nd factor is RADIUS authentication.

This sample deployment requires the user to login to both factors using a single login form. Therefore, we define a single login form that accepts two passwords. The first password is used for LDAP authentication and the other for RADIUS authentication. View XML file.

Here are the configurations that are performed.

1. Configure the load balancing virtual server for authentication

   > **add lb vserver** lbvs55 HTTP 1.217.193.55 80 -AuthenticationHost auth56.aaatm.com -Authentication ON

2. Configure the authentication virtual server.

   > **add authentication vserver** auth56 SSL 1.217.193.56 443 -AuthenticationDomain aaatm.com

3. Configure the login schema for the login form and bind it to a login schema policy.

   > **add authentication loginSchema** login1 -authenticationSchema login-2passwd.xml -userCredentialIndex 1 -passwordCredentialIndex 2

   > **add authentication loginSchemaPolicy** login1 -rule true -action login1

4. Configure a login schema for the pass-through and bind it to a policy label

   > **add authentication loginSchema** login2 -authenticationSchema noschema

   > **add authentication policylabel** label1 -loginSchema login2

5. Configure the LDAP and RADIUS policies.

   > **add authentication ldapAction** ldapAct1 -serverIP 1.217.28.180 -ldapBase "dc=aaatm, dc=com" -ldapBindDn administrator@aaatm.com -ldapBindDnPassword 71ca2b11ad800ce2787fb7deb54842875b8f3c360d7d46e3d49ae65c41550519 -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName samAccountName -groupAttrName memberOf -subAttributeName CN

   > **add authentication Policy** ldap -rule true -action ldapAct1

   > **add authentication radiusAction** radius -serverIP 1.217.22.20 -radKey a740d6a0aeb3288fa0a6fbe932d329acddd8f448ecb4a3038daa87b36599fd16 -encrypted -encryptmethod ENCMTHD_3 -radNASip ENABLED -radNASid NS28.50 -radAttributeType 11 -ipAttributeType 8

   > **add authentication Policy** radius -rule true -action radius

6. Bind the login schema policy to the authentication virtual server

   > **bind authentication vserver** auth56 -policy login1 -priority 1 -gotoPriorityExpression END

© 1999-2017 Citrix Systems, Inc. All rights reserved.

7. Bind the LDAP policy (first factor) to the authentication virtual server.

   > **bind authentication vserver** auth56 -policy ldap -priority 1 -nextFactor label1 -gotoPriorityExpression next

8. Bind the RADIUS policy (second factor) to the authentication policy label.

   > **bind authentication policylabel** label1 -policyName radius -priority 2 -gotoPriorityExpression end

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Firewall

Aug 17, 2017

The following topics cover installation and configuration of the Citrix Application Firewall feature.

Introduction	An overview of web application security and how the application firewall works.
Configuration	How to configure the application firewall to protect a web site, a web service, or a Web 2.0 site.
Signatures	A detailed description of the signatures feature and how to configure the signatures, add signatures from a supported vulnerability scanning tool, and define your own signatures, with examples.
Overview of Security Checks	A detailed description of all of the application firewall security checks, with configuration information and examples.
Profiles	A description of how profiles are configured and used in the application firewall.
Policies	A description of how policies are used when configuring the application firewall, with examples of useful policies.
Imports	A description of how the application firewall uses different types of imported files, and how to import and export files.
Global Configuration	A description of application firewall features that apply to all profiles, and how to configure them.
Use Cases	Extended examples that demonstrate how to set up the application firewall to best protect specific types of more complex web sites and web services.
Logs, Statistics, and Reports	How to access and use the application firewall logs, the statistics, and the reports to assist in configuring the application firewall.

The Citrix application firewall offers easy to configure options to meet a wide range of application security requirements. Application firewall profiles, which consist of sets of security checks, can be used to protect both the requests and the responses by providing deep packet-level inspections. Each profile includes an option to select basic protections or advanced protections. Some protections might require use of other files. For example, xml validation checks might require WSDL or schema files. The profiles can also use other files, such as signatures or error objects. These files can be added locally, or they can be imported ahead of time and saved on the appliance for future use. They can be shared by multiple profiles.

Profiles work in conjunction with the application firewall policies. Each policy identifies a type of traffic, and that traffic is inspected for the security check violations specified in the profile that is associated with the policy. The policies can have different bind points, which determine the scope of the policy. For example, a policy that is bound to a specific virtual server is invoked and evaluated for only the traffic flowing through that virtual server. The policies are evaluated in the order of their designated priorities, and the first one that matches the request or response is applied.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Quick Deployment of Application Firewall Protection

You can use the following procedure for quick deployment of application firewall security:

1. Add an appfw profile and select the appropriate type (html, xml, web2.0) for the security requirements of the application.
2. Select the required level of security (basic or advanced).
3. Add or import the required files, such as signatures or WSDL.
4. Configure the profile to use the files, and make any other necessary changes to the default settings.
5. Add an appfw policy for this profile.
6. Bind the policy to the target bind point and specify the priority.

Application firewall entities

Following are brief overviews of the application firewall entities. For details, see the Application Firewall Guide.

Profile—An application firewall profile specifies what to look for and what to do. It inspects both the request and the response to determine which potential security violations should be checked and what actions should be taken when processing a transaction. A profile can protect an HTML, XML or HTML and XML payload. Depending on the security requirements of the application, you can create either a basic or an advanced profile. A basic profile can protect against known attacks. If higher security is required, you can deploy an advanced profile to allow controlled access to the application resources, blocking zero day attacks. However, a basic profile can be modified to offer advanced protections, and vice versa. Multiple action choices (for example, block, log, learn, and transform) are available. Advanced security checks might use session cookies and hidden form tags for controlling and monitoring the client connections. Application firewall profiles can learn the triggered violations and suggest the relaxation rules.

Basic Protections—A basic profile includes a preconfigured set of Start URL and Deny URL relaxation rules. These relaxation rules determine which requests should be allowed and which should be denied. Incoming requests are matched against these lists and the configured actions are applied. This allows the user to be able to secure applications with minimal configuration for relaxation rules. The Start URL rules protect against forceful browsing. Known web server vulnerabilities that are exploited by hackers can be detected and blocked by enabling a set of default Deny URL rules. Commonly launched attacks, such as Buffer Overflow, SQL, or Cross-site scripting can also be easily detected.

Advanced Protections—As the name indicates, advanced protections are used for applications that have higher security requirements. Relaxation rules are configured to allow access to only specific data and block the rest. This positive security model mitigates unknown attacks, which might not be detected by basic security checks. In addition to all the basic protections, an advanced profile keeps track of a user session by controlling the browsing, checking for cookies, specifying input requirements for various form fields, and protecting against tampering of forms or cross-site request forgery attacks. Learning, which observes the traffic and deploys the appropriate relaxations, is enabled by default for many security checks. Although easy to use, advanced protections require due consideration, because they offer tighter security but also require more processing and do not allow use of caching, which can affect performance.

Import—Import functionality is useful when application firewall profiles need to use external files, that is, files hosted on an external or internal web server, or that have to be copied from a local machine. Importing a file and storing it on the appliance is very useful, especially in situations where you have to control access to external websites, or where compilation takes a long time, large files have to be synced across HA deployments, or you can reuse a file by copying it across multiple devices. For example:

- WSDLs hosted on external web servers can be imported locally before blocking access to external websites.
- Large signature files generated by an external scan tool such as Cenzic can be imported and precompiled, using schema on the Citrix appliance.
- A customized HTML or XML error page can be imported from an external web server or copied from a local file.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Signatures—Signatures are very powerful, because they use pattern matching to detect malicious attacks and can be configured to check both the request and the response of a transaction. They are a preferred option when a customizable security solution is needed. Multiple choices (for example, block, log, learn, and transform) are available for the action to take when a signature match is detected. The application firewall has a built-in default signature object consisting of more than 1,300 signature rules, with an option to get the latest rules by using the auto-update feature. Rules created by other scan tools can also be imported. The signature object can be customized by adding new rules, which can work in conjunction with the other security checks specified in the application firewall profile. A signature rule can have multiple patterns and can flag a violation only when all the patterns are matched, thereby avoiding false positives. Careful selection of a literal fastmatch pattern for a rule can significantly optimize processing time.

Policies—Application Firewall Policies are used to filter and separate the traffic into different types. This provides the flexibility to implement different levels of security protections for the application data. Access to highly sensitive data can be directed to advanced security-check inspections, while less sensitive data is protected by basic-level security inspections. Policies can also be configured to bypass security-check inspection for harmless traffic. Higher security requires more processing, so careful design of the policies can provide desired security along with optimized performance. The priority of the policy determines the order in which it is evaluated, and its bind point determines the scope of its application.

Highlights

1. Ability to secure a wide range of applications by protecting different types of data, implementing the right level of security for different resources, and still getting maximum performance.
2. Flexibility to add or modify a security configuration. You can tighten or relax security checks by enabling or disabling basic and advanced protections.
3. Option to convert an HTML profile to an XML or Web2.0 (HTML+XML) profile and vice-versa, providing the flexibility to add security for different types of payload.
4. Easily deployed actions to block attacks, monitor them in logs, collect statistics, or even transform some attack strings to render them harmless.
5. Ability to detect attacks by inspecting incoming requests, and to prevent leakage of sensitive data by inspecting the responses sent by the servers.
6. Capability to learn from the traffic pattern to get recommendations for easily editable relaxation rules that can be deployed to allow exceptions.
7. Hybrid security model that applies the power of customizable signatures to block attacks that match specified patterns, and provides the flexibility to use the positive-security-model checks for basic or advanced security protections.
8. Availability of comprehensive configuration reports, including information about PCI-DSS compliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

FAQs and Deployment Guide

Aug 25, 2015

Q: Why is Citrix application firewall the preferred choice for securing applications?

With the following features, the Citrix NetScaler application firewall offers a comprehensive security solution:

- **Hybrid security model:** NetScaler hybrid security model allows you to take advantage of both a positive security model and a negative security model to come up with a configuration ideally suited for your applications.
  - **Positive security model** protects against Buffer Overflow, CGI-BIN Parameter Manipulation, Form/Hidden Field Manipulation, Forceful Browsing, Cookie or Session Poisoning, Broken ACLs, Cross-Site Scripting (XSS), Command Injection, SQL Injection, Error Triggering Sensitive Information Leak, Insecure Use of Cryptography, Server Misconfiguration, Back Doors and Debug Options, Rate-Based Policy Enforcement, Well Known Platform Vulnerabilities, Zero-Day Exploits, Cross Site Request Forgery (CSRF), and leakage of Credit Card and other sensitive data.
  - **Negative security model** uses a rich set signatures to protect against L7 and HTTP application vulnerabilities. The application firewall is integrated with several third party scanning tools, such as those offered by Cenzic, Qualys, Whitehat, and IBM. The built-in XSLT files allow easy importation of rules, which can be used in conjunction with the native-format Snort based rules. An auto-update feature gets the latest updates for new vulnerabilities.

    The positive security model might be the preferred choice for protecting applications that have a high need for security, because it gives you the option to fully control who can access what data. You allow only what you want and block the rest. This model includes a built-in security check configuration, which is deployable with few clicks. However, keep in mind that the tighter the security, the greater the processing overhead.

    The negative security model might be preferable for customized applications. The signatures allow you to combine multiple conditions, and a match and the specified action are triggered only when all the conditions are satisfied. You block only what you don't want and allow the rest. A specific fast-match pattern in a specified location can significantly reduce processing overhead to optimize performance. The option to add your own signature rules, based on the specific security needs of your applications, gives you the flexibility to design your own customized security solutions.

- **Request as well as response side detection and protection:** You can inspect the incoming requests to detect any suspicious behavior and take appropriate actions, and you can check the responses to detect and protect against leakage of sensitive data.
- **Rich set of built-in protections for HTML, XML and JSON payloads:** The application firewall offers 19 different security checks. Six of them (such as Start URL and Deny URL) apply to both HTML and XML data. Five checks (such as Field Consistency and Field Format) are specific to HTML, and eight (such as XML Format and Web Service Interoperability) are specific to XML payloads. This feature includes a rich set of actions and options. For example, URL Closure enables you to control and optimize the navigation through your website, to safeguard against forceful browsing without having to configure relaxation rules to allow each and every legitimate URL. You have the option to remove or x-out the sensitive data, such as credit-card numbers, in the response. Be it SOAP Array attack protection, XML denial of Service (XDoS), WSDL scan prevention, Attachment check, or any number of other XML attacks, you have the comfort of knowing that you have an ironclad shield protecting your data when your applications are protected by the application firewall. The signatures allow you to configure rules using XPATH-Expressions to detect violations in the body as well as the header of a JSON payload.
- **GWT:** Support for protecting Google Web Toolkit applications to safeguard against SQL, XSS and Form Field

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Consistency check violations.

- **Java-free, user friendly graphical user interface (GUI):** An intuitive GUI and preconfigured security checks make it easy to deploy security by clicking a few buttons. A wizard prompts and guides you to create the required elements, such as profiles, policies, signatures, and bindings. The HTML5 based GUI is free of any Java dependency. It's performance is significantly better than that of the older, Java based versions.
- **Easy to Use and automatable CLI:** Most of the configuration options that are available in GUI are also available in the command line interface (CLI). The CLI commands can be executed by a batch file and are easy to automate.
- **Support for REST API:** The NetScaler NITRO protocol supports a rich set of REST API's to automate application firewall configuration and collect pertinent statistics for ongoing monitoring of security violations.
- **Learning:** The application firewall's ability to learn by monitoring traffic to fine tune security is very user friendly. The learning engine recommends rules, which makes it easy to deploy relaxations without proficiency in regular expressions.
- **RegEx editor support:** Regular expression offer an elegant solution to the dilemma of wanting to consolidate rules and yet optimize search. You can capitalize on the power of regular expressions to configure URLs, field names, signature patterns, and so on. The rich built-in GUI RegEx editor offers you a quick reference for the expressions and provides a convenient way to validate and test your RegEx for accuracy.
- **Customized error page:** Blocked requests can be redirected to an error URL. You also have the option to display a customized error object that uses supported variables and Citrix default syntax (advanced PI expressions) to embed troubleshooting information for the client.
- **PCI-DSS, stats, and other violation reports:** The rich set of reports makes it easy to meet the PCI-DSS compliance requirement, gather stats about traffic counters, and view violation reports for all profiles or just one profile.
- **Logging and click-to-rule from log:** Detailed logging is supported for native as well as CEF format. The application firewall offers you the ability to filter targeted log messages in the syslog viewer. You can select a log message and deploy a corresponding relaxation rule by a simple click of a button. You have the flexibility to customize log messages and also have support for generating web logs. For additional details, see http://docs.citrix.com/en-us/netscaler/11/security/application-firewall/logs.html.
- **Include violation logs in trace records:** The ability to include log messages in the trace records makes it very easy to debug unexpected behavior such as reset and block.
- **Cloning:** The useful Import/Export profile option allows you to clone the security configuration from one NetScaler appliance to others. Export learned data options make it easy to export the learned rules to an Excel file. You can then get them reviewed and approved by application owner before applying them.
- **An AppExpert template** (a set of configuration settings) can be designed to provide appropriate protection for your websites. You can simplify and expedite the process of deploying similar protection on other appliances by exporting these cookie-cutter templates to a template. For additional details, see http://support.citrix.com/proddocs/topic/ns-main-appexpert-10-5-map/ns-aapexpert-apptemp-wrapper-con.html
- **Sessionless security checks:** Deploying sessionless security checks can help you reduce the memory footprint and expedite the processing.
- **Interoperability with other NetScaler features:** The application firewall works seamlessly with other NetScaler features, such as rewrite, URL transformation, integrated caching, CVPN, and rate limiting.
- **Support of PI expressions in policies:** You can leverage the power of advanced PI expressions to design policies to implement different levels of security for different parts of your application.
- **Support for IPv6:** The application firewall supports both IPv4 and IPv6 protocols.
- **Geolocation based security protection:** You have the flexibility of using Citrix default syntax (PI Expressions) for configuring location based policies, which can be used in conjunction with a built-in location database to customize firewall protection. You can identify the locations from which malicious requests originate, and enforce the desired level of security-check inspections for requests that originate from a specific geographical location.
- **Performance:** Request-side **streaming** significantly improves performance. As soon as a field is processed, the resulting

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

data is forwarded to the back end while evaluation continues for the remaining fields. The improvement in processing time is especially significant when handling large posts.

- **Other security features:** The application firewall has several other security knobs that can help ensure the security of your data. For example, the **Confidential Field** lets you block leakage of sensitive information in the log messages, and **Strip HTML Comment** allows you to remove the HTML comments from the response before forwarding it to the client. **Field Types** can be used to specify what inputs are allowed in the forms submitted to your application.

Q: What do I need to do to configure Application firewall?

Do the following:

- Add an application firewall profile and select the appropriate type (html, xml, web2.0) for the security requirements of the application.
- Select the required level of security (basic or advanced).
- Add or import the required files, such as signatures or WSDL.
- Configure the profile to use the files, and make any other necessary changes to the default settings.
- Add an application firewall policy for this profile.
- Bind the policy to the target bind point and specify the priority.

Q: How do I know what profile type to choose?

The application firewall profile offers protection for both HTML and XML payloads. Depending on the need of your application, you can choose either a HTML profile or XML profile. If your application supports both HTML and XML data, you can choose a Web2.0 profile.

Q: What is the difference between basic and advanced profiles? How do I decide which one I need?

The decision to use a basic or an advance profile depends on the security need of your application. A basic profile includes a preconfigured set of Start URL and Deny URL relaxation rules. These relaxation rules determine which requests are allowed and which are denied. Incoming requests are matched with the preconfigured rules, and the configured actions are applied. The user can secure applications with minimal configuration of relaxation rules. The Start URL rules protect against forceful browsing. Known web server vulnerabilities that are exploited by hackers can be detected and blocked by enabling a set of default Deny URL rules. Commonly launched attacks, such as Buffer Overflow, SQL, or Cross-Site Scripting can also be easily detected.

As the name indicates, advanced protections are for applications that have higher security requirements. Relaxation rules are configured to allow access to only specific data and block the rest. This positive security model mitigates unknown attacks, which might not be detected by basic security checks. In addition to all the basic protections, an advanced profile keeps track of a user session by controlling the browsing, checking for cookies, specifying input requirements for various form fields, and protecting against tampering of forms or Cross-Site Request Forgery attacks. Learning, which observes the traffic and recommends the appropriate relaxations, is enabled by default for many security checks. Although easy to use, advanced protections require due consideration, because they offer tighter security but also require more processing. Some advance security checks do not allow use of caching, which can affect performance.

Keep the following points in mind when deciding whether to use basic or advanced profiles:

- Basic and advanced profiles are just starting templates. You can always modify the basic profile to deploy advanced security features, and vice versa.
- Advanced security checks require more processing and can affect performance. Unless your application needs advanced security, you might want to start with a basic profile and tighten the security as required for your application.
- You do not want to enable all security checks unless your application needs it.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Q: What is a policy? How do I select the bind point and set the priority?

Application firewall policies can help you sort your traffic into logical groups for configuring different levels of security implementation. Carefully select the bind points for the policies to determine which traffic is matched against which policy. For example, if you want every incoming request to be checked for SQL/XSS attacks, you can create a generic policy and bind it globally. Or, if you want to apply more stringent security checks to the traffic of a virtual server hosting applications that contain sensitive data, you can bind a policy to that virtual server.

Careful assignment of priorities can enhance the traffic processing. You want to assign higher priorities to more specific policies and lower priorities to generic policies. Note that the higher the number, the lower the priority. A policy with a priority of 10 is evaluated before a policy that has a priority of 15.

You can apply different levels of security for different kinds of contents, e.g. requests for static objects like images and text can be by-passed by using one policy and requests for other sensitive contents can be subjected to a much stringent check by using a second policy.

Q: How do I go about configuring the rules to secure my application?

The application firewall makes it very easy to design the right level of security for your web-site. You can have multiple application firewall policies, bound to different application firewall profiles, to implement different levels of security-check inspections for your applications. You can initially monitor the logs to observe what security threats are being detected and which violations are being triggered. You can either manually add the relaxation rules or take advantage of the application firewall's recommended learned rules to deploy the required relaxations to avoid false positives.

The Citrix application firewall offers **visualizer** support in GUI, which makes rule management very easy. You can easily view all the data on one screen, and take action on several rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate several rules. You can select a subset of the rules, basing your selection on the delimiter and Action URL. Visualizer support is available for viewing 1) learned rules and 2) relaxation rules.

> 1) The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. You can also skip (ignore) rules.
>
> 2) The visualizer for deployed relaxations offers you the option to add a new rule or edit an existing one. You can also enable or disable a group of rules by selecting a node and clicking the **Enable** or **Disable** button in the relaxation visualizer.

Q: What are signatures? How do I know which signatures to use?

A signature is an object that can have multiple rules. Each rule consists of one or more patterns that can be associated with a specified set of actions. The application firewall has a built-in default signature object consisting of more than 1,300 signature rules, with an option to get the latest rules by using the **auto-update** feature to get protection against new vulnerabilities. Rules created by other scan tools can also be imported.

Signatures are very powerful because they use pattern matching to detect malicious attacks and can be configured to check both the request and the response of a transaction. They are a preferred option when a customizable security solution is needed. Multiple action choices (for example, block, log, learn, and transform) are available for when a signature match is detected. The default signatures cover rules to protect different types of applications, such as web-cgi, web-coldfusion, web-frontpage, web-iis, web-php, web-client, web-activex, web-shell-shock, and web-struts. To match the needs of your application, you can select and deploy the rules belonging to a specific category.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Signature-usage tips:

- You can just make a copy of the default signature object and modify it to enable the rules you need and configure the actions you want.
- The signature object can be customized by adding new rules, which can work in conjunction with other signature rules.
- The signature rules can also be configured to work in conjunction with the security checks specified in the application firewall profile. If a match indicating a violation is detected by a signature as well as a security check, the more restrictive action is the one that gets enforced.
- A signature rule can have multiple patterns and be configured to flag a violation only when all the patterns are matched, thereby avoiding false positives.
- Careful selection of a literal fast-match pattern for a rule can significantly optimize processing time.

Q: Does the application firewall work with other NetScaler features?

The application firewall is fully integrated into the NetScaler appliance and works seamlessly with other features. You can configure maximum security for your application by using other NetScaler security features in conjunction with the application firewall. For example, **AAA-TM** can be used to authenticate the user, check the user's authorization to access the content, and log the accesses, including invalid login attempts. **Rewrite** can be used to modify the URL or to add, modify or delete headers, and **Responder** can be used to deliver customized content to different users. You can define the maximum load for your website by using **Rate Limiting** to monitor the traffic and throttle the rate if it is too high. **HTTP Denial-of-Service (DoS)** protection can help distinguish between real HTTP clients and malicious DoS clients. You can narrow the scope of security-check inspection by binding the application firewall policies to virtual servers, while still optimizing the user experience by using the **Load Balancing** feature to manage heavily used applications. Requests for static objects such as images or text can bypass security check inspection, taking advantage of **integrated caching** or **compression** to optimize the bandwidth usage for such content.

Q: How is the payload processed by the application firewall and the other NetScaler features?

A diagram showing details of the L7 packet flow in a NetScaler appliance is available in the Processing Order of Features section at http://docs.citrix.com/en-us/netscaler/11/getting-started-with-netscaler.html.

Q: What is the recommended workflow for application firewall deployment?

Now that you know the advantages of using the state-of-the-art security protections of the Citrix application firewall, you might want to collect additional information that can help you design the optimal solution for your security needs. Citrix recommends that you do the following:

- **Know your environment:** Knowing your environment will help you to identify the best security protection solution (signatures, security checks, or both) for your needs. Before you begin configuration, you should gather the following information.
  - **OS:** What kind of OS (MS Windows, Linux, BSD, Unix, others) do you have?
  - **Web Sever:** What web server (IIS, Apache or NetScaler Enterprise Server) are you running?
  - **Application:** What type of applications are running on your application server (for example, ASP.NET, PHP, Cold Fusion, ActiveX, FrontPage, Struts, CGI, Apache Tomcat, Domino, and WebLogic)?
  - Do you have customized applications or off-the-shelf (for example, Oracle, SAP) applications? What version you are using?
  - **SSL:** Do you require SSL? If so, what key size (512, 1024, 2048, 4096) is used for signing certificates?
  - **Traffic Volume:** What is the average traffic rate through your applications? Do you have seasonal or time-specific spikes in the traffic?
  - **Server Farm:** How many servers do you have? Do you need to use load balancing?

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Database:** What type of database (MS-SQL, MySQL, Oracle, Postgres, SQLite, nosql, Sybase, Informix etc.) do you use?
- **DB Connectivity:** What kind of data base connectivity do you have (DSN, per-file connection string, single file connection string) and what drivers are used?

- **Identify your security needs:** You might want to evaluate which applications or specific data need maximum security protection, which ones are less vulnerable, and the ones for which security inspection can safely be bypassed. This will help you in coming up with an optimal configuration, and in designing appropriate policies and bind points to segregate the traffic. For example, you might want to configure a policy to bypass security inspection of requests for static web content, such as images, MP3 files, and movies, and configure another policy to apply advanced security checks to requests for dynamic content. You can use multiple policies and profiles to protect different contents of the same application.
- **License requirement:** Citrix offers a unified solution to optimize the performance of your application by taking advantage of a rich set of features such as load balancing, content switching, caching, compression, responder, rewrite, and content filtering, to name a few. Identifying the features that you want can help you decide which license you need.
- **Install and baseline a NetScaler appliance:** Create a virtual server and run test traffic through it to get an idea of the rate and amount of traffic flowing through your system. This information will help you to identify your capacity requirement and select the right appliance (VPX, MPX, or SDX). For a detailed description of various available platforms and their throughput capabilities, see the following data sheet:

  https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/netscaler-data-sheet.pdf?accessmode=direct

- **Deploy the application firewall:** Use the application firewall wizard to proceed with a simple security configuration. The wizard walks you through several screens and prompts you to add a profile, policy, signature, and security checks.
  - **Profile:** Select a meaningful name and the appropriate type (HTML, XML or WEB 2.0) for your profile. The policy and signatures will be auto-generated using the same name.
  - **Policy:** The auto-generated policy has the default Expression (true), which selects all traffic and is bound globally. This is a good starting point unless you have in mind a specific policy that you want to use.
  - **Protections:** The wizard helps you take advantage of the hybrid security model, in which you can use the default signatures offering a rich set of rules to protect different types of applications. **Simple** edit mode allows you to view the various categories (CGI, Cold Fusion, PHP, etc.). You can select one or more categories to identify a specific set of rules applicable to your application. Use the **Action** option to enable all the signature rules in the selected categories. Make sure that blocking is disabled, so that you can monitor the traffic before tightening the security. Click **Continue**. In the **Specify Deep protections** pane, you can make changes as needed to deploy the security check protections. In most cases, basic protections are sufficient for initial security configuration. Run the traffic for a while to collect a representative sample of the security-inspection data.
  - **Tightening the security:** After deploying application firewall and observing the traffic for a while, you can start tightening the security of your applications by deploying relaxations and then enabling blocking. **Learning**, **Visualizer**, and **Click to deploy rules** are useful features that make it very easy to tweak your configuration to come up with just the right level of relaxation. At this point, you can also change the policy expression and/or configure additional policies and profiles to implement desired levels of security for different types of content.
  - **Debugging:** If you see unexpected behavior of your application, the application firewall offers various options for easy debugging:
    - **Log**. If legitimate requests are getting blocked, your first step is to check the ns.log file to see if any unexpected security-check violation is being triggered.
    - **Disable feature.** If you do not see any violations but are still seeing unexpected behavior, such as an application

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

resetting or sending partial responses, you can disable the application firewall feature for debugging. If the issue persists, it rules out the application firewall as a suspect.

- **Trace records with log messages.** If the issue appears to be application firewall related and needs closer inspection, you have the option to include security violation messages in an nstrace. You can use “Follow TCP stream” in the trace to view the details of the individual transaction, including headers, payload, and the corresponding log message, together on the same screen. Details of how to use this functionality are available at http://docs.citrix.com/en-us/netscaler/11/security/application-firewall/appendixes/nstrace-with-violation-logs.html.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Introduction

Oct 08, 2014

The Citrix NetScaler Application Firewall prevents security breaches, data loss, and possible unauthorized modifications to web sites that access sensitive business or customer information. It does so by filtering both requests and responses, examining them for evidence of malicious activity, and blocking those that exhibit such activity. Your site is protected not only from common types of attacks, but also from new, as yet unknown attacks. In addition to protecting web servers and web sites from unauthorized access and misuse by hackers and malicious programs, the application firewall provides protection against security vulnerabilities in legacy CGI code or scripts, other web frameworks, web server software, and the underlying operating systems.

The NetScaler Application Firewall is available as a stand-alone appliance, or as a feature on a Citrix NetScaler application delivery controller (ADC) or Citrix NetScaler virtual appliance (VPX). In the application firewall documentation, the term NetScaler ADC refers to the platform on which the application firewall is running, regardless of whether that platform is a dedicated firewall appliance, a NetScaler ADC on which other features have also been configured, or a NetScaler VPX.

To use the application firewall, you must create at least one security configuration to block connections that violate the rules that you set for your protected web sites. The number of security configurations that you might want to create depends on the complexity of your web site. In some cases, a single configuration is sufficient. In other cases, particularly those that include interactive web sites, web sites that access database servers, online stores with shopping carts, you might need several different configurations to best protect sensitive data without wasting significant effort on content that is not vulnerable to certain types of attacks. You can often leave the defaults for the global settings, which affect all security configurations, unchanged. However, you can change the global settings if they conflict with other parts of your configuration or you prefer to customize them.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Web Application Security

Oct 08, 2014

Web application security is network security for computers and programs that communicate by using the HTTP and HTTPS protocols. This is an extremely broad area in which security flaws and weaknesses abound. Operating systems on both servers and clients have security issues and are vulnerable to attack. Web server software and web site enabling technologies such as CGI, Java, JavaScript, PERL and PHP have underlying vulnerabilities. Browsers and other client applications that communicate with web-enabled applications also have vulnerabilities. Web sites that use any technology but the simplest of HTML, including any site that allows interaction with visitors, often have vulnerabilities of their own.

In the past, a breach in security was often just an annoyance, but today that is seldom the case. For example, attacks in which a hacker gained access to a web server and made unauthorized modifications to (defaced) a web site used to be common. They were usually launched by hackers who had no motivation beyond demonstrating their skills to fellow hackers or embarrassing the targeted person or company. Most current security breaches, however, are motivated by a desire for money. The majority attempt to accomplish one or both of the following goals: to obtain sensitive and potentially valuable private information, or to obtain unauthorized access to and control of a web site or web server.

Certain forms of web attacks focus on obtaining private information. These attacks are often possible even against web sites that are secure enough to prevent an attacker from taking full control. The information that an attacker can obtain from a web site can include customer names, addresses, phone numbers, social security numbers, credit card numbers, medical records, and other private information. The attacker can then use this information or sell it to others. Much of the information obtained by such attacks is protected by law, and all of it by custom and expectation. A breach of this type can have extremely serious consequences for customers whose private information is compromised. At best, these customers will have to exercise vigilance to prevent others from abusing their credit cards, opening unauthorized credit accounts in their name, or appropriating their identities outright (identity theft). At worst, the customers may face ruined credit ratings or even be blamed for criminal activities in which they had no part.

Other web attacks are aimed at obtaining control of (or *compromising*) a web site or the server on which it operates, or both. A hacker who gains control of a web site or server can use it to host unauthorized content, act as a proxy for content hosted on another web server, provide SMTP services to send unsolicited bulk email, or provide DNS services to support such activities on other compromised web servers. Most web sites that are hosted on compromised web servers promote questionable or outright fraudulent businesses. For example, the majority of phishing web sites and child exploitation web sites are hosted on compromised web servers.

Protecting your web sites and web services against these attacks requires a multilayered defense capable of both blocking known attacks with identifiable characteristics and protecting against unknown attacks, which can often be detected because they look different from the normal traffic to your web sites and web services.

Known Web Attacks

Updated: 2014-10-08

The first line of defense for your web sites is protection against the large number of attacks that are known to exist and have been observed and analyzed by web security experts. Common types of attacks against HTML-based web sites include:

- **Buffer overflow attacks.** Sending an extremely long URL, extremely long cookie, or other extremely long bit of information to a web server in hopes of causing it or the underlying operating system to hang, crash, or provide the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attacker with access to the underlying operating system. A buffer overflow attack can be used to gain access to unauthorized information, to compromise a web server, or both.

- **Cookie security attacks.** Sending a modified cookie to a web server, usually in hopes of obtaining access to unauthorized content by using falsified credentials.
- **Forceful browsing.** Accessing URLs on a web site directly, without navigating to the URLs by means of hyperlinks on the home page or other common start URLs on the web site. Individual instances of forceful browsing may simply indicate a user who bookmarked a page on your web site, but repeated attempts to access nonexistent content, or content that users should never access directly, often represent an attack on web site security. Forceful browsing is normally used to gain access to unauthorized information, but can also be combined with a buffer overflow attack in an attempt to compromise your server.
- **Web form security attacks.** Sending inappropriate content to your web site in a web form. Inappropriate content can include modified hidden fields, HTML or code in a field intended for alphanumeric data only, an overly long string in a field that accepts only a short string, an alphanumeric string in a field that accepts only an integer, and a wide variety of other data that your web site does not expect to receive in that web form. A web form security attack can be used either to obtain unauthorized information from your web site or to compromise the web site outright, usually when combined with a buffer overflow attack.

Two specialized types of attacks on web form security deserve special mention:

- **SQL injection attacks.** Sending an active SQL command or commands in a web form or as part of a URL, with the goal of causing an SQL database to execute the command or commands. SQL injection attacks are normally used to obtain unauthorized information.
- **Cross-site scripting attacks.** Using a URL or a script on a web page to violate the same-origin policy, which forbids any script from obtaining properties from or modifying any content on a different web site. Since scripts can obtain information and modify files on your web site, allowing a script access to content on a different web site can provide an attacker the means to obtain unauthorized information, to compromise a web server, or both.

Attacks against XML-based web services normally fall into at least one of the following two categories: attempts to send inappropriate content to a web service, or attempts to breach security on a web service. Common types of attacks against XML-based web services include:

- **Malicious code or objects.** XML requests that contain code or objects that can either directly obtain sensitive information or can give an attacker control of the web service or underlying server.
- **Badly-formed XML requests.** XML requests that do not conform to the W3C XML specification, and that can therefore breach security on an insecure web service.
- **Denial of service (DoS) attacks.** XML requests that are sent repeatedly and in high volume, with the intent of overwhelming the targeted web service and denying legitimate users access to the web service.

In addition to standard XML-based attacks, XML web services and Web 2.0 sites are also vulnerable to SQL injection and cross-site scripting attacks, as described below:

- **SQL injection attacks.** Sending an active SQL command or commands in an XML-based request, with the goal of causing an SQL database to execute that command or commands. As with HTML SQL injection attacks, XML SQL injection attacks are normally used to obtain unauthorized information.
- **Cross-site scripting attacks.** Using a script included in an XML based application to violate the same-origin policy, which does not allow any script to obtain properties from or modify any content on a different application. Since scripts can obtain information and modify files by using your XML application, allowing a script access to content belonging to a different application can give an attacker the means to obtain unauthorized information, to compromise the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

application, or both.

Known web attacks can usually be stopped by filtering web site traffic for specific characteristics (signatures) that always appear for a specific attack and should never appear in legitimate traffic. This approach has the advantages of requiring relatively few resources and posing relatively little risk of false positives. It is therefore a valuable tool in fighting attacks on web sites and web services, and configuring basic signature protections that intercept most known web attacks is easy to do.

Unknown Web Attacks

The greatest threat against web sites and applications does not come from known attacks, but from unknown attacks. Most unknown attacks fall into one of two categories: newly-launched attacks for which security firms have not yet developed an effective defense (zero-day attacks), and carefully-targeted attacks on a specific web site or web service rather than many web sites or web services (spear attacks). These attacks, like known attacks, are usually intended to obtain sensitive private information, compromise the web site or web service and allow it to be used for further attacks, or both of those goals.

Zero-day attacks are a major threat to all users. These attacks are usually of the same types as known attacks; zero-day attacks often involve injected SQL, a cross-site script, a cross-site request forgery, or another type of attack similar to known attacks. In most cases, they target vulnerabilities that the developers of the targeted software, web site, or web service either are unaware of or have just learned about. Security firms have therefore usually not developed defenses against these attacks, and even if they have, users have usually not obtained and installed the patches or performed the workarounds necessary to protect against these attacks. The time between discovery of a zero-day attack and availability of a defense (the vulnerability window) is shrinking, but perpetrators can still count on hours or even days in which many web sites and web services lack any specific protection against the attack.

Spear attacks are a major threat, but to a more select group of users. A common type of spear attack, a spear phish, is usually targeted at customers of a specific bank or financial institution, or (less commonly) at employees of a specific company or organization. Unlike other phishes, which are often crudely written forgeries that a user with any familiarity with the actual communications of that bank or financial institution can recognize, spear phishes are letter perfect and extremely convincing. They can contain information specific to the individual that, at first look, no stranger should know or be able to obtain. The spear phisher is therefore able to convince his or her target to provide the requested information, which the phisher can then use to loot accounts, to process illegitimately obtained money from other sources, or to gain access to other, even more sensitive information.

Both of these types of attack have certain characteristics that can usually be detected, although not by using static patterns that look for specific characteristics, as do standard signatures. Detecting these types of attacks requires more sophisticated and more resource-intensive approaches, such as heuristic filtering and positive security model systems. Heuristic filtering looks, not for specific patterns, but for patterns of behaviors. Positive security model systems model the normal behavior of the web site or web service that they are protecting, and then block connections that do not fit within that model of normal use. URL based and web-form based security checks profile normal use of your web sites, and then control how users interact with your web sites, using both heuristics and positive security to block anomalous or unexpected traffic. Both heuristic and positive security, properly designed and deployed, can catch most attacks that signatures miss. However, they require considerably more resources than do signatures, and you must spend some time configuring them properly to avoid false positives. They are therefore usually used, not as the primary line of defense, but as backups to signatures or other less resource-intensive approaches.

By configuring these advanced protections in addition to signatures, you create a hybrid security model, which enables the application firewall to provide comprehensive protection against both known and unknown attacks.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

How The Application Firewall Works

Sep 03, 2013

When you install the application firewall, you create an initial security configuration, which consists of a policy, a profile, and a signatures object. The policy is a rule that identifies the traffic to be filtered, and the profile identifies the patterns and types of behavior to allow or block when the traffic is filtered. The simplest patterns, which are called signatures, are not specified within the profile, but in a signatures object that is associated with the profile.

A signature is a string or pattern that matches a known type of attack. The application firewall contains over a thousand signatures in seven categories, each directed at attacks on specific types of web servers and web content. Citrix updates the list with new signatures as new threats are identified. During configuration, you specify the signature categories that are appropriate for the web servers and content that you need to protect. Signatures provide good basic protection with low processing overhead. If your applications have special vulnerabilities or you detect an attack against them for which no signature exists, you can add your own signatures.

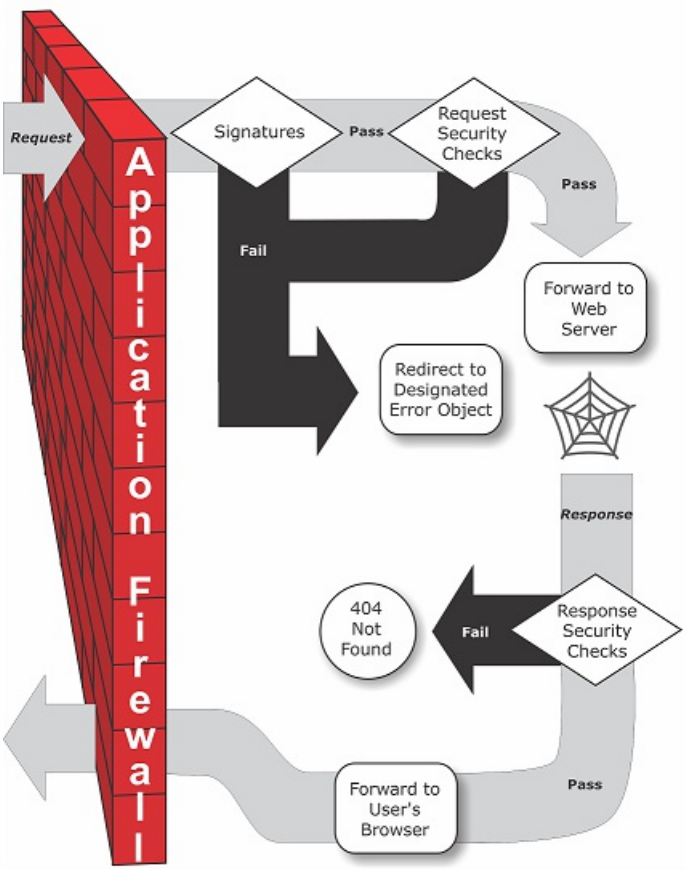
The more advanced protections are called security checks. A security check is a more rigorous, algorithmic inspection of a request for specific patterns or types of behavior that might indicate an attack or constitute a threat to your protected web sites and web services. It can, for example, identify a request that attempts to perform a certain type of operation that might breach security, or a response that includes sensitive private information such as a social security number or credit card number. During configuration, you specify the security checks that are appropriate for the web servers and content that you need to protect. The security checks are restrictive. Many of them can block legitimate requests and responses if you do not add the appropriate exceptions (relaxations) when configuring them. Identifying the needed exceptions is not difficult if you use the adaptive learning feature, which observes normal use of your web site and creates recommended exceptions.

The application firewall can be installed as either a Layer 3 network device or a Layer 2 network bridge between your servers and your users, usually behind your company's router or firewall. It must be installed in a location where it can intercept traffic between the web servers that you want to protect and the hub or switch through which users access those web servers. You then configure the network to send requests to the application firewall instead of directly to your web servers, and responses to the application firewall instead of directly to your users. The application firewall filters that traffic before forwarding it to its final destination, using both its internal rule set and your additions and modifications. It blocks or renders harmless any activity that it detects as harmful, and then forwards the remaining traffic to the web server. The following figure provides an overview of the filtering process.

Note: The figure omits the application of a policy to incoming traffic. It illustrates a security configuration in which the policy is to process all requests. Also, in this configuration, a signatures object has been configured and associated with the profile, and security checks have been configured in the profile.

Figure 1. A Flowchart of Application Firewall Filtering

© 1999-2017 Citrix Systems, Inc. All rights reserved.



As the figure shows, when a user requests a URL on a protected web site, the application firewall first examines the request to ensure that it does not match a signature. If the request matches a signature, the application firewall either displays the error object (a web page that is located on the application firewall appliance and which you can configure by using the imports feature) or forwards the request to the designated error URL (the error page). Signatures do not require as many resources as do security checks, so detecting and stopping attacks that are detected by a signature before running any of the security checks reduces the load on the server.

If a request passes signature inspection, the application firewall applies the request security checks that have been enabled. The request security checks verify that the request is appropriate for your web site or web service and does not contain material that might pose a threat. For example, security checks examine the request for signs indicating that it might be of an unexpected type, request unexpected content, or contain unexpected and possibly malicious web form data, SQL commands, or scripts. If the request fails a security check, the application firewall either sanitizes the request and then sends it back to the NetScaler appliance (or NetScaler virtual appliance), or displays the error object. If the request passes the security checks, it is sent back to the NetScaler appliance, which completes any other processing and forwards the request to the protected web server.

When the web site or web service sends a response to the user, the application firewall applies the response security checks that have been enabled. The response security checks examine the response for leaks of sensitive private information, signs of web site defacement, or other content that should not be present. If the response fails a security check, the application firewall either removes the content that should not be present or blocks the response. If the response passes the security checks, it is sent back to the NetScaler appliance, which forwards it to the user.

Application Firewall Features

Updated: 2013-09-03

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The basic application firewall features are policies, profiles, and signatures, which provide a hybrid security model as described in "Known Web Attacks," "Unknown Web Attacks," and "How the Application Firewall Works." Of special note is the learning feature, which observes traffic to your protected applications and recommends appropriate configuration settings for certain security checks.

The imports feature manages files that you upload to the application firewall. These files are then used by the application firewall in various security checks, or when responding to a connection that matches a security check.

You can use the logs, statistics, and reports features to evaluate the performance of the application firewall and identify possible needs for additional protections.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

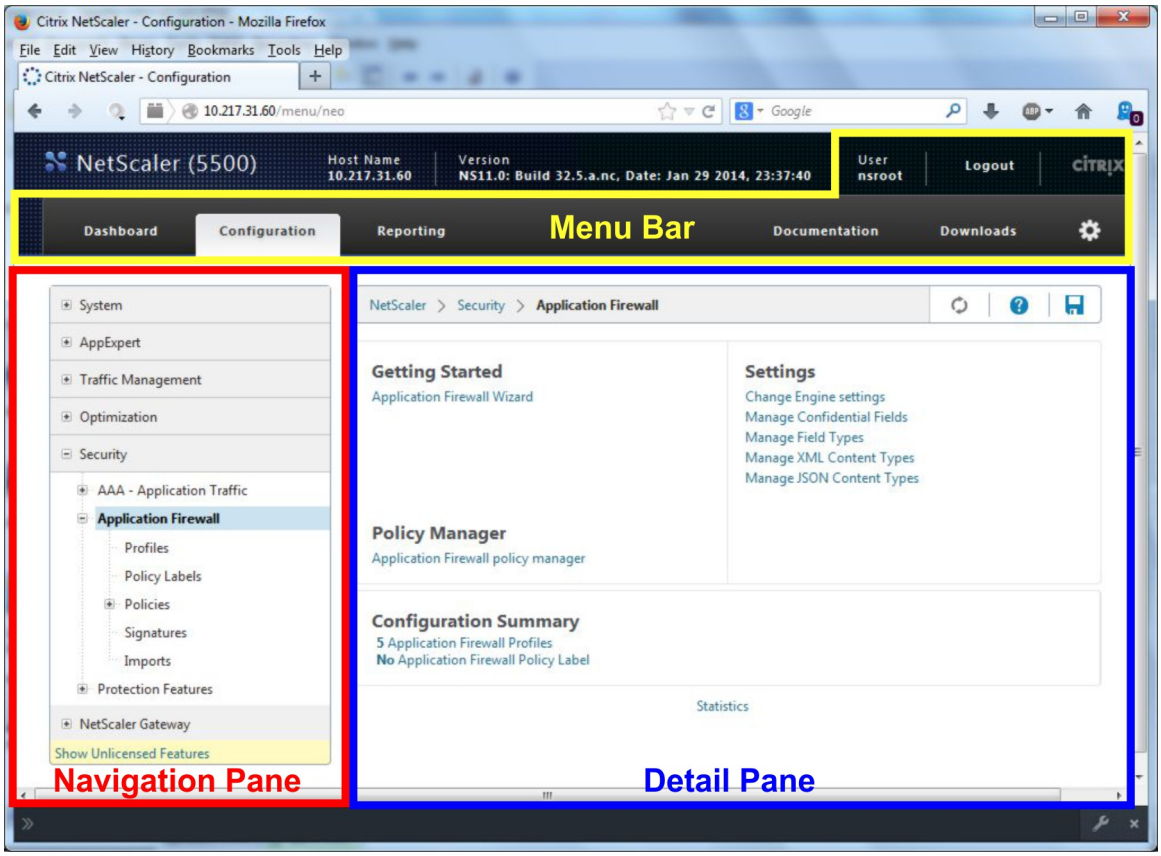
The Application Firewall Configuration Interfaces

Oct 09, 2014

All hardware and virtual versions of the Citrix NetScaler application delivery controller (ADC) can be configured and managed from the Citrix NetScaler command line interface or the web-based configuration utility. All features of most NetScaler features can be configured using either of these tools. The Citrix Application Firewall is an exception: not all application firewall configuration tasks can be performed at the command line. Inexperienced users also find the configuration utility easier to use. In particular, the application firewall wizard considerably reduces the complexity of configuring the application firewall. Unlike most NetScaler wizards, the application firewall wizard can serve as your primary interface to the application firewall.

The command line interface is a modified UNIX shell based on the FreeBSD bash shell. To configure the application firewall from the command line interface, you type commands at the prompt and press the Enter key, just as you do with any other Unix shell. For instructions for using the command line interface, see "Command Reference."

The configuration utility is a web-based GUI interface to the ADC. The application firewall configuration section is found under Security > Application Firewall. Figure 1 shows the navigation pane expanded to display the application firewall screens, and in the detail pane the main application firewall screen.



The configuration utility has two main areas on all screens. The panel on the left, called the navigation pane, contains a navigation tree, with which you navigate to the screens on which you configure the features that are installed on your appliance. The screens to which you navigate appear to the right of the navigation pane, in the details pane.

When you access the configuration utility, the details pane displays the System Overview screen. If, in the navigation pane,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

you click plus sign next to the application firewall folder, the Application Firewall node expands to include the main application firewall elements that you can configure. If you click the first element, Profiles, the details pane displays the configured profiles, if any profiles have been configured. At the bottom of the details pane, you can click Add to configure a new profile. Other buttons at the bottom of the details pane are grayed out until you select an existing profile. Screens for the other elements work in the same way.

If, instead of expanding the application firewall node, you click the node itself, the details pane displays different options, one of which is the application firewall wizard, as shown in Figure 1. Citrix recommends that you use the wizard for initial configuration, and many users use it almost exclusively. It includes most of the functionality that is available elsewhere in the configuration utility.

For information and instructions on accessing the configuration utility, see "Citrix NetScaler Getting Started Guide."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring the Application Firewall

Oct 09, 2014

You can configure the Citrix Application Firewall (application firewall) by using any of the following methods:

- **Application Firewall Wizard.** A dialog box consisting of a series of screens that step you through the configuration process.
- **Citrix Web Interface AppExpert Template.** A NetScaler AppExpert template (a set of configuration settings) that are designed to provide appropriate protection for web sites. This AppExpert template contains appropriate Application Firewall configuration settings for protecting many web sites.
- **Citrix NetScaler Configuration Utility.** The NetScaler web-based configuration interface.
- **Citrix NetScaler Command Line Interface.** The NetScaler command line configuration interface.

Citrix recommends that you use the Application Firewall Wizard. Most users will find it the easiest method to configure the application firewall, and it is designed to prevent mistakes. If you have a new Citrix NetScaler ADC or VPX that you will use primarily to protect web sites, you may find the Web Interface AppExpert template a better option because it provides a good default configuration, not just for the application firewall, but for the entire appliance. Both the configuration utility and the command line interface are intended for experienced users, primarily to modify an existing configuration or use advanced options.

The Application Firewall Wizard

The application firewall wizard is a dialog box that consists of several screens that prompt you to configure each part of a simple configuration. The application firewall then creates the appropriate configuration elements from the information that you give it. This is the simplest and, for most purposes, the best way to configure the application firewall.

To use the wizard, connect to the configuration utility with the browser of your choice. When the connection is established, verify that the application firewall is enabled, and then run the application firewall wizard, which prompts you for configuration information. You do not have to provide all of the requested information the first time you use the wizard. Instead, you can accept default settings, perform a few relatively straightforward configuration tasks to enable important features, and then allow the application firewall to collect important information to help you complete the configuration.

For example, when the wizard prompts you to specify a rule for selecting the traffic to be processed, you can accept the default, which selects all traffic. When it presents you with a list of signatures, you can enable the appropriate categories of signatures and turn on the collection of statistics for those signatures. For this initial configuration, you can skip the advanced protections (security checks). The wizard automatically creates the appropriate policy, signatures object, and profile (collectively, the security configuration) , and binds the policy to global. The application firewall then begins filtering connections to your protected web sites, logging any connections that match one or more of the signatures that you enabled, and collecting statistics about the connections that each signature matches. After the application firewall processes some traffic, you can run the wizard again and examine the logs and statistics to see if any of the signatures that you have enabled are matching legitimate traffic. After determining which signatures are identifying the traffic that you want to block, you can enable blocking for those signatures. If your web site or web service is not complex, does not use SQL, and does not have access to sensitive private information, this basic security configuration will probably provide adequate protection.

You may need additional protection if, for example, your web site is dynamic. Content that uses scripts may need protection against cross-site scripting attacks. Web content that uses SQL—such as shopping carts, many blogs, and most

© 1999-2017 Citrix Systems, Inc. All rights reserved.

content management systems—may need protection against SQL injection attacks. Web sites and web services that collect sensitive private information such as social security numbers or credit card numbers may require protection against unintentional exposure of that information. Certain types of web-server or XML-server software may require protection from types of attacks tailored to that software. Another consideration is that specific elements of your web sites or web services may require different protection than do other elements. Examining the application firewall logs and statistics can help you identify the additional protections that you might need.

After deciding which advanced protections are needed for your web sites and web services, you can run the wizard again to configure those protections. Certain security checks require that you enter exceptions (relaxations) to prevent the check from blocking legitimate traffic. You can do so manually, but it is usually easier to enable the adaptive learning feature and allow it to recommend the necessary relaxations. You can use the wizard as many times as necessary to enhance your basic security configuration and/or create additional security configurations.

The wizard automates some tasks that you would have to perform manually if you did not use the wizard. It automatically creates a policy, a signatures object, and a profile, and assigns them the name that you provided when you were prompted for the name of your configuration. The wizard also adds your advanced-protection settings to the profile, binds the signatures object to the profile, associates the profile with the policy, and puts the policy into effect by binding it to Global.

A few tasks cannot be performed in the wizard. You cannot use the wizard to bind a policy to a bind point other than Global. If you want the profile to apply to only a specific part of your configuration, you must manually configure the binding. You cannot configure the engine settings or certain other global configuration options in the wizard. While you can configure any of the advanced protection settings in the wizard, if you want to modify a specific setting in a single security check, it may be easier to do so on the manual configuration screens in the configuration utility.

For more information on using the Application Firewall Wizard, see "The Application Firewall Wizard."

The Citrix Web Interface AppExpert Template

AppExpert Templates are a different and simpler approach to configuring and managing complex enterprise applications. The AppExpert display in the configuration utility consists of a table. Applications are listed in the left-most column, with the NetScaler features that are applicable to that application appearing each in its own column to the right. (In the AppExpert interface, those features that are associated with an application are called *application units*.) In the AppExpert interface, you configure the interesting traffic for each application, and turn on rules for compression, caching, rewrite, filtering, responder and the application firewall, instead of having to configure each feature individually.

The Web Interface AppExpert Template contains rules for the following application firewall signatures and security checks:

- "**Deny URL check**." Detects connections to content that is known to pose a security risk, or to any other URLs that you designate.
- "**Buffer Overflow check**." Detects attempts to cause a buffer overflow on a protected web server.
- "**Cookie Consistency check**." Detects malicious modifications to cookies set by a protected web site.
- "**Form Field Consistency check**." Detects modifications to the structure of a web form on a protected web site.
- "**CSRF Form Tagging check**." Detects cross-site request forgery attacks.
- "**Field Formats check**." Detects inappropriate information uploaded in web forms on a protected web site.
- "**HTML SQL Injection check**." Detects attempts to inject unauthorized SQL code.
- "**HTML Cross-Site Scripting check**." Detects cross-site scripting attacks.

For information on installing and using an AppExpert Template, see "AppExpert Applications and Templates."

The Citrix NetScaler Configuration Utility

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The NetScaler configuration utility is a web-based interface that provides access to all configuration options for the application firewall feature, including advanced configuration and management options that are not available from any other configuration tool or interface. Specifically, many advanced Signatures options can be configured only in the configuration utility. You can review recommendations generated by the learning feature only in the configuration utility. You can bind policies to a bind point other than Global only in the configuration utility.

For a description of the configuration utility, see "The Application Firewall Configuration Interfaces." For more information on using the configuration utility to configure the application firewall, see "Manual Configuration By Using the Configuration Utility."

For instructions on configuring the application firewall by using the configuration utility, see "Manual Configuration By Using the Configuration Utility." For information on the Citrix NetScaler Configuration Utility, see "The Application Firewall Configuration Interfaces."

The Citrix NetScaler Command Line Interface

The Citrix NetScaler command line interface is a modified UNIX shell based on the FreeBSD bash shell. To configure the Application Firewall from the command line interface, you type commands at the prompt and press the Enter key, just as you do with any other Unix shell. You can configure most parameters and options for the application firewall by using the NetScaler command line. Exceptions are the signatures feature, many of whose options can be configured only by using the configuration utility or the Application Firewall wizard, and the learning feature, whose recommendations can only be reviewed in the configuration utility.

For instructions on configuring the application firewall by using the NetScaler command line, see "Manual Configuration By Using the Command Line Interface."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling the Application Firewall

Oct 09, 2014

Before you can create an application firewall security configuration, you must make sure that the application firewall feature is enabled.

- If you are configuring a dedicated Citrix Application Firewall ADC or are upgrading an existing Citrix NetScaler ADC or VPX, the feature is already enabled. You do not have to perform either of the procedures described here.
- If you have a new NetScaler ADC or VPX, you need to enable the application firewall feature before you configure it.
- If you are upgrading a NetScaler ADC or VPX from a previous version of the NetScaler operating system to the current version, you might need to enable the application firewall feature before you configure it.

Note: If you are upgrading a NetScaler ADC or VPX from a previous version, you might also need to update the licenses on your ADC or VPX before you can enable the application firewall. Check with your Citrix representative or reseller to obtain the correct license.

You can enable the application firewall by using the command line or the configuration utility.

To enable the application firewall by using the command line interface

At the command prompt, type the following command:

```
enable ns feature AppFW
```

To enable the application firewall by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, click Configure Basic Features.
3. In the Configure Basic Features dialog box, check the Application Firewall check box.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The Application Firewall Wizard

Oct 01, 2013

Unlike most wizards, the Application Firewall wizard is designed not just to simplify the initial configuration process, but also to modify previously created configurations and to maintain your Application Firewall setup. A typical user runs the wizard multiple times, skipping some of the screens each time.

Opening the Wizard

To run the Application Firewall wizard, first open the configuration utility. Next, in the navigation pane, expand Application Firewall, and then in the details pane click Application Firewall Wizard. (For more information about the configuration utility, see "The Application Firewall User Interfaces.") Then:

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard. The first screen of the wizard appears.
3. To advance to the next screen, click Next.

The Wizard Screens

The Application Firewall wizard displays the following screens, in the following order:

1. **Introduction screen.** Provides an introduction to the Application Firewall wizard. There is nothing that you can configure on this screen.

2. **Specify Name screen.** On this screen, when creating a new security configuration, you specify the name that the wizard is to assign to the configuration. The name can begin with a letter, number, or the underscore symbol, and can consist of from 1 to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols. Choose a name that makes it easy for others to tell what content your new security configuration protects.

   Note: Because the wizard uses this name for both the policy and the profile, it is limited to 31 characters. Manually created policies can have names up to 127 characters in length.
   When creating an existing configuration, you select Modify Existing Configuration and then, in the Name drop-down list, select the name of the existing configuration that you want to modify.

   Note: Only policies that are bound to global or to a bind point appear in this list; you cannot modify an unbound policy by using the Application Firewall wizard. You must either manually bind it to Global or a bind point, or modify it manually. (For manual modification, in the configuration utility's Application Firewall --> Policies --> Firewall pane, select the policy and click Open).
   You also select a profile type on this screen. The profile type determines the types of advanced protection (security checks) that can be configured. Because certain kinds of content are not vulnerable to certain types of security threats, restricting the list of available checks saves time during configuration. The types of Application Firewall profiles are:

   - **Web Application (HTML).** Any HTML-based Web site that does not use XML or Web 2.0 technologies.
   - **XML Application (XML, SOAP).** Any XML-based Web service.
   - **Web 2.0 Application (HTML, XML, REST).** Any Web 2.0 site that combines HTML and XML-based content, such as an ATOM-based site, a blog, an RSS feed, or a wiki.

   Note: If you are unsure which type of content is used on your Web site, you can choose Web 2.0 Application to ensure that you protect all types of Web application content.
3. **Specify Rule screen.** On this screen, you specify the policy rule (*expression*) that defines the traffic to be examined by

© 1999-2017 Citrix Systems, Inc. All rights reserved.

this security configuration. If you are creating an initial configuration to protect your Web sites and Web services, you can simply accept the default value, true, which selects all web traffic .

If you want this security configuration to examine, not all HTTP traffic that is routed through the appliance, but specific traffic, you can write a policy rule specifying the traffic that you want it to examine. Rules are written in Citrix NetScaler expressions language, which is a fully functional object-oriented programming language.

- For a simple description of using the NetScaler expressions syntax to create Application Firewall rules, and a list of useful rules, see "[Firewall Policies]."
- For a detailed explanation of how to create policy rules in NetScaler expressions syntax, see "Policies and Expressions."

Note: In addition to the default expressions syntax, for backward compatibility the NetScaler operating system supports the NetScaler classic expressions syntax on NetScaler Classic and nCore appliances and virtual appliances. Classic expressions are not supported on NetScaler Cluster appliances and virtual appliances. Current users who want to migrate their existing configurations to the NetScaler cluster must migrate any policies that contain classic expressions to the default expressions syntax.

4. **Select Signature Protections screen.** On this screen, you select the categories of signatures that you want to use to protect your web sites and web services. The default categories are:

   - **CGI.** Protection against attacks on web sites that use CGI scripts in any language, including PERL scripts, Unix shell scripts, and Python scripts.
   - **Cold Fusion.** Protection against attacks on web sites that use the Adobe Systems® ColdFusion® Web development platform.
   - **FrontPage.** Protection against attacks on web sites that use the Microsoft® FrontPage® Web development platform.
   - **PHP.** Protection against attacks on web sites that use the PHP open-source Web development scripting language.
   - **Client side.** Protection against attacks on client-side tools used to access your protected web sites, such as Microsoft Internet Explorer, Mozilla Firefox, the Opera browser, and the Adobe Acrobat Reader.
   - **Microsoft IIS.** Protection against attacks on Web sites that run the Microsoft Internet Information Server (IIS).
   - **Miscellaneous.** Protection against attacks on other server-side tools, such as Web servers and database servers.

   If you are creating a new security configuration, the signature categories that you select are enabled, and by default they are recorded in a new signatures object. The new signatures object is assigned the same name that you entered on the Specify name screen as the name of the security configuration.

   If you have previously configured signatures objects and want to use one of them as the signatures object associated with the security configuration that you are creating, click Select Existing Signature and select a signatures object from the Signatures list.

   If you are modifying an existing security configuration, you can click Select Existing Signature and assign a different signatures object to the security configuration.

5. **Select Signature Actions screen.** On this screen, you select the actions associated with the signature categories that you selected on the Select signature protections screen. If you are creating an initial configuration, you might want to accept the defaults, which enable the Log and Stats actions but not the Block action. You can decide later, after

© 1999-2017 Citrix Systems, Inc. All rights reserved.

reviewing the collected logs and statistics, which signatures you should use to block traffic, and then enable the Block action for those signatures. Signatures are designed to catch specific known attacks on your web sites, and therefore they have extremely low false positive rates. However, with any new configuration, you should probably observe how the settings you chose are working before you use them to block traffic.

If you select More for one of the signature categories, the Configure Actions for Signatures dialog box appears. Its contents are the same as the contents of the Modify Signatures Object dialog box, as described in "To Configure a Signatures Object."

If the signatures object has already logged connections, you can click Logs to display the Syslog Viewer with the logs, as described in "Logs, Statistics, and Reports." If a signature is blocking legitimate access to your protected web site or web service, you can create and implement a relaxation for that signature by selecting a log that shows the unwanted blocking, and then clicking Deploy.

6. **Select Advanced Protections screen.** On this screen, you choose the advanced protections (also called *security checks* or simply *checks*) that you want to use to protect your web sites and web services. The checks are divided into categories. Which categories are available (and which checks are available within a category) depends on the profile type that you chose on the Specify Name screen. All checks are available for Web 2.0 Application profiles. If you chose that profile type, the Select advanced protections screen displays the following categories of security checks:
   - Top--level protections (Some checks appear at the top level, not in any category.)
   - Data Leak Prevention Protections
   - Advanced Form Protections
   - URL Protections
   - XML Protections

   To display the individual checks in a category, click the icon to the left of the category. To apply a security check to your filtered data, select the check box next to the name of the security check. For descriptions of the security checks see "Advanced Protections" and its subtopics.
7. **Select Advanced Actions screen.** On this screen, you configure the actions for the advanced protections that you have enabled.
   Note: If no advanced protections are enabled, the Wizard skips the Advanced Actions screen and goes directly to the Summary screen.
   The actions that you can configure are:
   - **Block.** Block connections that match the signature. Disabled by default.
   - **Log.** Log connections that match the signature for later analysis. Enabled by default.
   - **Stats.** Maintain statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.
   - **Learn.** Observe traffic to this Web site or Web service, and use connections that repeatedly violate this check to generate recommended exceptions to the check, or new rules for the check. Available only for some checks.

   To enable or disable an action for a check, in the list, select or clear the check box for that action to the right of that check.

   To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check. You can also select a check and, at the bottom of the dialog box, click Open to display a dialog box for modifying any of the options for that check. These dialog boxes also vary from check to check. Most of them include a Checks tab and a General tab. If the check supports relaxations, the Checks tab includes an Add button, which opens yet another dialog box, in which you can specify a relaxation for the check. A relaxation is a rule for exempting specified traffic from the check.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For information about the settings available for a check, see the detailed description of that check.

To review the recommendations generated by the learning engine for a specific check, select that check and then click Learned Violations to open the Manage Learned Rules dialog box for that check. For more information on how learning works and how to configure exceptions (relaxations) or deploy learned rules for a check, see "Manual Configuration By Using the Configuration Utility" under To configure and use the learning feature

To view all logs for a specific check, select that check, and then click Logs to display the Syslog Viewer, as described in "Logs, Statistics, and Reports." If a security check is blocking legitimate access to your protected web site or web service, you can create and implement a relaxation for that security check by selecting a log that shows the unwanted blocking, and then clicking Deploy.

8. **Summary screen.** On this screen, you review your configuration choices to verify that they are what you want. If you want to make changes, you click Back until you have returned to the appropriate screen, and make your changes. If the configuration is as you want it, you click Finish to save it , and then click Exit to close the Application Firewall wizard.

Following are four procedures that show how to perform specific types of configuration by using the Application Firewall wizard.

To configure the Application Firewall: Initial Configuration

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, in the Name text box, type a name for your new security configuration, and from the Type drop-down list, select the type of security configuration. Then, click Next.
5. On the Specify Rule screen, click Next again.
   Note: The default rule, **true**, protects all Web traffic that is sent via your NetScaler appliance or virtual appliances. You can create specific security configurations to protect specific parts of your Web sites or Web applications later.
6. On the Select Signature Protections screen, select check boxes to specify the groups of signatures that are appropriate for protecting the content on your protected web sites, and then click Next.
   For more information about signatures, see "Signatures."

7. On the Select Signature Actions screen, select or clear the associated check boxes to choose the signature actions that you want for each signature category that you selected in the previous step, and then click Next.
8. On the Select Advanced Protections screen, click Next again.
   You typically do not need to configure the security checks during initial configuration.

9. On the Summary screen, review your choices to verify that they are what you want. Then, click Finish, or click Back to return to a previous screen and make changes. When you are finished, click Exit to close the Application Firewall wizard.

To configure the Application Firewall: Enabling Blocking for Signatures

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, select Modify Existing Configuration and, in the Name drop-down list, choose the security configuration that you created during simple configuration, and then click Next.
5. In the Specify Rule screen, click Next again.
6. In the Select Signature Protections screen, click Next again.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

7. In the Select Signature Actions screen, enable blocking for your chosen signatures by selecting the Block check box to the left of each of those signature.
   For more information about which signatures to consider for blocking and how to determine when you can safely enable blocking for a signature, see "Signatures."

8. In the Select advanced protections screen, click Next.
9. On the Summary screen, review your choices to verify that they appear correct. Then, click Finish, or click Back to return to the Select Signature Actions screen and make changes. When you are finished, click Exit to close the Application Firewall wizard.

To configure the Application Firewall: Enabling and Configuring advanced protection

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, select Modify Existing Configuration and, in the Name drop-down list, choose the security configuration that you created during simple configuration. Then, click Next.
5. On the Specify Rule screen, click Next again.
6. On the Select Signature Protections screen, click Next.
7. On the Select Signature Actions screen, click Next again.
8. On the Select advanced protections screen, select the check box beside each security check that you want to enable, and then click Next.
   For information about the security checks, see "Advanced Protections" and its subtopics.

9. On the Select Deep Actions screen, select check boxes to specify the actions that you want the Application Firewall to perform for each security check, and then click Next.
   For general information about the actions, see "Advanced Protections" and its subtopics. For information about the learning feature, which is available for some security checks, see "To configure and use the Learning feature."

10. On the Summary screen, review your choices to verify that they appear correct. Then, click Finish, or click Back to return to the Select Signature Actions screen and make changes. When you are finished, click Exit to close the Application Firewall wizard.

To configure the Application Firewall: Creating A Policy

The following procedure describes how to use the Application Firewall wizard to create a specialized security configuration to protect only specific content. In this case, you create a new security configuration instead of modifying the initial configuration. This type of security configuration requires a custom rule, so that the policy applies the configuration to only the selected Web traffic.

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, type a name for your new security configuration in the Name text box, select the type of security configuration from the Type drop-down list, and then click Next.
5. On the Specify Rule screen, enter a rule that matches only that content that you want this Web application to protect, and then click Next.
   For a description of policies and policy rules, see "Policies."

6. On the Select Signature Protections screen, choose the appropriate groups of signatures to protect the content on

© 1999-2017 Citrix Systems, Inc. All rights reserved.

your protected web sites by selecting the check box beside each group of signatures, and then click Next.
For detailed information about signatures, see "Signatures."

7. On the Select Signature Actions screen, select or clear the associated check boxes to choose the signature actions that you want for each signature category that you selected in the previous step, and then click Next. For a detailed description of actions, see "Signatures."
8. In the Select Advanced Protections screen, select the check box beside each security check that you want to enable, and then click Next.
   For detailed information about the security checks, see "Advanced Protections" and its subtopics.
9. In the Select Advanced Actions screen, select check boxes to specify the actions that you want the Application Firewall to perform for each security check. Then, click Next.
   For information about each security check to help you determine which actions to enable, see the Advanced Protections section.
10. On the Summary screen, review your choices to verify that they appear correct. Then, click Finish, or click Back to return to the Select Signature Actions screen and make changes. When you are finished, click Exit to close the wizard.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Manual Configuration

Dec 03, 2015

If you want to bind a profile to a bind point other than Global, you must manually configure the binding. Also, certain security checks require that you either manually enter the necessary exceptions or enable the learning feature to generate the exceptions that your Web sites and Web services need. Some of these tasks cannot be performed by using the application firewall wizard.

If you are familiar with how the application firewall works and prefer manual configuration, you can manually configure a signatures object and a profile, associate the signatures object with the profile, create a policy with a rule that matches the web traffic that you want to configure, and associate the policy with the profile. You then bind the policy to Global, or to a bind point, to put it into effect, and you have created a complete security configuration.

For manual configuration, you can use the configuration utility (a graphical interface) or the command line. Citrix recommends that you use the configuration utility. Not all configuration tasks can be performed at the command line. Certain tasks, such as enabling signatures and reviewing learned data, must be done in the configuration utility. Most other tasks are easier to perform in the configuration utility.

Replicating Configuration

When you use the configuration utility (GUI) or the command line interface (CLI) to manually configure the application firewall, the configuration is saved in the /nsconfig/ns.conf file. You can use the commands in that file to replicate the configuration on another appliance. You can cut and paste the commands into the CLI one by one, or you can save multiple commands in a text file in the /var/tmp folder and run them as a batch file. Following is an example of running a batch file containing commands copied from the /nsconfig/ns.conf file of a different appliance:

> **batch -f** /var/tmp/appfw_add.txt

Warning

Import commands are not saved in the ns.conf file. Before running commands from the ns.conf file to replicate the configuration on another appliance, you must import all the objects used in the configuration (for example, signatures, error page, WSDL, and Schema) to the appliance on which you will replicate the configuration. The add command to add an application firewall profile saved in an ns.conf file might include the name of an imported object, but such a command might fail when executed on another appliance if the referenced object does not exist on that appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Manual Configuration By Using the Configuration Utility

Sep 03, 2013

If you need to configure the Application Firewall feature manually, Citrix recommends that you use the configuration utility. For a description of the configuration utility, see "The Application Firewall User Interfaces."

To create and configure a signatures object

Before you can configure the signatures, you must create a new signatures object from the appropriate default signatures object template. Assign the copy a new name, and then configure the copy. You cannot configure or modify the default signatures objects directly. The following procedure provides basic instructions for configuring a signatures object. For more detailed instructions, see "Manually Configuring the Signatures Feature." If you need to create your own, user defined signatures, see "The Signatures Editor."

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to use as a template, and then click Add.
   Your choices are:
   - *** Default Signatures.** Contains the signatures rules, the SQL injection rules, and the cross-site scripting rules.
   - *** XPath Injection.** Contains all of the items in the * Default Signatures, and in addition contains the XPath injection rules.
3. In the Add Signatures Object dialog box, type a name for your new signatures object, click OK, and then click Close. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols.
4. Select the signatures object that you created, and then click Open.
5. In the Modify Signatures Object dialog box, set the Display Filter Criteria options at the left to display the filter items that you want to configure.
   As you modify these options, the results that you specify are displayed in the Filtered Results window at the right. For more information about the categories of signatures, see "Signatures."
6. In the Filtered Results area, configure the settings for a signature by selecting and clearing the appropriate check boxes.
7. When finished, finished, click Close.

To create an application firewall profile by using the configuration utility

Creating an application firewall profile requires that you specify only a few configuration details.

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, click Add.
3. In the Create Application Firewall Profile dialog box, type a name for your profile.
   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.
4. Choose the profile type from the drop-down list.
5. Click Create, and then click Close.

To configure an application firewall profile by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, select the profile that you want to configure, and then click Edit.
3. In the Configure Application Firewall Profile dialog box, on the Security Checks tab, configure the security checks.
   - To enable or disable an action for a check, in the list, select or clear the check box for that action.
   - To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check.
     You can also select a check and, at the bottom of the dialog box, click Open to display the Configure Relaxation dialog box or Configure Rule dialog box for that check. These dialog boxes also vary from check to check. Most of them include a Checks tab and a General tab. If the check supports relaxations or user-defined rules, the Checks tab includes an Add button, which opens yet another dialog box, in which you can specify a relaxation or rule for the check. (A relaxation is a rule for exempting specified traffic from the check.) If relaxations have already been configured, you can select one and click Open to modify it.

   - To review learned exceptions or rules for a check, select the check, and then click Learned Violations. In the Manage Learned Rules dialog box, select each learned exception or rule in turn.
     - To edit the exception or rule, and then add it to the list, click Edit & Deploy.
     - To accept the exception or rule without modification, click Deploy.
     - To remove the exception or rule from the list, click Skip.
   - To refresh the list of exceptions or rules to be reviewed, click Refresh.
   - To open the Learning Visualizer and use it to review learned rules, click Visualizer.
   - To review the log entries for connections that matched a check, select the check, and then click Logs. You can use this information to determine which checks are matching attacks, so that you can enable blocking for those checks. You can also use this information to determine which checks are matching legitimate traffic, so that you can configure an appropriate exemption to allow those legitimate connections. For more information about the logs, see "Logs, Statistics, and Reports."
   - To completely disable a check, in the list, clear all of the check boxes to the right of that check.
4. On the Settings tab, configure the profile settings.
   - To associate the profile with the set of signatures that you previously created and configured, under Common Settings, choose that set of signatures in the Signatures drop-down list.
     Note: You may need to use the scroll bar on the right of the dialog box to scroll down to display the Common Settings section.
   - To configure an HTML or XML Error Object, select the object from the appropriate drop-down list.
     Note: You must first upload the error object that you want to use in the Imports pane. For more information about importing error objects, see "Imports."
   - To configure the default XML Content Type, type the content type string directly into the Default Request and Default Response text boxes, or click Manage Allowed Content Types to manage the list of allowed content types. ">>More..."
5. If you want to use the learning feature, click Learning, and configure the learning settings for the profile, as described in "Configuring and Using the Learning Feature".
6. Click OK to save your changes and return to the Profiles pane.

Configuring an Application Firewall Rule or Relaxation

Updated: 2014-06-12

You configure two different types of information in this dialog box, depending upon which security check you are configuring. In the majority of cases, you configure an exception (or relaxation) to the security check. If you are configuring the Deny URL check or the Field Formats check, you configure an addition (or rule). The process for either of these is the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

same.

To configure a relaxation or rule by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the Profiles pane, select the profile you want to configure, and then click Edit.
3. In the Configure Application Firewall Profile dialog box, click the Security Checks tab. The Security Checks tab contains the complete list of application firewall security checks, also called *advanced protections* in some places.
4. In the Security Checks tab, click the check that you want to configure, and then click Open. The Modify Check dialog box for the check that you chose is displayed, with the Checks tab selected. The Checks tab contains a list of existing relaxations or rules for this check. The list might be empty if you have not either manually added any relaxations or approved any relaxations that were recommended by the learning engine. Beneath the list is a row of buttons that allow you to add, modify, delete, enable, or disable the relaxations on the list.
5. To add or modify a relaxation or a rule, do one of the following:
   - To add a new relaxation, click Add.
   - To modify an existing relaxation, select the relaxation that you want to modify, and then click Open.

   The Add Check Relaxation or Modify Check Relaxation dialog box for the selected check is displayed. Except for the title, these dialog boxes are identical.

6. Fill in the dialog box as described below. The dialog boxes for each check are different; the list below covers all elements that might appear in any dialog box.
   - **Enabled check box**—Select to place this relaxation or rule in active use; clear to deactivate it.
   - **Attachment Content Type**—The Content-Type attribute of an XML attachment. In the text area, enter a regular expression that matches the Content-Type attribute of the XML attachments to allow.
   - **Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the web form is delivered.
   - **Cookie**—In the text area, enter a PCRE-format regular expression that defines the cookie.
   - **Field Name**—A web form field name element may be labeled Field Name, Form Field, or another similar name. In the text area, enter a PCRE-format regular expression that defines the name of the form field.
   - **Form Origin URL**—In the text area, enter a PCRE-format regular expression that defines the URL that hosts the web form.
   - **Form Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the web form is delivered.
   - **Name**—An XML element or attribute name. In the text area, enter a PCRE-format regular expression that defines the name of the element or attribute.
   - **URL**—A URL element may be labeled Action URL, Deny URL, Form Action URL, Form Origin URL, Start URL, or simply URL. In the text area, enter a PCRE-format regular expression that defines the URL.
   - **Format**—The format section contains multiple settings that include list boxes and text boxes. Any of the following can appear:
     - **Type**—Select a field type in the Type drop-down list. To add a new field type definition, click Manage—
     - **Minimum Length**—Type a positive integer that represents the minimum length in characters if you want to force

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

users to fill in this field. Default: 0 (Allows field to be left blank.)
    - **Maximum length**—To limit the length of data in this field, type a positive integer that represents the maximum length in characters. Default: 65535
- **Location**—Choose the element of the request that your relaxation will apply to from the drop-down list. For HTML security checks, the choices are:
    - FORMFIELD—Form fields in web forms.
    - HEADER—Request headers.
    - COOKIE—Set-Cookie headers.

    For XML security checks, the choices are:
    - ELEMENT—XML element.
    - ATTRIBUE—XML attribute.
- **Maximum Attachment Size**—The maximum size in bytes allowed for an XML attachment.
- **Comments**—In the text area, type a comment. Optional.

Note: For any element that requires a regular expression, you can type the regular expression, use the Regex Tokens menu to insert regular expression elements and symbols directly into the text box, or click Regex Editor to open the Add Regular Expression dialog box, and use it to construct the expression.

7. To remove a relaxation or rule, select it, and then click Remove.
8. To enable a relaxation or rule, select it, and then click Enable.
9. To disable a relaxation or rule, select it, and then click Disable.
10. To configure the settings and relationships of all existing relaxations in an integrated interactive graphic display, click Visualizer, and use the display tools.
    Note: The Visualizer button does not appear on all check relaxation dialog boxes.
11. To review learned rules for this check, click Learning and perform the steps in "To configure and use the Learning feature."
12. Click OK.

To configure the Learning feature by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the Profiles pane, select the profile, and then click Edit.
3. Click the Learning tab. At the top of the Learning tab is list of the security checks that are available in the current profile and that support the learning feature.
4. To configure the learning thresholds, select a security check, and then type the appropriate values in the following text boxes:
    - **Minimum number threshold.** Depending on which security check's learning settings you are configuring, the minimum number threshold might refer to the minimum number of total user sessions that must be observed, the minimum number of requests that must be observed, or the minimum number of times a specific form field must be observed, before a learned relaxation is generated. Default: 1
    - **Percentage of times threshold.** Depending on which security check's learning settings you are configuring, the percentage of times threshold might refer to the percentage of total observed user sessions that violated the security check, the percentage of requests, or the percentage of times a form field matched a particular field type, before a learned relaxation is generated. Default: 0
5. To remove all learned data and reset the learning feature, so that it must start its observations again from the beginning, click Remove All Learned Data.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: This button removes only learned recommendations that have not been reviewed and either approved or skipped. It does not remove learned relaxations that have been accepted and deployed.

6. To restrict the learning engine to traffic from a specific set of IPs, click Trusted Learning Clients, and add the IP addresses that you want to use to the list.
   1. To add an IP address or IP address range to the Trusted Learning Clients list, click Add.
   2. In the Add Trusted Learning Clients dialog box, Trusted Clients IP list box, type the IP address or an IP address range in CIDR format.
   3. In the Comments text area, type a comment that describes this IP address or range.
   4. Click Create to add your new IP address or range to the list.
   5. To modify an existing IP address or range, click the IP address or range, and then click Open. Except for the name, the dialog box that appears is identical to the Add Trusted Learning Clients dialog box.
   6. To disable or enable an IP address or range, but leave it on the list, click the IP address or range, and then click Disable or Enable, as appropriate.
   7. To remove an IP address or range completely, click the IP address or range, and then click Remove.
7. Click Close to return to the Configure Application Firewall Profile dialog box.
8. Click Close to close the Configure Application Firewall Profile dialog box, and return to the Application Firewall Profile screen.

To create and configure a policy by using the configuration utility

1. Navigate to Security > Application Firewall > Policies.
2. In the details pane, do one of the following:
   - To create a new firewall policy, click Add. The Create Application Firewall Policy is displayed.
   - To edit an existing firewall policy, select the policy, and then click Edit.

   The Create Application Firewall Policy or Configure Application Firewall Policy is displayed.
3. If you are creating a new firewall policy, in the Create Application Firewall Policy dialog box, Policy Name text box, type a name for your new policy.
   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 128 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

   If you are configuring an existing firewall policy, this field is read-only. You cannot modify it.

4. Select the profile that you want to associate with this policy from the Profile drop-down list. You can create a new profile to associate with your policy by clicking New, and you can modify an existing profile by clicking Modify.
5. In the Expression text area, create a rule for your policy.
   - You can type a rule directly into the text area.
   - You can click Prefix to select the first term for your rule, and follow the prompts. See "To Create an Application Firewall Rule (Expression)" for a complete description of this process.
   - You can click Add to open the Add Expression dialog box, and use it to construct the rule. See "The Add Expression Dialog Box" for a complete description of this process.
6. Click Create or OK, and then click Close.

To create or configure an Application Firewall rule (expression)

The policy rule, also called the *expression*, defines the web traffic that the application firewall filters by using the profile associated with the policy. Like other NetScaler policy rules (or *expressions*), application firewall rules use NetScaler expressions syntax. This syntax is powerful, flexible, and extensible. It is too complex to describe completely in this set of instructions. You can use the following procedure to create a simple firewall policy rule, or you can read it as an overview of the policy creation process.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. If you have not already done so, navigate to the appropriate location in the Application Firewall wizard or the NetScaler configuration utility to create your policy rule:
   - If you are configuring a policy in the Application Firewall wizard, in the navigation pane, click Application Firewall, then in the details pane click Application Firewall Wizard, and then navigate to the Specify Rule screen.
   - If you are configuring a policy manually, in the navigation pane, expand Application Firewall, then Policies, and then Firewall. In the details pane, to create a new policy, click Add. To modify an existing policy, select the policy, and then click Open.
2. On the Specify Rule screen, the Create Application Firewall Profile dialog box, or the Configure Application Firewall Profile dialog box, click Prefix, and then choose the prefix for your expression from the drop-down list. Your choices are:
   - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

   After you choose a prefix, the application firewall displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom.

3. Choose your next term.
   If you chose HTTP as your prefix, your only choice is REQ, which specifies the Request/Response pair. (The application firewall operates on the request and response as a unit instead of on each separately.) If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

   When you have decided which term you want, double-click it to insert it into the Expression window.

4. Type a period after the term you just chose. You are then prompted to choose your next term, as described in the previous step. When a term requires that you type a value, fill in the appropriate value. For example, if you choose HTTP.REQ.HEADER(""), type the header name between the quotation marks.
5. Continue choosing terms from the prompts and filling in any values that are needed, until your expression is finished. Following are some examples of expressions for specific purposes.

   - **Specific web host.** To match traffic from a particular web host:

     HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
     For shopping.example.com, substitute the name of the web host that you want to match.

   - **Specific web folder or directory.** To match traffic from a particular folder or directory on a Web host:

     HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
     For www.example.com, substitute the name of the web host. For folder, substitute the folder or path to the content that you want to match. For example, if your shopping cart is in a folder called /solutions/orders, you substitute that string for folder.

   - **Specific type of content: GIF images.** To match GIF format images:

     HTTP.REQ.URL.ENDSWITH(".gif")
     To match other format images, substitute another string in place of .gif.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Specific type of content: scripts.** To match all CGI scripts located in the CGI-BIN directory:

  HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
  To match all JavaScripts with .js extensions:

  HTTP.REQ.URL.ENDSWITH(".js")
  For more information about creating policy expressions, see "Policies and Expressions."

Note: If you use the command line to configure a policy, remember to escape any double quotation marks within NetScaler expressions. For example, the following expression is correct if entered in the configuration utility:
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
If entered at the command line, however, you must type this instead:

HTTP.REQ.HEADER(\"Host\").EQ(\"shopping.example.com\")

To add a firewall rule (expression) by using the Add Expression dialog box

The Add Expression dialog box (also referred to as the Expression Editor) helps users who are not familiar with the NetScaler expressions language to construct a policy that matches the traffic that they want to filter.

1. If you have not already done so, navigate to the appropriate location in the Application Firewall wizard or the NetScaler configuration utility:
   - If you are configuring a policy in the Application Firewall wizard, in the navigation pane, click Application Firewall, then in the details pane click Application Firewall Wizard, and then navigate to the Specify Rule screen.
   - If you are configuring a policy manually, in the navigation pane, expand Application Firewall, then Policies, and then Firewall. In the details pane, to create a new policy, click Add. To modify an existing policy, select the policy, and then click Open.
2. On the Specify Rule screen, in the Create Application Firewall Profile dialog box, or in the Configure Application Firewall Profile dialog box, click Add.
3. In the Add Expression dialog box, in the Construct Expression area, in the first list box, choose one of the following prefixes:
   - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol. The default choice.
   - **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.
4. In the second list box, choose your next term. The available terms differ depending on the choice you made in the previous step, because the dialog box automatically adjusts the list to contain only those terms that are valid for the context. For example, if you selected HTTP in the previous list box, the only choice is REQ, for requests. Because the application firewall treats requests and associated responses as a single unit and filters both, you do not need to specific responses separately. After you choose your second term, a third list box appears to the right of the second. The Help window displays a description of the second term, and the Preview Expression window displays your expression.
5. In the third list box, choose the next term. A new list box appears to the right, and the Help window changes to display a description of the new term. The Preview Expression window updates to display the expression as you have specified it to that point.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

6. Continue choosing terms, and when prompted filling in arguments, until your expression is complete. If you make a mistake or want to change your expression after you have already selected a term, you can simply choose another term. The expression is modified, and any arguments or additional terms that you added after the term that you modified are cleared.
7. When you have finished constructing your expression, click OK to close the Add Expression dialog box. Your expression is inserted into the Expression text area.

To bind an application firewall policy by using the configuration utility

1. Do one of the following:
   - Navigate to Security > Application Firewall, and in the details pane, click Application Firewall policy manager.
   - Navigate to Security > Application Firewall > Policies > Firewall Policies, and in the details pane, click Policy Manager.
2. In the Application Firewall Policy Manager dialog, choose the bind point to which you want to bind the policy from the drop-down list. The choices are:
   - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance, and are applied before any other policies.
   - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
   - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
   - **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance.
   - **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.
   - **None.** Do not bind the policy to any bind point.
3. Click Continue. A list of existing application firewall policies appears.
4. Select the policy you want to bind by clicking it.
5. Make any additional adjustments to the binding.
   - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
   - To modify the policy expression, double click that field to open the Configure Application Firewall Policy dialog box, where you can edit the policy expression.
   - To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
   - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression
6. Repeat steps 3 through 6 to add any additional application firewall policies you want to globally bind.
7. Click OK. A message appears in the status bar, stating that the policy has been successfully bound.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Manual Configuration By Using the Command Line Interface

Dec 17, 2013

You can configure many application firewall features from the NetScaler command line. There are important exceptions, however. You cannot enable signatures from the command line. There are over 1,000 default signatures in seven categories; the task is simply too complex for the command line interface. You can configure the check actions and parameters for security checks from the command line, but cannot enter manual relaxations. While you can configure the adaptive learning feature and enable learning from the command line, you cannot review learned relaxations or learned rules and approve or skip them. The command line interface is intended for advanced users who are thoroughly familiar with the NetScaler appliance and the application firewall feature.

To manually configure the application firewall by using the NetScaler command line, use a telnet or secure shell client of your choice to log on to the NetScaler command line.

To create a profile by using the command line interface

At the command prompt, type the following commands:

- add appfw profile <name> [-defaults (**basic** | **advanced**)]
- set appfw profile <name> -type (**HTML** | **XML** | **HTML XML**)
- save ns config

Example

The following example adds a profile named pr-basic, with basic defaults, and assigns a profile type of HTML. This is the appropriate initial configuration for a profile to protect an HTML Web site.

```
add appfw profile pr-basic -defaults basic
set appfw profile pr-basic -type HTML
save ns config
```

To configure a profile by using the command line interface

At the command prompt, type the following commands:

- set appfw profile <name> <arg1> [<arg2> ...] where <arg1> represents a parameter and <arg2> represents either another parameter or the value to assign to the parameter represented by <arg1>. For descriptions of the parameters to use when configuring specific security checks, see [Advanced Protections] and its subtopics. For descriptions of the other parameters, see "Parameters for Creating a Profile."
- save ns config

Example

The following example shows how to configure an HTML profile created with basic defaults to begin protecting a simple HTML-based Web site. This example turns on logging and maintenance of statistics for most security checks, but enables blocking only for those checks that have extremely low false positive rates and require no special configuration. It also turns on transformation of unsafe HTML and unsafe SQL, which prevents attacks but does not block requests to your Web sites. With logging and statistics enabled, you can later review the logs to determine whether to enable blocking for a

© 1999-2017 Citrix Systems, Inc. All rights reserved.

specific security check.

```
set appfw profile -startURLAction log stats
set appfw profile -denyURLAction block log stats
set appfw profile -cookieConsistencyAction log stats
set appfw profile -crossSiteScriptingAction log stats
set appfw profile -crossSiteScriptingTransformUnsafeHTML ON
set appfw profile -fieldConsistencyAction log stats
set appfw profile -SQLInjectionAction log stats
set appfw profile -SQLInjectionTransformSpecialChars ON
set appfw profile -SQLInjectionOnlyCheckFieldsWithSQLChars ON
set appfw profile -SQLInjectionParseComments checkall
set appfw profile -fieldFormatAction log stats
set appfw profile -bufferOverflowAction block log stats
set appfw profile -CSRFtagAction log stats
save ns config
```

To create and configure a policy

At the command prompt, type the following commands:

- add appfw policy <name> <rule> <profile>
- save ns config

Example

The following example adds a policy named pl-blog, with a rule that intercepts all traffic to or from the host blog.example.com, and associates that policy with the profile pr-blog. This is an appropriate policy to protect a blog hosted on a specific hostname.

```
add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com")" pr-blog
```

To bind an Application Firewall policy

At the command prompt, type the following commands:

- bind appfw global <policyName> <priority>
- save ns config

Example

The following example binds the policy named pl-blog and assigns it a priority of 10.

```
bind appfw global pl-blog 10
save ns config
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

The Application Firewall Wizard

Oct 01, 2013

Unlike most wizards, the Application Firewall wizard is designed not just to simplify the initial configuration process, but also to modify previously created configurations and to maintain your Application Firewall setup. A typical user runs the wizard multiple times, skipping some of the screens each time.

Opening the Wizard

To run the Application Firewall wizard, first open the configuration utility. Next, in the navigation pane, expand Application Firewall, and then in the details pane click Application Firewall Wizard. (For more information about the configuration utility, see "The Application Firewall User Interfaces.") Then:

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard. The first screen of the wizard appears.
3. To advance to the next screen, click Next.

The Wizard Screens

The Application Firewall wizard displays the following screens, in the following order:

1. **Introduction screen.** Provides an introduction to the Application Firewall wizard. There is nothing that you can configure on this screen.

2. **Specify Name screen.** On this screen, when creating a new security configuration, you specify the name that the wizard is to assign to the configuration. The name can begin with a letter, number, or the underscore symbol, and can consist of from 1 to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols. Choose a name that makes it easy for others to tell what content your new security configuration protects.

   Note: Because the wizard uses this name for both the policy and the profile, it is limited to 31 characters. Manually created policies can have names up to 127 characters in length.
   When creating an existing configuration, you select Modify Existing Configuration and then, in the Name drop-down list, select the name of the existing configuration that you want to modify.

   Note: Only policies that are bound to global or to a bind point appear in this list; you cannot modify an unbound policy by using the Application Firewall wizard. You must either manually bind it to Global or a bind point, or modify it manually. (For manual modification, in the configuration utility's Application Firewall --> Policies --> Firewall pane, select the policy and click Open).
   You also select a profile type on this screen. The profile type determines the types of advanced protection (security checks) that can be configured. Because certain kinds of content are not vulnerable to certain types of security threats, restricting the list of available checks saves time during configuration. The types of Application Firewall profiles are:

   - **Web Application (HTML).** Any HTML-based Web site that does not use XML or Web 2.0 technologies.
   - **XML Application (XML, SOAP).** Any XML-based Web service.
   - **Web 2.0 Application (HTML, XML, REST).** Any Web 2.0 site that combines HTML and XML-based content, such as an ATOM-based site, a blog, an RSS feed, or a wiki.

   Note: If you are unsure which type of content is used on your Web site, you can choose Web 2.0 Application to ensure that you protect all types of Web application content.
3. **Specify Rule screen.** On this screen, you specify the policy rule (*expression*) that defines the traffic to be examined by

© 1999-2017 Citrix Systems, Inc. All rights reserved.

this security configuration. If you are creating an initial configuration to protect your Web sites and Web services, you can simply accept the default value, true, which selects all web traffic .

If you want this security configuration to examine, not all HTTP traffic that is routed through the appliance, but specific traffic, you can write a policy rule specifying the traffic that you want it to examine. Rules are written in Citrix NetScaler expressions language, which is a fully functional object-oriented programming language.

- For a simple description of using the NetScaler expressions syntax to create Application Firewall rules, and a list of useful rules, see "Firewall Policies."
- For a detailed explanation of how to create policy rules in NetScaler expressions syntax, see "Policies and Expressions."

Note: In addition to the default expressions syntax, for backward compatibility the NetScaler operating system supports the NetScaler classic expressions syntax on NetScaler Classic and nCore appliances and virtual appliances. Classic expressions are not supported on NetScaler Cluster appliances and virtual appliances. Current users who want to migrate their existing configurations to the NetScaler cluster must migrate any policies that contain classic expressions to the default expressions syntax.

4. **Select Signature Protections screen.** On this screen, you select the categories of signatures that you want to use to protect your web sites and web services. The default categories are:

   - **CGI.** Protection against attacks on web sites that use CGI scripts in any language, including PERL scripts, Unix shell scripts, and Python scripts.
   - **Cold Fusion.** Protection against attacks on web sites that use the Adobe Systems® ColdFusion® Web development platform.
   - **FrontPage.** Protection against attacks on web sites that use the Microsoft® FrontPage® Web development platform.
   - **PHP.** Protection against attacks on web sites that use the PHP open-source Web development scripting language.
   - **Client side.** Protection against attacks on client-side tools used to access your protected web sites, such as Microsoft Internet Explorer, Mozilla Firefox, the Opera browser, and the Adobe Acrobat Reader.
   - **Microsoft IIS.** Protection against attacks on Web sites that run the Microsoft Internet Information Server (IIS).
   - **Miscellaneous.** Protection against attacks on other server-side tools, such as Web servers and database servers.

   If you are creating a new security configuration, the signature categories that you select are enabled, and by default they are recorded in a new signatures object. The new signatures object is assigned the same name that you entered on the Specify name screen as the name of the security configuration.

   If you have previously configured signatures objects and want to use one of them as the signatures object associated with the security configuration that you are creating, click Select Existing Signature and select a signatures object from the Signatures list.

   If you are modifying an existing security configuration, you can click Select Existing Signature and assign a different signatures object to the security configuration.

5. **Select Signature Actions screen.** On this screen, you select the actions associated with the signature categories that you selected on the Select signature protections screen. If you are creating an initial configuration, you might want to accept the defaults, which enable the Log and Stats actions but not the Block action. You can decide later, after

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

reviewing the collected logs and statistics, which signatures you should use to block traffic, and then enable the Block action for those signatures. Signatures are designed to catch specific known attacks on your web sites, and therefore they have extremely low false positive rates. However, with any new configuration, you should probably observe how the settings you chose are working before you use them to block traffic.

If you select More for one of the signature categories, the Configure Actions for Signatures dialog box appears. Its contents are the same as the contents of the Modify Signatures Object dialog box, as described in "To Configure a Signatures Object."

If the signatures object has already logged connections, you can click Logs to display the Syslog Viewer with the logs, as described in "Logs, Statistics, and Reports." If a signature is blocking legitimate access to your protected web site or web service, you can create and implement a relaxation for that signature by selecting a log that shows the unwanted blocking, and then clicking Deploy.

6. **Select Advanced Protections screen.** On this screen, you choose the advanced protections (also called *security checks* or simply *checks*) that you want to use to protect your web sites and web services. The checks are divided into categories. Which categories are available (and which checks are available within a category) depends on the profile type that you chose on the Specify Name screen. All checks are available for Web 2.0 Application profiles. If you chose that profile type, the Select advanced protections screen displays the following categories of security checks:
   - Top--level protections (Some checks appear at the top level, not in any category.)
   - Data Leak Prevention Protections
   - Advanced Form Protections
   - URL Protections
   - XML Protections

   To display the individual checks in a category, click the icon to the left of the category. To apply a security check to your filtered data, select the check box next to the name of the security check. For descriptions of the security checks see "Advanced Protections" and its subtopics.
7. **Select Advanced Actions screen.** On this screen, you configure the actions for the advanced protections that you have enabled.
   Note: If no advanced protections are enabled, the Wizard skips the Advanced Actions screen and goes directly to the Summary screen.
   The actions that you can configure are:
   - **Block.** Block connections that match the signature. Disabled by default.
   - **Log.** Log connections that match the signature for later analysis. Enabled by default.
   - **Stats.** Maintain statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.
   - **Learn.** Observe traffic to this Web site or Web service, and use connections that repeatedly violate this check to generate recommended exceptions to the check, or new rules for the check. Available only for some checks.

   To enable or disable an action for a check, in the list, select or clear the check box for that action to the right of that check.

   To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check. You can also select a check and, at the bottom of the dialog box, click Open to display a dialog box for modifying any of the options for that check. These dialog boxes also vary from check to check. Most of them include a Checks tab and a General tab. If the check supports relaxations, the Checks tab includes an Add button, which opens yet another dialog box, in which you can specify a relaxation for the check. A relaxation is a rule for exempting specified traffic from the check.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

For information about the settings available for a check, see the detailed description of that check.

To review the recommendations generated by the learning engine for a specific check, select that check and then click Learned Violations to open the Manage Learned Rules dialog box for that check. For more information on how learning works and how to configure exceptions (relaxations) or deploy learned rules for a check, see "Manual Configuration By Using the Configuration Utility" under To configure and use the learning feature

To view all logs for a specific check, select that check, and then click Logs to display the Syslog Viewer, as described in "Logs, Statistics, and Reports." If a security check is blocking legitimate access to your protected web site or web service, you can create and implement a relaxation for that security check by selecting a log that shows the unwanted blocking, and then clicking Deploy.

8. **Summary screen.** On this screen, you review your configuration choices to verify that they are what you want. If you want to make changes, you click Back until you have returned to the appropriate screen, and make your changes. If the configuration is as you want it, you click Finish to save it , and then click Exit to close the Application Firewall wizard.

Following are four procedures that show how to perform specific types of configuration by using the Application Firewall wizard.

To configure the Application Firewall: Initial Configuration

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, in the Name text box, type a name for your new security configuration, and from the Type drop-down list, select the type of security configuration. Then, click Next.
5. On the Specify Rule screen, click Next again.
   Note: The default rule, true, protects all Web traffic that is sent via your NetScaler appliance or virtual appliances. You can create specific security configurations to protect specific parts of your Web sites or Web applications later.
6. On the Select Signature Protections screen, select check boxes to specify the groups of signatures that are appropriate for protecting the content on your protected web sites, and then click Next.
   For more information about signatures, see "Signatures."
7. On the Select Signature Actions screen, select or clear the associated check boxes to choose the signature actions that you want for each signature category that you selected in the previous step, and then click Next.
8. On the Select Advanced Protections screen, click Next again.
   You typically do not need to configure the security checks during initial configuration.
9. On the Summary screen, review your choices to verify that they are what you want. Then, click Finish, or click Back to return to a previous screen and make changes. When you are finished, click Exit to close the Application Firewall wizard.

To configure the Application Firewall: Enabling Blocking for Signatures

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, select Modify Existing Configuration and, in the Name drop-down list, choose the security configuration that you created during simple configuration, and then click Next.
5. In the Specify Rule screen, click Next again.
6. In the Select Signature Protections screen, click Next again.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

7. In the Select Signature Actions screen, enable blocking for your chosen signatures by selecting the Block check box to the left of each of those signature.
   For more information about which signatures to consider for blocking and how to determine when you can safely enable blocking for a signature, see "Signatures."

8. In the Select advanced protections screen, click Next.
9. On the Summary screen, review your choices to verify that they appear correct. Then, click Finish, or click Back to return to the Select Signature Actions screen and make changes. When you are finished, click Exit to close the Application Firewall wizard.

To configure the Application Firewall: Enabling and Configuring advanced protection

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, select Modify Existing Configuration and, in the Name drop-down list, choose the security configuration that you created during simple configuration. Then, click Next.
5. On the Specify Rule screen, click Next again.
6. On the Select Signature Protections screen, click Next.
7. On the Select Signature Actions screen, click Next again.
8. On the Select advanced protections screen, select the check box beside each security check that you want to enable, and then click Next.
   For information about the security checks, see "Advanced Protections" and its subtopics.

9. On the Select Deep Actions screen, select check boxes to specify the actions that you want the Application Firewall to perform for each security check, and then click Next.
   For general information about the actions, see "Advanced Protections" and its subtopics. For information about the learning feature, which is available for some security checks, see "To configure and use the Learning feature."

10. On the Summary screen, review your choices to verify that they appear correct. Then, click Finish, or click Back to return to the Select Signature Actions screen and make changes. When you are finished, click Exit to close the Application Firewall wizard.

To configure the Application Firewall: Creating A Policy

The following procedure describes how to use the Application Firewall wizard to create a specialized security configuration to protect only specific content. In this case, you create a new security configuration instead of modifying the initial configuration. This type of security configuration requires a custom rule, so that the policy applies the configuration to only the selected Web traffic.

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, type a name for your new security configuration in the Name text box, select the type of security configuration from the Type drop-down list, and then click Next.
5. On the Specify Rule screen, enter a rule that matches only that content that you want this Web application to protect, and then click Next.
   For a description of policies and policy rules, see "Policies."

6. On the Select Signature Protections screen, choose the appropriate groups of signatures to protect the content on

© 1999-2017 Citrix Systems, Inc. All rights reserved.

your protected web sites by selecting the check box beside each group of signatures, and then click Next.
For detailed information about signatures, see "Signatures."

7. On the Select Signature Actions screen, select or clear the associated check boxes to choose the signature actions that you want for each signature category that you selected in the previous step, and then click Next. For a detailed description of actions, see "Signatures."
8. In the Select Advanced Protections screen, select the check box beside each security check that you want to enable, and then click Next.
   For detailed information about the security checks, see "Advanced Protections" and its subtopics.
9. In the Select Advanced Actions screen, select check boxes to specify the actions that you want the Application Firewall to perform for each security check. Then, click Next.
   For information about each security check to help you determine which actions to enable, see the Advanced Protections section.
10. On the Summary screen, review your choices to verify that they appear correct. Then, click Finish, or click Back to return to the Select Signature Actions screen and make changes. When you are finished, click Exit to close the wizard.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Signatures

Jan 27, 2016

The application firewall signatures function provides specific, configurable rules to simplify the task of protecting your web sites against known attacks. A signature represents a pattern that is a component of a known attack on an operating system, web server, website, XML-based web service, or other resource. A rich set of preconfigured application firewall built-in or Native rules offers an easy to use security solution, leveraging the power of pattern matching to detect attacks and protect against application vulnerabilities.

You can create your own signatures or use the signatures in the built-in templates. The application firewall has two built-in templates:

- ***Default Signatures**: This template contains a preconfigured list of over 1,300 signatures, in addition to a complete list of SQL injection keywords, SQL special strings, SQL transform rules, and SQL wildchar characters. It also contains denied patterns for cross-site scripting, and allowed attributes and tags for cross-site scripting. This is a read-only template. You can view the contents, but you cannot add, edit, or delete anything in this template. To use it, you must make a copy. In your own copy, you can enable the signature rules that you want to apply to your traffic, and specify the actions to be taken when the signature rules match the traffic.

  The application firewall signatures are derived from the rules published by Snort, which is an open source intrusion prevention system capable of performing real-time traffic analysis to detect a variety of attacks and probes.

- ***Xpath Injection Patterns**: This template contains a preconfigured set of literal and PCRE keywords and special strings that are used to detect XPath (XML Path Language) injection attacks.

Blank Signatures: In addition to making a copy of the built-in *Default Signatures template, you can use a blank signatures template to create a new signature object. The signature object that you create with the blank signatures option does not have any native signature rules, but, just like the *Default template, it has all the SQL/XSS built-in entities.
External-Format Signatures: The application firewall also supports the use of external format signatures. You can import the scan files of the third party scan tools by using the XSLT files that are supported by the Citrix application firewall. A set of built-in XSLT files are available for the following scan tools to translate these external format files to the Native format:

- Cenzic
- Deep Security for Web Apps
- IBM AppScan Enterprise
- IBM AppScan Standard.
- Qualys
- Whitehat
- Hewlett Packard Enterprise WebInspect

Protection Options for Your Application

Tighter security increases processing overhead. Signatures provide the following deployment options to help you to optimize the protection of your applications:

- **Negative Security Model**: With the negative security model, you use a rich set of preconfigured signature rules to leverage the power of pattern matching to detect attacks and protect against application vulnerabilities. You block only what you don't want and allow the rest.[DR1] You can add your own signature rules, based on the specific security needs of your applications, to design your own customized security solutions.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Hybrid security Model**: In addition to using signatures, you can use positive security checks to create a configuration ideally suited for your applications. Use signatures to block what you don't want, and use positive security checks to enforce what is allowed.

To protect your application by using signatures, you must configure one or more profiles to use your signatures object. In a hybrid security configuration, the SQL injection and Cross-Site scripting patterns, and the SQL transformation rules, in your signatures object are used not only by the signature rules, but also by the positive security checks configured in the application firewall profile that is using the signatures object.

The application firewall examines the traffic to your protected web sites and web services to detect traffic that matches a signature. A match is triggered only when every pattern in the rule matches the traffic. When a match occurs, the specified actions for the rule are invoked. You can display an error page or error object when a request is blocked. Log messages can help you to identify attacks being launched against your application. If you enable statistics, the application firewall maintains data about requests that match an application firewall signature or security check.

If the traffic matches both a signature and a positive security check, the more restrictive of the two actions is enforced. For example, if a request matches a signature rule for which the block action is disabled, but the request also matches an SQL Injection positive security check for which the action is block, the request is blocked. In this case, the signature violation might be logged as <not blocked>, although the request is blocked by the SQL injection check.

Customization: If necessary, you can add your own rules to a signatures object. You can also customize the SQL/XSS patterns. The option to add your own signature rules, based on the specific security needs of your applications, gives you the flexibility to design your own customized security solutions. You block only what you don't want and allow the rest. A specific fast-match pattern in a specified location can significantly reduce processing overhead to optimize performance. You can add, modify, or remove SQL injection and cross-site scripting patterns. Built-in RegEx and expression editors help you configure your patterns and verify their accuracy.

Auto-update: You can manually update the signature object to get the latest signature rules, or you can leverage the auto-update feature so that the application firewall can automatically update the signatures from the cloud-based application firewall updates service.

> **Note**
>
> If new signature rules are added during auto-update, they are disabled by default. You should periodically review the updated signatures and enable the newly added rules that are pertinent for protecting your applications.

Getting Started

Using Citrix signatures to protect your application is quite easy and can be accomplished in a few simple steps:

1. Add a signature object.

- You can use the Wizard that prompts you to create the entire application firewall configuration, including adding the profile and policy, selecting and enabling signatures, and specifying actions for signatures and positive security checks. The signatures object is created automatically.
- You can create a copy of the signatures object from the *Default Signatures template, use a blank template to create a new signature with your own customized rules, or add an external format signature. Enable the rules and configure the actions that you want to apply.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

2. Configure the target application firewall profile to use this signatures object.

3. Send traffic to validate the functionality

Highlights:

- The *Default signatures object is a template. It cannot be edited or deleted. To use it, you must create a copy. In your own copy, you can enable the rules and the desired action for each rule as required for your application. To protect the application, you must configure the target profile to use this signature.
- Processing signature patterns has overhead. Try to enable only those signatures that are applicable for protecting your application, rather than enabling all signature rules.
- Every pattern in the rule must match to trigger a signature match.
- You can add your own customized rules to inspect incoming requests to detect various types of attacks, such as SQL injection or cross-site scripting attacks. You can also add rules to inspect the responses to detect and block leakage of sensitive information such as credit card numbers.
- You can make a copy of an existing signatures object and tweak it, by adding or editing rules and SQL/XSS patterns, to protect another application.
- You can use auto-update to download the latest version of the application firewall default rules without need for ongoing monitoring to check for the availability of the new update.
- A signature object can be used by more than one profile. Even after you have configured one or more profile(s) to use a signature object, you can still enable or disable signatures or change the action settings. You can manually create and modify your own custom signature rules. The changes will apply to all the profiles that are currently configured to use this signature object.
- You can configure signatures to detect violations in various types of payloads, such as HTML, XML, JSON, and GWT.
- You can export a configured signatures object and import it to another NetScaler appliance for easy replication of your customized signature rules.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Manually Configuring the Signatures Feature

Dec 07, 2015

To use signatures to protect your web sites, you must review the rules, and enable and configure the ones that you want to apply. The rules are disabled by default. Citrix recommends that you enable all rules that are applicable to the type of content that your web site uses.

To manually configure the signatures feature, use a browser to connect to the configuration utility. Then, create a signatures object from a built-in template, an existing signatures object, or by importing a file. Next, configure the new signatures object as explained in Configuring or Modifying a Signatures Object.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding or Removing a Signatures Object

Oct 06, 2014

You can add a new signatures object to the application firewall by:

- Copying a built-in template.
- Copying an existing signatures object.
- Importing a signatures object from an external file.

You must use the configuration utility to copy a template or existing signatures object. You can use either the configuration utility or the command line to import a signatures object. You can also use either the configuration utility or the command line to remove a signatures object.

To create a signatures object from a template

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to use as a template.
   Your choices are:
   - *** Default Signatures.** Contains the signatures rules, the SQL injection rules, and the cross-site scripting rules.
   - *** XPath Injection.** Contains the XPath injection patterns.
   - **Any existing signatures object.**

   Attention: If you do not choose a signatures type to use as a template, the application firewall will prompt you to create signatures from scratch.
3. Click Add.
4. In the Add Signatures Object dialog box, type a name for your new signatures object, and then click OK. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols.
5. Click Close.

To create a signatures object by importing a file

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, click Add.
3. In the Add Signatures Object dialog box, select the format of the signatures you want to import.
   - To import a NetScaler format signatures file, select the Native Format tab.
   - To import an external signatures format file, select the External Format tab.
4. Choose the file that you want to use to create your new signatures object.
   - To import a native NetScaler format signatures file, in the Import section select either Import from Local File or Import from URL, then type or browse to the path or URL to the file.
   - To import a Cenzic, IBM AppScan, Qualys, or Whitehat format file, in the XSLT section select Use Built-in XSLT File, Use Local File, or Reference from URL. Next, if you chose Use Built-in XSLT File, select the appropriate file format from the drop-down list. If you chose Use Local File or Reference from URL, then type or browse to the path or URL to the file.
5. Click Add, and then click Close.

To create a signatures object by importing a file by using the command line

At the command prompt, type the following commands:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- import appfw signatures <src> <name> [-xslt <string>] [-comment <string>] [-overwrite] [-merge] [-sha1 <string>]
- save ns config

Example #1

The following example creates a new signatures object from a file named signatures.xml and assigns it the name MySignatures.

```
import appfw signatures signatures.xml MySignatures
save ns config
```

To remove a signatures object by using the configuration utility

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to remove.
3. Click Remove.

To remove a signatures object by using the command line

At the command prompt, type the following commands:

- rm appfw signatures <name>
- save ns config

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring or Modifying a Signatures Object

Oct 06, 2014

You configure a signatures object after creating it, or modify an existing signatures object, to enable or disable signature categories or specific signatures, and configure how the application firewall responds when a signature matches a connection.

To configure or modify a signatures object

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to configure, and then click Open.
3. In the Modify Signatures Object dialog box, set the Display Filter Criteria options at the left to display the filter items that you want to configure.
   As you modify these options, the results that you requested are displayed in the Filtered Results window at the right.

   - To display only selected categories of signatures, check or clear the appropriate signature-category check boxes. The signature categories are:

Name	Type of Attack that this Signature Protects Against
cgi	CGI scripts. Includes Perl and UNIX shell scripts.
client	Browsers and other clients.
coldfusion	Web sites that use the Adobe Systems ColdFusion application server.
frontpage	Web sites that use Microsoft's FrontPage server.
iis	Web sites that use the Microsoft Internet Information Server (IIS).
misc	Miscellaneous attacks.
php	Web sites that use PHP
web-activex	Web sites that contain ActiveX controls.
web-struts	Web sites that contain Apache struts, which are java-ee based applets.

   - To display only signatures that have specific check actions enabled, select the ON check box for each of those actions, clear the ON check boxes for the other actions, and clear all of the OFF check boxes. To display only signatures that have a specific check action disabled, select their respective OFF check boxes and clear all of the ON check boxes. To display signatures regardless of whether they have a check action enabled or disabled, select or clear both the ON and the OFF check boxes for that action. The check actions are:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Criterion	Description
Enabled	The signature is enabled. The application firewall checks only for signatures that are enabled when it processes traffic.
Block	Connections that match this signature are blocked.
Log	A log entry is produced for any connection that matches this signature.
Stats	The application firewall includes any connection that matches this signature in the statistics that it generates for that check.

- To display only signatures that contain a specific string, type the string into the text box under the filter criteria, and then click Search.
- To reset all display filter criteria to the default settings and display all signatures, click Show All.

4. For information about a specific signature, select the signature, and then click the blue double arrow in the More field. The Signature Rule Vulnerability Detail message box appears. It contains information about the purpose of the signature and provides links to external web-based information about the vulnerability or vulnerabilities that this signature addresses. To access an external link, click the blue double arrow to the left of the description of that link.
5. Configure the settings for a signature by selecting the appropriate check boxes.
6. If you want to add a local signature rule to the signatures object, or modify an existing local signature rule, see "The Signatures Editor."
7. If you have no need for SQL injection, cross-site scripting, or Xpath injection patterns, click OK, and then click Close. Otherwise, in the lower left-hand corner of the details pane, click Manage SQL/XSS Patterns.
8. In the Manage SQL/XSS Patterns dialog box, Filtered Results window, navigate to the pattern category and pattern that you want to configure. For information about the SQL injection patterns, see "HTML SQL Injection Check." For information about the cross-site scripting patterns, see "HTML Cross-Site Scripting Check."
9. To add a new pattern:
   1. Select the branch to which you want to add the new pattern.
   2. Click the Add button directly below the lower section of the Filtered Results window.
   3. In the Create Signature Item dialog box, fill in the Element text box with the pattern that you want to add. If you are adding a transformation pattern to the transform rules branch, under Elements, fill in the From text box with the pattern that you want to change and the To text box with the pattern to which you want to change the previous pattern.
   4. Click OK.
10. To modify an existing pattern:
    1. In the Filtered Results window, select the branch that contains the pattern that you want to modify.
    2. In the detail window beneath the Filtered Results window, select the pattern that you want to modify.
    3. Click Modify.
    4. In the Modify Signature Item dialog box, Element text box, modify the pattern. If you are modifying a transformation pattern, you can modify either or both patterns under Elements, in the From and the To text boxes.
    5. Click OK.
11. To remove a pattern, select the pattern that you want to remove, then click the Remove button below the details pane beneath the Filtered Results window. When prompted, confirm your choice by clicking Close.
12. To add the patterns category to the XSS branch:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Select the branch to which you want to add the patterns category.
2. Click the Add button directly below the Filtered Results window.
   Note: Currently you can add only one category, named patterns, to the XSS branch, so after you click Add, you must accept the default choice, which is patterns.
3. Click OK.

13. To remove a branch, select that branch, and then click the Remove button directly below the Filtered Results window. When prompted, confirm your choice by clicking OK.
    Note: If you remove a default branch, you remove all of the patterns in that branch. Doing so can disable the security checks that use that information.
14. When you are finished modifying the SQL injection, cross-site scripting, and XPath injection patterns, click OK, and then click Close to return to the Modify Signatures Object dialog box.
15. Click OK at any point to save your changes, and when you are finished configuring the signatures object, click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Protecting JSON Applications using Signatures

Aug 27, 2015

JavaScript Object Notation (JSON) is a text-based open standard derived from the JavaScript scripting language. JSON is preferred for human readable representation of simple data structures and associative arrays, called objects. It serves as an alternative to XML and is primarily used to transmit serialized data structures for communicating with web applications. The JSON files are typically saved with a .json extension.

The JSON payload is typically sent with the MIME type specified as **application/json**. The other "standard" content types for JSON are:

- **application/x-javascript**
- **text/javascript**
- **text/x-javascript**
- **text/x-json**

Using the Citrix Application Firewall Signatures to Protect JSON Applications

To allow JSON requests, the appliance is preconfigured with the JSON content type as shown in the following show-command output:

```
> sh appfw jsonContentType
1)    JSONContenttypevalue:  "^application/json$" IsRegex:  REGEX
Done
```

The Citrix application firewall processes the post body for the following content-types only:

- **application/x-www-form-urlencoded**
- **multipart/form-data**
- **text/x-gwt-rpc**

The requests that are received with other content-type headers including application/json (or any other allowed content type) are forwarded to the backend after header inspection. The post body in such requests is not inspected for security check violations even when the profile's security checks such as SQL or XSS are enabled.
In order to protect JSON applications and detect violations, application firewall signatures can be used. All requests that contain the allowed content-type header are processed by the application firewall for signature match. You can add your own customized signature rules to process JSON payload to perform various security check inspections (for example, XSS, SQL, and Field Consistency), to detect violations in the headers as well as the post body, and take specified actions.

Tip

Unlike the other built-in defaults, the preconfigured JSON content type can be edited or removed by using the CLI or the configuration utility (GUI). If legitimate requests for JSON applications are getting blocked and triggering content-type violations, check to make sure that the content type value is configured accurately. For additional details regarding how application firewall processes content-type header, see http://docs.citrix.com/en-us/netscaler/11/security/application-firewall/content-type-protection.html

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To add or remove JSON content-type by using the command line interface

At the command prompt, type one of the following commands:

add appfw jsonContentType ^application/json$ IsRegEx REGEX

rm appfw JSONContentType "^application/json$"

To managing JSON content types by using the configuration utility

Navigate to **Security** > **Application Firewall** and, in the **Settings** section, select **Manage JSON Content Types**.

In the **Configure Application Firewall JSON Content Type** panel, add, edit, or delete JSON content types to suit the needs of your applications.

Configuring Signature Protection to Detect Attacks in JSON Payload

In addition to a valid JSON content type, you need to configure signatures to specify the pattern(s) that, when detected in a JSON request, indicate a security breach. The specified actions, such as block and log, are taken when an incoming request triggers a match for all the target patterns in the signature rule.

To add a customized signature rule, Citrix recommends that you use the configuration utility. Navigate to **System** > **Security** > **Application Firewall** > **Signatures**. Double click the target signature object to access the **Edit Application Firewall Signatures** panel. Click on the **Add** button to configure the actions, category, log string, rule patterns and so on. Although application firewall inspects all allowed content-type payload for signature match, you can optimize the processing by specifying the JSON expression in the rule. When you **Add** a new rule pattern, select **Expression** in the drop-down options for **Match** and provide the target match expression from your JSON payload to identify the specific requests that need to be inspected. An expression must begin with a **TEXT.** prefix. You can add other rule patterns to specify additional match patterns to identify the attack.

The following example shows a signature rule. If any cross-site script tag is detected in the POST body of the JSON payload that matches the specified XPATH_JSON expression, a signature match is triggered.

Example of a signature to detect XSS in JSON payload

```
<SignatureRule actions="log,stats" category="JSON" enabled="ON" id="1000001" severity="" source="" type="" version="1">

 <PatternList>

  <RequestPatterns>

   <Pattern>

    <Location area="HTTP_POST_BODY"/>

    <Match type="Expression">TEXT.XPATH_JSON(xp%/glossary/title%).CONTAINS("example glossary")</Match>

   </Pattern>

   <Pattern>

    <Location area="HTTP_METHOD"/>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
      <Match type="LITERAL">POST</Match>
    </Pattern>
    <Pattern>
      <Location area="HTTP_POST_BODY"/>
      <Match type="CrossSiteScripting"/>
     </Pattern>
   </RequestPatterns>
  </PatternList>
  <LogString>Cross-site scripting violation detected in json payload</LogString>
  <Comment/>
</SignatureRule>
```

Example of the payload

The following payload triggers the signature match, because it includes the cross-site scripting tag **<Gotcha!!>**.

{"glossary": {"title": "example glossary","GlossDiv": {"title": "S","GlossList": {"GlossEntry": {"ID": "SGML","SortAs": "SGML","GlossTerm": "Standard Generalized Markup Language","Acronym": "SGML","Abbrev": "ISO 8879:1986","GlossDef": {"para": "A meta-markup language, used to create markup languages **<Gotcha!!>** such as DocBook.","GlossSeeAlso": ["GML", "XML"]},"GlossSee": "markup"}}}}}

Example of the log message

Aug 21 12:21:42 <local0.info> 10.217.31.239 08/21/2015:23:21:42 GMT ns 0-PPE-1 : APPFW APPFW_SIGNATURE_MATCH 1471 0 : 10.217.253.62 990-PPE0 NtJnVMNnvPeQJnaUzXYW/GTvAQsA010 prof1 http://10.217.31.212/FFC/login_post.php Signature violation rule ID 1000001: cross-site scripting violation detected in json payload <not blocked>

> **Note**
>
> If you send the same payload after removing the cross-site script tag (**<Gotcha!!>**), the signature rule match is not triggered.

Highlights:

- To protect JSON payload, use application firewall signatures to detect XSS, SQL and other violations.
- Verify that the JSON content type is configured on the appliance as the allowed content type.
- Make sure that the content type in the payload matches the configured JSON content type.
- Make sure that all the patterns configured in the signature rule match for the signature violation to be triggered.
- When you add a signature rule, it MUST have at least one Rule pattern to match the Expression in the JSON payload. All the PI expressions in signature rules must start with the prefix TEXT. and must be Boolean.

Code COPY

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updating a Signatures Object

Jul 21, 2015

You should update your signatures objects frequently to ensure that your application firewall is providing protection against current threats. You should regularly update both the default application firewall signatures and any signatures that you import from a supported vulnerability scanning tool.

Citrix regularly updates the default signatures for the application firewall. You can update the default signatures manually or automatically. In either case, ask your Citrix representative or Citrix reseller for the URL to access the updates. You can enable automatic updates of the Citrix native format signatures in the "Engine Settings" and "Signature Auto Update Settings" dialog boxes.

Most makers of vulnerability scanning tools regularly update the tools. Most web sites also change frequently. You should update your tool and rescan your web sites regularly, exporting the resulting signatures to a file and importing them into your application firewall configuration.

Note: When you update the application firewall signatures from the NetScaler command line, you must first update the default signatures, and then issue additional update commands to update each custom signatures file that is based on the default signatures. If you do not update the default signatures first, a version mismatch error prevents updating of the custom signatures files.

To update the application firewall signatures from the source by using the command line

At the command prompt, type the following commands:

- update appfw signatures <name> [-mergedefault]
- save ns config

Example

The following example updates the signatures object named MySignatures from the default signatures object, merging new signatures in the default signatures object with the existing signatures. This command does not overwrite any user-created signatures or signatures imported from another source, such as an approved vulnerability scanning tool.

```
update appfw signatures MySignatures -mergedefault
save ns config
```

Updating a Signatures Object from a Citrix Format File

Updated: 2014-10-06

Citrix regularly updates the signatures for the Application Firewall. You should regularly update the signatures on your Application Firewall to ensure that your Application Firewall is using the most current list. Ask your Citrix representative or Citrix reseller for the URL to access the updates.

To update a signatures object from a Citrix format file by using the command line

At the command prompt, type the following commands:

- update appfw signatures <name> [-mergeDefault]
- save ns config

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To update a signatures object from a Citrix format file by using the configuration utility

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to update.
3. In the Action drop-down list, select Merge.
4. In the Update Signatures Object dialog box, choose one of the following options.
   - **Import from URL**—Choose this option if you download signature updates from a web URL.
   - **Import from Local File**—Choose this option if you import signature updates from a file on your local hard drive, network hard drive, or other storage device.
5. In the text area, type the URL, or type or browse to the local file.
6. Click Update. The update file is imported, and the Update Signatures dialog box changes to a format nearly identical to that of the Modify Signatures Object dialog box. The Update Signatures Object dialog box displays all branches with new or modified signature rules, SQL injection or cross-site scripting patterns, and XPath injection patterns if there are any.
7. Review and configure the new and modified signatures.
8. When you are finished, click OK, and then click Close.

Updating a Signatures Object from a Supported Vulnerability Scanning Tool

Updated: 2014-01-17

Note: Before you update a signatures object from a file, you must create the file by exporting signatures from the vulnerability scanning tool.

To import and update signatures from a vulnerability scanning tool

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to update, and then click Merge.
3. In the Update Signatures Object dialog box, on the External Format tab, Import section, choose one of the following options.
   - **Import from URL**—Choose this option if you download signature updates from a Web URL.
   - **Import from Local File**—Choose this option if you import signature updates from a file on your local or a network hard drive or other storage device.
4. In the text area, type the URL, or browse or type the path to the local file.
5. In the XSLT section, choose one of the following options.
   - **Use Built-in XSLT File**—Choose this option if you want to use a built-in XSLT files.
   - **Use Local XSLT File**—Choose this option to use an XSLT file on your local computer.
   - **Reference XSLT from URL**—Choose this option to import an XSLT file from a web URL.
6. If you chose Use Built-in XSLT File, in the Built-In XSLT drop-down list select the file that you want to use from the following options:
   - **Cenzic.**
   - **Deep_Security_for_Web_Apps.**
   - **Hewlett_Packard_Enterprise_WebInspect.**
   - **IBM-AppScan-Enterprise.**
   - **IBM-AppScan-Standard.**
   - **Qualys.**
   - **Whitehat.**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

7. Click Update. The update file is imported, and the Update Signatures dialog box changes to a format nearly identical to that of the Modify Signatures Object dialog box, which is described in "Configuring or Modifying a Signatures Object." The Update Signatures Object dialog box displays all branches with new or modified signature rules, SQL injection or cross-site scripting patterns, and XPath injection patterns if there are any.
8. Review and configure the new and modified signatures.
9. When you are finished, click OK, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exporting a Signatures Object to a File

Oct 06, 2014

You export a signatures object to a file so that you can import it to another NetScaler ADC.

To export a signatures object to a file

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to configure.
3. In the Actions drop-down list, select Export.
4. In the Export Signatures Object dialog box, Local File text box, type the path and name of the file to which you want to export the signatures object, or use the Browse dialog to designate a path and name.
5. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The Signatures Editor

Oct 14, 2014

You can use the signatures editor, which is available in the configuration utility, to add a new user-defined (local) signature rule to an existing signatures object, or to modify a previously configured local signature rule. Except that it is defined by the user (you), a local signature rule has the same attributes as a default signature rule from Citrix, and it functions in the same way. You enable or disable it, and configure the signature actions for it, just as you do for a default signature.

Add a local rule if you need to protect your web sites and services from a known attack that the existing signatures do not match. For example, you might discover a new type of attack and determine its characteristics by examining the logs on your web server, or you might obtain third-party information about a new type of attack.

At the heart of a signature rule are the rule *patterns*, which collectively describe the characteristics of the attack that the rule is designed to match. Each pattern can consist of a simple string, a PCRE-format regular expression, or the built-in SQL injection or cross-site scripting patterns.

You might want to modify a signature rule by adding a new pattern or modifying an existing pattern to match an attack. For example, you might find out about changes to an attack, or you might determine a better pattern by examining the logs on your web server, or from third-party information.

To add or modify a local signature rule by using the Signatures Editor

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to edit, and then click Open.
3. In the Modify Signatures Object dialog box, in the middle of the screen beneath the Filtered Results window, do one of the following:.
   - To add a new local signature rule, click Add.
   - To modify an existing local signature rule, select that rule, and then click Open.
4. In the Add Local Signature Rule or the Modify Local Signature Rule dialog box, configure the actions for a signature by selecting the appropriate check boxes.
   - **Enabled.** Enables the new signature rule. If you do not select this, this new signature rule is added to your configuration, but is inactive.
   - **Block.** Blocks connections that violate this signature rule.
   - **Log.** Logs violations of this signature rule to the NetScaler log.
   - **Stat.** Includes violations of this signature rule in the statistics.
   - **Remove.** Strips information that matches the signature rule from the response. (Applies only to response rules.)
   - **X-Out.** Masks information that matches the signature rule with the letter X. (Applies only to response rules.)
   - **Allow Duplicates.** Allows duplicates of this signature rule in this signatures object.
5. Choose a category for the new signature rule from the Category drop-down list.
   You can also create a new category by clicking the icon to the right of the list and using the Add Signature Rule Category dialog box to add a new category to the list, The rule you are modifying is automatically added to the new category. For instructions, see "To add a signature rule category."
6. In the **LogString** text box, type a brief description of the signature rule to be used in the logs.
7. In the **Comment** text box, type a comment. (Optional)
8. Click More..., and modify the advanced options.
   1. To strip HTML comments before applying this signature rule, in the Strip Comments drop-down list choose All or

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exclude Script Tag.

2. To set CSRF Referer Header checking, in the CSRF Referer Header checking radio button array, select either the If Present or Always radio button.
3. To manually modify the Rule ID assigned to this local signature rule, modify the number in the Rule ID text box. The ID must be a positive integer between 1000000 and 1999999 that has not already been assigned to a local signature rule.
4. To assign a version number to the new signature rule, modify the number in the Version Number text box.
5. To assign a Source ID, modify the string in the Source ID text box.
6. To specify the source, choose Local or Snort from the Source drop-down list, or click the Add icon to the right of the list and add a new source.
7. To assign a harm score to violations of this local signature rule, type a number between 1 and 10 in the Harm Score text box.
8. To assign a severity rating to this local signature rule, in the Severity drop-down list choose High, Medium, or Low, or click the Add icon to the right of the list and add a new severity rating.
9. To assign a violation type to this local signature rule, in the Type drop-down list choose Vulnerable or Warning, or click the Add icon to the right of the list and add a new violation type.

9. In the **Patterns** list, add or edit a pattern.
   - To add a pattern, click Add. In the Create New Signature Rule Pattern dialog box, add one or more patterns for your signature rule, and then click OK.
   - To edit a pattern, select the pattern, and then click Open. In the Edit Signature Rule Pattern dialog box, modify the pattern, and then click OK.

   For more information about adding or editing patterns, see "Signature Rule Patterns."
10. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To add a signature rule category

Sep 03, 2013

Putting signature rules into a category enables you to configure the actions for a group of signatures instead of for each individual signature. You might want to do so for the following reasons:

- **Ease of selection.** For example, assume that all of signature rules in a particular group protect against attacks on a specific type of web server software or technology. If your protected web sites use that software or technology, you want to enable them all. If they do not, you do not want to enable any of them.
- **Ease of initial configuration.** It is easiest to set defaults for a group of signatures as a category, instead of one-by-one. You can then make any changes to individual signatures as needed.
- **Ease of ongoing configuration.** It is easier to configure signatures if you can display only those that meet specific criteria, such as belonging to a specific category.

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select that signatures object that you want to configure, and then click Open.
3. In the Modify Signatures Object dialog box, in the middle of the screen, beneath the Filtered Results window, click Add.
4. In the Add Local Signature Rule dialog box, click the icon to the right of the Category drop-down list.
5. In the Add Signature Rule Category dialog box, New Category text box, type a name for your new signature category. The name can consist of from one to 64 characters.
6. Click **OK**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Signature Rule Patterns

Aug 20, 2014

You can add a new pattern to a signature rule or modify an existing pattern of a signature rule to specify a string or expression that characterizes an aspect of the attack that the signature matches. To determine which patterns an attack exhibits, you can examine the logs on your web server, use a tool to observe connection data in real time, or obtain the string or expression from a third-party report about the attack.

Caution: Any new pattern that you add to a signature rule is in an AND relationship with the existing patterns. Do not add a new pattern to an existing signature rule if you do not want a potential attack to have to match all of the patterns in order to match the signature.

Each pattern can consist of a simple string, a PCRE-format regular expression, or the built-in SQL injection or cross-site scripting pattern. Before you attempt to add a pattern that is based on a regular expression, you should make sure that you understand PCRE-format regular expressions. PCRE expressions are complex and powerful; if you do not understand how they work, you can unintentionally create a pattern that matches something that you did not want (a *false positive*) or that fails to match something that you did want (a *false negative*).

If you are not already familiar with PCRE-format regular expressions, you can use the following resources to learn the basics, or for help with some specific issue:

- *— "Mastering Regular Expressions"*
  , Third Edition. Copyright (c) 2006 by Jeffrey Friedl. O'Reilly Media, ISBN: 9780596528126
- *— "Regular Expressions Cookbook"*
  . Copyright (c) 2009 by Jan Goyvaerts and Steven Levithan. O'Reilly Media, ISBN: 9780596520687
- **PCRE Man page/Specification** (text/official): "http://www.pcre.org/pcre.txt"
- **PCRE Man Page/Specification** (html/gammon.edu.au): "http://www.gammon.com.au/pcre/index.html"
- **Wikipedia PCRE entry**: "http://en.wikipedia.org/wiki/PCRE"
- **PCRE Mailing List** (run by exim.org): "http://lists.exim.org/mailman/listinfo/pcre-dev"

If you need to encode non-ASCII characters in a PCRE-format regular expression, the NetScaler platform supports encoding of hexadecimal UTF-8 codes. For more information, see "[PCRE Character Encoding Format]."

To configure a signature rule pattern

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select that signatures object that you want to configure, and then click **Open**.
3. In the Modify Signatures Object dialog box, in the middle of the screen beneath the Filtered Results window, either click Add to create a signature rule, or select an existing signature rule and click Open.
   Note: You can modify only signature rules that you added. You cannot modify the default signature rules.
   Depending on your action, either the Add Local Signature Rule or the Modify Local Signature Rule dialog box appears. Both dialog boxes have the same contents.
4. Under the Patterns window in the dialog box, either click Add to add a new pattern, or select an existing pattern from the list beneath the Add button and click Open. Depending on your action, either the Create New Signature Rule Pattern or the Edit Signature Rule Pattern dialog box appears. Both dialog boxes have the same contents.
5. In the Pattern Type drop-down list, choose the type of connection that the pattern is intended to match.
   - If the pattern is intended to match request elements or features, such as injected SQL code, attacks on web forms, cross-site scripts, or inappropriate URLs, choose **Request**.
   - If the pattern is intended to match response elements or features, such as credit card numbers or safe objects,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

choose **Response**.

6. In the Location area, define the elements to examine with this pattern.
   The Location area describes what elements of the HTTP request or response to examine for this pattern. The choices that appear in the Location area depend upon the chosen pattern type. If you chose Request as the pattern type, items relevant to HTTP requests appear; if you chose Response, items relevant to HTTP responses appear.

   In addition, as you choose a value from the Area drop-down list, the remaining parts of the Location area change interactively. Following are all configuration items that might appear in this section.

   Area

   Drop-down list of elements that describe a particular portion of the HTTP connection. The choices are as follows:
   - **HTTP_ANY**. All parts of the HTTP connection.
   - **HTTP_COOKIE**. All cookies in the HTTP request headers after any cookie transformations are performed.
     Note: Does not search HTTP response "Set-Cookie:" headers.
   - **HTTP_FORM_FIELD**. Form fields and their contents, after URL decoding, percent decoding, and removal of excess whitespace. You can use the <Location> tag to further restrict the list of form field names to be searched.
   - **HTTP_HEADER**. The value portions of the HTTP header after any cross-site scripting or URL decoding transformations.
   - **HTTP_METHOD**. The HTTP request method.
   - **HTTP_ORIGIN_URL**. The origin URL of a web form.
   - **HTTP_POST_BODY**. The HTTP post body and the web form data that it contains.
   - **HTTP_RAW_COOKIE**. All HTTP request cookie, including the "Cookie:" name portion.
     Note: Does not search HTTP response "Set-Cookie:" headers.
   - **HTTP_RAW_HEADER**. The entire HTTP header, with individual headers separated by linefeed characters (\n) or carriage return/line-feed strings (\r\n).
   - **HTTP_RAW_RESP_HEADER**. The entire response header, including the name and value parts of the response header after URL transformation has been done, and the complete response status. As with HTTP_RAW_HEADER, individual headers are separated by linefeed characters (\n) or carriage return/line-feed strings (\r\n).
   - **HTTP_RAW_SET_COOKIE**. The entire Set-Cookie header after any URL transformations have been performed.
     Note: URL transformation can change both the domain and path parts of the Set-Cookie header.
   - **HTTP_RAW_URL**. The entire request URL before any URL transformations are performed, including any query or fragment parts.
   - **HTTP_RESP_HEADER**. The value part of the complete response headers after any URL transformations have been performed.
   - **HTTP_RESP_BODY**. The HTTP response body.
   - **HTTP_SET_COOKIE**. All "Set-Cookie" headers in the HTTP response headers.
   - **HTTP_STATUS_CODE**. The HTTP status code.
   - **HTTP_STATUS_MESSAGE**. The HTTP status message.
   - **HTTP_URL**. The value portion of the URL in the HTTP headers, excluding any query or fragment ports, after conversion to the UTF-* character set, URL decoding, stripping of whitespace, and conversion of relative URLs to absolute. Does not include HTML entity decoding.

   URL

   Examines any URLs found in the elements specified by the Area setting. Select one of the following settingss.
   - **Any.** Checks all URLs.
   - Literal. Checks URLs that contain a literal string. After you select Literal, a text box is displayed. Type the literal string that you want in the text box.
   - PCRE. Checks URLs that match a PCRE-format regular expression. After you select this choice, the regular expression

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

window is displayed. Type the regular expression in the window. You can use the **Regex Tokens** to insert common regular expression elements at the cursor, or you can click Regex Editor to display the Regular Expression Editor dialog box, which provides more assistance in constructing the regular expression that you want.

- Expression. Checks URLs that match a NetScaler default expression.

Field Name

Examines any form field names found in the elements specified by the Area selection.

- **Any.** Checks all URLs.
- Literal. Checks URLs that contain a literal string. After you select Literal, a text box is displayed. Type the literal string that you want in the text box.
- PCRE. Checks URLs that match a PCRE-format regular expression. After you select this choice, the regular expression window is displayed. Type the regular expression in the window. You can use the **Regex Tokens** to insert common regular expression elements at the cursor, or you can click Regex Editor to display the Regular Expression Editor dialog box, which provides more assistance in constructing the regular expression that you want.
- Expression. Checks URLs that match a NetScaler default expression.

7. In the Pattern area, define the pattern. A pattern is a literal string or PCRE-format regular expression that defines the pattern that you want to match. The Pattern area contains the following elements:

   Match

   A drop-down list of search methods that you can use for the signature. This list differs depending on whether the pattern type is Request or Response.

   Request Match Types

   - **Literal**. A literal string.
   - **PCRE**. A PCRE-format regular expression.
     NOTE: When you choose PCRE, the regular expression tools beneath the Pattern window are enabled. These tools are not useful for most other types of patterns.
   - **Injection**. Directs the application firewall to look for injected SQL in the specified location. The Pattern window disappears, because the application firewall already has the patterns for SQL injection.
   - **CrossSiteScripting**. Directs the application firewall to look for cross-site scripts in the specified location. The Pattern window disappears, because the application firewall already has the patterns for cross-site scripts.
   - **Expression**. An expression in the NetScaler default expressions language. This is the same expressions language that is used to create application firewall policies and other policies on the NetScaler appliance. Although the NetScaler expressions language was originally developed for policy rules, it is a highly flexible general purpose language that can also be used to define a signature pattern.
     When you choose Expression, the NetScaler Expression Editor appears beneath Pattern window. For more information about the Expression Editor and instructions on how to use it, see "To add a firewall rule (expression) by using the Add Expression dialog box." For more information about NetScaler expressions, see "Policies and Expressions."

   Response Match Types

   - **Literal**. A literal string.
   - **PCRE**. A PCRE-format regular expression.
     NOTE: When you choose PCRE, the regular expression tools beneath the Pattern window are enabled. These tools are not useful for most other types of patterns.
   - **Credit Card**. A built-in pattern to match one of the six supported types of credit card number.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: The Expression match type is not available for Response-side signatures.

Pattern Window (unlabeled)

In this window, type the pattern that you want to match, and fill in any additional data.

- **Literal.** Type the string you want to search for in the text area.
- **PCRE.** Type the regular expression in the text area. Use the **Regex Editor** for more assistance in constructing the regular expression that you want, or the Regex Tokens to insert common regular expression elements at the cursor. To enable UTF-8 characters, click UTF-8.
- **Expression.** Type the NetScaler advanced expression in the text area. Use Prefix to choose the first term in your expression, or Operator to insert common operators at the cursor. Click **Add** to open the Add Expression dialog box for more assistance in constructing the regular expression that you want. Click Evaluate to open the Advanced Expression Evaluator to help determine what effect your expression has.
- **Offset.** The number of characters to skip over before starting to match on this pattern. You use this field to start examining a string at some point other than the first character.
- **Depth.** How many characters from the starting point to examine for matches. You use this field to limit searches of a large string to a specific number of characters.
- **Min-Length.** The string to be searched must be at least the specified number of bytes in length. Shorter strings are not matched.
- **Max-Length.** The string to be searched must be no longer than the specified number of bytes in length. Longer strings are not matched.
- **Search method.** A check box labeled fastmatch. You can enable fastmatch only for a literal pattern, to improve performance.

8. Click OK.
9. Repeat the previous four steps to add or modify additional patterns.
10. When finished adding or modifying patterns, click OK to save your changes and return to the Signatures pane.
    Caution: Until you click **OK** in the **Add Local Signature Rule** or **Modify Local Signature Rule** dialog box, your changes are not saved. Do not close either of these dialog boxes without clicking **OK** unless you want to discard your changes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Signature Updates in High-Availability Deployment and Build Upgrades

Feb 01, 2017

The signature update occurs on the primary node. While the signatures are updated on the primary node, in parallel the updated files are simultaneously synchronized with the secondary node.

The *Default signature is always updated first and then the rest of the user-defined signatures are updated.

Connecting to Amazon AWS

The default route NSIP is used to connect to the Amazon AWS. If there is a specific use case scenario where SNIP is used, and if there are multiple SNIPs, the first one to receive the ARP response from the hosting site will hold the route.

Signature updates during version upgrades

In case of an upgrade, if the NS has an older base version for the signatures, *Default signature is automatically updated if a newer signature version is available.

If the schema has changed, the schema version of all the signature objects gets updated when the version is upgraded.

However, for the base version of the user-defined signatures, the behavior is different in release 10.5 versus release 11.0.

In release 10.5, only the default signature was updated and the base version of the rest of the signatures remained unchanged after the build upgrade.

In release 11.0, this behavior has changed. When the appliance is upgraded to install a new build, not only the *Default signature object but all the other user-defined signatures that currently exist in the appliance are also updated and will have the same version after the build upgrade.

In both 10.5 and 11.0 release builds, if auto-update is configured, the *Default Signatures as well as all non-zero version signatures get auto-updated to the latest released signature version and will have the same base version.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Overview of Security checks

Sep 03, 2013

The application firewall advanced protections (security checks) are a set of filters designed to catch complex or unknown attacks on your protected web sites and web services. The security checks use heuristics, positive security, and other techniques to detect attacks that may not be detected by signatures alone. You configure the security checks by creating and configuring an application firewall profile, which is a collection of user-defined settings that tell the application firewall which security checks to use and how to handle a request or response that fails a security check. A profile is associated with a signatures object and with a policy to create a security configuration.

The application firewall provides twenty security checks, which differ widely in the types of attacks that they target and how complex they are to configure. The security checks are organized into the following categories:

- **Common security checks.** Checks that apply to any aspect of web security that either does not involve content or is equally applicable to all types of content.
- **HTML security checks.** Checks that examine HTML requests and responses. These checks apply to HTML-based web sites and to the HTML portions of Web 2.0 sites, which contain mixed HTML and XML content.
- **XML security checks.** Checks that examine XML requests and responses. These checks apply to XML-based web services and to the XML portions of Web 2.0 sites.

The security checks protect against a wide range of types of attack, including attacks on operation system and web server software vulnerabilities, SQL database vulnerabilities, errors in the design and coding of web sites and web services, and failures to secure sites that host or can access sensitive information.

All security checks have a set of configuration options, the check actions, which control how the application firewall handles a connection that matches a check. Three check actions are available for all security checks. They are:

- **Block.** Block connections that match the signature. Disabled by default.
- **Log.** Log connections that match the signature, for later analysis. Enabled by default.
- **Stats.** Maintain statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.

A fourth check action, **Learn**, is available for more than half of the check actions. It observes traffic to a protected Web site or web service and uses connections that repeatedly violate the security check to generate recommended exceptions (relaxations) to the check, or new rules for the check. In addition to the check actions, certain security checks have parameters that control the rules that the check uses to determine which connections violate that check, or that configure the application firewall's response to connections that violate the check. These parameters are different for each check, and they are described in the documentation for each check.

To configure security checks, you can use the application firewall wizard, as described in "The Application Firewall Wizard," or you can configure the security checks manually, as described in "Manual Configuration By Using the Configuration Utility." Some tasks, such as manually entering relaxations or rules or reviewing learned data, can be done only by using the configuration utility, not the command line. Using the wizard is usually best configuration method, but in some cases manual configuration might be easier if you are thoroughly familiar with it and simply want to adjust the configuration for a single security check.

Regardless of which method you use to configure the security checks, each security check requires that certain tasks be performed. Many checks require that you specify exceptions (relaxations) to prevent blocking of legitimate traffic before

© 1999-2017 Citrix Systems, Inc. All rights reserved.

you enable blocking for that security check. You can do this manually, by observing the log entries after a certain amount of traffic has been filtered and then creating the necessary exceptions. However, it is usually much easier to enable the learning feature and let it observe the traffic and recommend the necessary exceptions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Top-Level Protections

Oct 12, 2015

Four of the application firewall protections are especially effective against common types of Web attacks, and are therefore more commonly used than any of the others. They are:

- **HTML Cross-Site Scripting.** Examines requests and responses for scripts that attempt to access or modify content on a different Web site than the one on which the script is located. When this check finds such a script, it either renders the script harmless before forwarding the request or response to its destination, or it blocks the connection.

- **HTML SQL Injection.** Examines requests that contain form field data for attempts to inject SQL commands into an SQL database. When this check detects injected SQL code, it either blocks the request or renders the injected SQL code harmless before forwarding the request to the Web server.

  Note: If both of the following conditions apply to your configuration, you should make certain that your Application Firewall is correctly configured:
  - If you enable the HTML Cross-Site Scripting check or the HTML SQL Injection check (or both), and
  - Your protected Web sites accept file uploads or contain Web forms that can contain large POST body data.

  For more information about configuring the Application Firewall to handle this case, see "Configuring the Application Firewall."

- **Buffer Overflow.** Examines requests to detect attempts to cause a buffer overflow on the Web server.
- **Cookie Consistency.** Examines cookies returned with user requests to verify that they match the cookies your Web server set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the Web server.

The Buffer Overflow check is simple; you can usually enable blocking for it immediately. The other three top-level checks are considerably more complex and require configuration before you can safely use them to block traffic. Citrix strongly recommends that, rather than attempting to configure these checks manually, you enable the learning feature and allow it to generate the necessary exceptions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTML Cross-Site Scripting Check

Mar 12, 2018

The HTML Cross-Site Scripting (XSS) check examines both the headers and the POST bodies of user requests for possible cross-site scripting attacks. If it finds a cross-site script, it either modifies (*transforms*) the request to render the attack harmless, or blocks the request.

To prevent misuse of the scripts on your protected web sites to breach security on your web sites, the HTML Cross-Site Scripting check blocks scripts that violate the *same origin rule*, which states that scripts should not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called cross-site scripting. The reason cross-site scripting is a security issue is that a web server that allows cross-site scripting can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

Unfortunately, many companies have a large installed base of JavaScript-enhanced web content that violates the same origin rule. If you enable the HTML Cross-Site Scripting check on such a site, you have to generate the appropriate exceptions so that the check does not block legitimate activity.

The application firewall offers various action options for implementing HTML Cross-Site Scripting protection. In addition to the **Block**, **Log**, **Stats** and **Learn** actions, you also have the option to **Transform cross-site scripts** to render an attack harmless by entity encoding the script tags in the submitted request. You can configure **Check complete URL's for cross-site scripting** parameter to specify if you want to inspect not just the query parameters but the entire URL to detect XSS attack.

You can deploy relaxations to avoid false positives. The application firewall learning engine can provide recommendations for configuring relaxation rules.

Following options are available for configuring an optimized HTML Cross-Site Scripting protection for your application:

- **Block**— If you enable block, the block action is triggered if the XSS tags are detected in the request.
- **Log**— If you enable the log feature, the HTML Cross-Site Scripting check generates log messages indicating the actions that it takes. If block is disabled, a separate log message is generated for each header or form field in which the XSS violation was detected. However, only one message is generated when the request is blocked. Similarly, one log message per request is generated for the transform operation, even when XSS tags are transformed in multiple fields. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.
- **Stats**— If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.
- **Learn**— If you are not sure which relaxation rules might be ideally suited for your application, you can use the learn feature to generate HTML Cross-Site Scripting rule recommendations based on the learned data. The application firewall learning engine monitors the traffic and provides learning recommendations based on the observed values. To get optimal benefit without compromising performance, you might want to enable the learn option for a short time to get a representative sample of the rules, and then deploy the rules and disable learning.
- **Transform cross-site scripts**— If enabled, the application firewall makes the following changes to requests that match the HTML Cross-Site Scripting check:
  - Left angle bracket (<) to HTML character entity equivalent (<)
  - Right angle bracket (>) to HTML character entity equivalent (>)

  This ensures that browsers do not interpret unsafe html tags, such as <script>, and thereby execute malicious code. If you enable both request-header checking and transformation, any special characters found in request headers are also modified as described above. If scripts on your protected web site contain cross-site scripting features, but your web site does not rely upon those scripts to operate correctly, you can safely disable blocking and enable transformation. This configuration ensures that no legitimate web traffic is blocked, while stopping any potential cross-site scripting attacks.

- **Check complete URLs for cross-site scripting**— If checking of complete URLs is enabled, the application firewall examines entire URLs for HTML cross-site scripting attacks instead of checking just the query portions of URLs.
- **Check Request headers**— If Request header checking is enabled, the application firewall examines the headers of requests for HTML cross-site scripting attacks, instead of just URLs. If you use the configuration utility, you can enable this parameter in the Settings tab of the application firewall profile.

> **Important**
>
> As part of the streaming changes, the application firewall processing of the Cross-site Scripting tags has changed. This change is applicable to 11.0 builds onwards. This change is also pertinent for the enhancement builds of 10.5.e that support request side streaming. In earlier releases, presence of either open bracket (<), or close bracket (>), or both open and close brackets (<>) was flagged as Cross-site Scripting Violation. The behavior has changed in the builds that include support for request side streaming. Only the close bracket character (>) is no longer considered as an attack. Requests are blocked even when an open bracket character (<) is present, and is considered as an attack. The Cross-site scripting attack gets flagged.

XSS Fine grained Relaxations

The application firewall gives you an option to exempt a specific form field, header, or Cookie from cross-site scripting inspection check. You can completely bypass the inspection for one or more of these fields by configuring relaxation rules.

The application firewall allows you to implement tighter security by fine tuning the relaxation rules. An application might require the flexibility to allow specific patterns, but configuring a relaxation rule to bypass the security inspection might make the application vulnerable to attacks, because the target field is exempted from inspection for any cross-site scripting attack patterns. Cross-site scripting fine grained relaxation provides the option to allow specific attributes, tags, and patterns. The rest of the attributes, tags and patterns are blocked. For example, the application firewall currently has a default set of more than 125 denied patterns. Because hackers can use these patterns in Cross-site script attacks, the application firewall flags them as potential threats. You can relax one or more pattern(s) that are considered safe for the specific location. The rest of the potentially dangerous XSS patterns are still checked for the target location and continue to trigger the security check violations. You now have much tighter control.

The commands used in relaxations have optional parameters for **Value Type** and **Value Expression**. The value type can be left blank or you have an option to select **Tag** or **Attribute** or **Pattern**. If you leave the value type blank, the configured field of the specified URL is exempted from the Cross-Site Scripting check inspection. If you select a value type, you must provide a value expression. You can specify whether the value expression is a regular expression or a literal string. When the input is matched against the allowed and denied list, only the specified expressions configured in the relaxation rules are exempted.

The application firewall has the following XSS built-in lists:

1. **XSS Allowed Attributes**: There are 52 default allowed attributes, such as, **abbr**, **accesskey**, **align**, **alt**, **axis**, **bgcolor**, **border**, **cellpadding**, **cellspacing**, **char**, **charoff**, **charset** etc.
2. **XSS Allowed Tags**: There are 47 default allowed tags, such as, **address**, **basefont**, **bgsound**, **big**, **blockquote**, **bg**, **br**, **caption**, **center**, **cite**, **dd**, **del** etc.
3. **XSS Denied Patterns**: There are 129 default denied patterns, such as, **FSCommand**, **javascript:**, **onAbort**, **onActivate** etc.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Warning**
>
> Application firewall action URL's are regular expressions. When configuring HTML cross-site scripting relaxation rules, you can specify **Name**, and **Value Expression** to be literal or RegEx. Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the rule that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the HTML Cross-Site Scripting check would otherwise have blocked.

Points to Consider:

- Value expression is an optional argument. A field name might not have any value expression.
- A field name can be bound to multiple value expressions.
- Value expressions should be assigned a value type. The XSS value type can be: 1) Tag, 2) Attribute, or 3) Pattern.
- You can have multiple relaxation rules per field name/URL combination
- The form field names and the action URL's are not case sensitive.

Using the Command Line to Configure the HTML Cross-Site Scripting check

To configure HTML Cross-Site Scripting check actions and other parameters by using the command line

If you use the command-line interface, you can enter the following commands to configure the HTML Cross-Site Scripting Check:

- **set appfw profile** <name> **-crossSiteScriptingAction** (([**block**] [**learn**] [**log**] [**stats**]) | [**none**])
- **set appfw profile** <name> **-crossSiteScriptingTransformUnsafeHTML** (**ON** | **OFF**)
- **set appfw profile** <name> **-crossSiteScriptingCheckCompleteURLs** (**ON** | **OFF**)
- **set appfw profile** <name> **-checkRequestHeaders** (**ON** | **OFF**)

To configure a HTML Cross-Site Scripting check relaxation rule by using the command line

Use the bind or unbind command to add or delete binding, as follows:

- **bind appfw profile** <name> **-crossSiteScripting** <String> [**isRegex** (**REGEX** | **NOTREGEX**)] <formActionURL> [**-location** <location>] [**-valueType** (**Tag** | **Attribute** | **Pattern**) [<valueExpression>] [**-isValueRegex** (**REGEX** | **NOTREGEX**)]]
- **unbind appfw profile** <name> **-crossSiteScripting** <String> <formActionURL> [**-location** <location>] [**-valueType** (**Tag** | **Attribute** | **Pattern**) [<valueExpression>]]

Using the Configuration Utility to Configure the HTML Cross-Site Scripting Check

In the configuration utility, you can configure the HTML Cross-Site Scripting check in the pane for the profile associated with your application.

To configure or modify the HTML Cross-Site Scripting check by using the configuration utility

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

   a. If you just want to enable or disable **Block**, **Log**, **Stats**, and **Learn** actions for the HTML Cross-Site Scripting, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the Security Check pane.

   b. If you want to configure additional options for this security check, double click **HTML Cross-Site Scripting**, or select the row and click **Action Settings**, to display the following options:

      Transform cross-site scripts—Transform unsafe script tags.

      Check complete URLs for Cross-site scripting—Instead of checking just the query part of the URL, check the complete URL for cross-site script violations.

   After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

To enable or disable the **Check request Header** setting, in the **Advanced Settings** pane, click **Profile Settings**. In **Common Settings**, Select or clear the **Check Request Headers** check box. Click **OK**. You can either use the X icon at the top right hand side of the Profile Settings pane to close this section or, if you have finished configuring this profile, you can click **Done** to return to the **Application Firewall** > **Profile**.

To configure a HTML Cross-Site Scripting relaxation rule by using the configuration utility

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**.
3. In the Relaxation Rules table, double-click the **HTML Cross-Site Scripting** entry, or select it and click **Edit**.
4. In the **HTML Cross-Site Scripting Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

> **Note**
>
> When you add a new rule, the **Value Expression** field is not displayed unless you select **Tag** or **Attribute** or **Pattern** option in the **Value Type** Field.

To manage HTML Cross-Site Scripting relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **HTML Cross-Site Scripting** row in the Relaxation Rules table, and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the Cross-Site Scripting patterns by using the configuration utility

You can use the configuration utility to view or customize the default list of XSS allowed attributes or allowed tags. You can also view or customize the default list of XSS denied Patterns.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The default lists are specified in **Application Firewall** > **Signatures** > **Default Signatures**. If you do not bind any signature object to your profile, the default XSS allowed and denied list specified in the Default Signatures object will be used by the profile for the Cross-Site Scripting security check processing. The Tags, Attributes, and Patterns, specified in the default signatures object, are read-only. You cannot edit or modify them. If you want to modify or change these, make a copy of the Default Signatures object to create a User-Defined signature object. Make changes in the allowed or denied lists in the new User-defined signature object and use this signature object in your profile that is processing the traffic for which you want to use these customized allowed and denied lists.

For more information about signatures, see the following:

http://docs.citrix.com/en-us/netscaler/11/security/application-firewall/signatures.html.

1. **To view default XSS patterns:**

   a. Navigate to **Application firewall** > **Signatures**, select ***Default Signatures**, and click **Edit**. Then click **Manage SQL/XSS Patterns**.

   The M**anage SQL/XSS Paths** table shows following three rows pertaining to XSS :

   xss/allowed/attribute

   xss/allowed/tag

   xss/denied/pattern

   b. Select a row and click **Manage Elements** to display the corresponding XSS Elements (Tag, Attribute, Pattern) used by the application firewall **Cross-Site Scripting** check.

2. **To customize XSS Elements**: You can edit the User-Defined signature object to customize the allowed Tag, allowed Attributes and denied Patterns. You can add new entries or remove the existing ones.

   a. Navigate to **Application firewall** > **Signatures**, highlight the target User-defined signature, and click **Edit**. Click **Manage SQL/XSS Patterns** to display the **Manage SQL/XSS paths** table.

   b. Select the target XSS row.

      i. Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding XSS element.

      ii. Click **Remove** to remove the selected row.

> **Warning**
>
> You must be very careful before you remove or modify any default XSS element, or delete the XSS path to remove the entire row. The signature rules as well as the Cross-Site Scripting security check rely on these elements for detecting attacks to protect your applications. Customizing the XSS Elements can make your application vulnerable to Cross-Site Scripting attacks if the required pattern is removed during editing.

Using the Learn Feature with the HTML Cross-Site Scripting Check

When the learn action is enabled, the application firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, you can deploy the learned rule as a HTML Cross-Site Scripting relaxation rule.

HTML Cross-Site Scripting Learning enhancement—An application firewall learning enhancement was introduced in release 11.0 of the NetScaler software. To deploy fine grained HTML Cross-Site Scripting relaxation, the application firewall offers fine grained HTML Cross-Site Scripting learning. The learning engine makes recommendations regarding the observed Value Type (Tag, Attribute, Pattern) and the corresponding Value expression observed in the input fields. In addition to checking the blocked requests to determine whether the current rule is too restrictive and needs to be relaxed, you can review the rules generated by the learning engine to determine which value type and value expressions are triggering violations and need to be addressed in relaxation rules.

> **Note**
>
> The application firewall's learning engine can distinguish only the first 128 bytes of the name. If a form has multiple fields with names that match for the first 128 bytes, the learning engine might not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields from HTML Cross-Site Scripting inspection.

To view or use learned data by using the command line interface

At the command prompt, type one of the following commands:

- **show appfw learningdata** <profilename> **crossSiteScripting**
- **rm appfw learningdata** <profilename> **-crossSiteScripting** <string> <formActionURL> [<location>] [<valueType> <valueExpression>]
- **export appfw learningdata** <profilename> **crossSiteScripting**

To view or use learned data by using the configuration utility

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the **HTML Cross-Site Scripting** entry in the Learned Rules table and double-click it to access the learned rules. The table displays the **Field Name**, A**ction URL**, **Value Type**, **Value** and **Hits** columns. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

   You also have the option to show a summarized view of the learned relaxations by selecting the **HTML Cross-Site Scripting** entry in the Learned Rules table and clicking **Visualizer** to get a consolidated view of all the learned violations. The visualizer makes it very easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the Log Feature with the HTML Cross-Site Scripting Check

When the log action is enabled, the HTML Cross-Site Scripting security check violations are logged in the audit log as **APPFW_XSS** violations. The application firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the HTML Cross-Site Scripting violations:

> Shell

> tail -f /var/log/ns.log | grep APPFW_XSS

Example of a Cross-Site Scripting security check violation log message in CEF log format

Jul 11 00:45:51 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|**APPFW_XSS**|6|src=10.217.253.62 geolocation=Unknown spt=4840 method=GET request=http://aaron.stratum8.net/FFC/CreditCardMind.html?abc\=%3Cdef%3E msg=**Cross-site script check failed for field abc\="Bad tag: def"** cn1=133 cn2=294 cs1=pr_ffc cs2=PPE1 cs3=eUljypvLa0BbabwfGVE52Sewg9U0001 cs4=ALERT cs5=2015 act=**not blocked**

Example of a Cross-Site Scripting security check violation log message in Native log format showing transform action

Jul 11 01:00:28 <local0.info> 10.217.31.98 07/11/2015:01:00:28 GMT ns 0-PPE-0 : default APPFW **APPFW_XSS** 132 0 : 10.217.253.62 392-PPE0 eUljypvLa0BbabwfGVE52Sewg9U0001 pr_ffc http://aaron.stratum8.net/FFC/login.php?login_name=%3CBOB%3E&passwd=&drinking_pref=on &text_area=&loginButton=ClickToLogin&as_sfid=AAAAAAVFqmYL68IGvkrcn2pzehjfIkm5E6EZ9FL8YLvIW_41AvAATuKYe9N7uGThSpEAxbb0iBx55jyvqOZNiVK_XwEPstMYvWHxfUWl62WINwRMrKsED FC4llF **Cross-site script special characters seen in fields <transformed>**

To access the log messages by using the configuration utility

The Citrix configuration utility includes a useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **HTML Cross-Site Scripting** row and click **Logs**. When you access the logs directly from the HTML Cross-Site Scripting check of the profile, the configuration utility filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

  The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **HTML Cross-Site Scripting** check, filter by selecting **APPFW** in the dropdown options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_XSS** check box and click the **Apply** button, only log messages pertaining to the HTML Cross-Site Scripting security check violations appear in the Syslog Viewer.

  If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, **Client IP** etc. appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Click to Deploy functionality is available only in the configuration utility. You can use the Syslog Viewer to not only view the logs but also to deploy HTML Cross-Site Scripting relaxation rules based on the log messages for the application firewall security check violations. The log messages must be in CEF log format for this operation. Click to deploy functionality is available only for log messages that are generated by the block (or not block) action. You cannot deploy a relaxation rule for a log message about the transform operation.

To deploy a relaxation rule from the Syslog Viewer, select the log message. A check box appears in the upper right corner of the Syslog Viewer box of the selected row. Select the check box, and then select an option from the **Action** list to deploy the relaxation rule. **Edit & Deploy**, **Deploy**, and **Deploy All** are available as **Action** options.

The HTML Cross-Site Scripting rules that are deployed by using the **Click to Deploy** option do not include the fine grain relaxation recommendations.

To use Click to Deploy functionality in the configuration utility

1. In the Syslog Viewer, select **APPFW** in the **Module** options.
2. Select the **APP_XSS** as the **Event Type** to filter corresponding log messages.
3. Select the check box to identify the rule to deploy.
4. Use the **Action** drop-down list of options to deploy the relaxation rule.
5. Verify that the rule appears in the corresponding relaxation rule section.

Statistics for the HTML Cross-Site Scripting violations

When the stats action is enabled, the counter for the HTML Cross-Site Scripting check is incremented when the application firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, the request for a page that contains 3 HTML Cross-Site Scripting violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display HTML Cross-Site Scripting check statistics by using the command line

At the command prompt, type:

> **sh appfw stats**

To display stats for a specific profile, use the following command:

> **stat appfw profile** <profile name>

To display HTML Cross-Site Scripting statistics by using the configuration utility

1. Navigate to **Security** > **Application Firewall > Profiles > Statistics.**
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about HTML Cross-Site Scripting violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

Note the following points about the HTML Cross-Site Scripting check:

- **Built-in Support for HTML Cross-Site Scripting attack Protection**—The NetScaler application firewall protects against Cross-Site Scripting attacks by monitoring a combination of allowed attributes and tags, as well as denied patterns in the received payload. All the built-in default allowed tags, allowed attributes and denied patterns used by the XSS check are specified in the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

/netscaler/default_custom_settings.xml file.

- **Customization**—You can change the default list of tags, attributes, and patterns to customize the Cross-Site Scripting security check inspection for the specific needs of your application. Make a copy of the default signature object, modify existing entries, or add new ones. Bind this signature object to your profile to make use of the customized configuration.
- **Hybrid Security Model**—Both signatures and deep security protections use the SQL/XSS patterns specified in the signature object that is bound to the profile. If no signature object is bound to the profile, the SQL/XSS patterns present in the default signature object are used.
- **Transform**—Note the following about the transform operation:

  The transform operation works independently of the other Cross-Site Scripting action settings. If transform is enabled and block, log, stats, and learn are all disabled, XSS tags will be transformed.

  If the block action is enabled, it takes precedence over the transform action.

- **Fine Grained Relaxation and Learning**—Fine tune the relaxation rule to relax a subset of XSS elements from security check inspection but detect the rest. The learning engine recommends a specific value type and value expressions based on the observed data.
- **Click to Deploy**—Select one, or multiple XSS violation log messages in the syslog viewer and deploy them as relaxation rules.
- **Charset**—The default charset for the profile should be set based on the need of the application. By default, the profile charset is set to English US (ISO-8859-1). If a request is received without the specified charset, the application firewall processes the request as if it is ISO-8859-1. The open bracket character (<) or the close bracket character (>) will not get interpreted as XSS tags if these characters are encoded in other charsets. For example, if a request contains a UTF-8 character string "**%uff1cscript%uff1e**" but the charset is not specified on the request page, the XSS violation might not get triggered unless the default charset for the profile is specified as Unicode.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTML SQL Injection Check

Dec 03, 2015

Many web applications have web forms that use SQL to communicate with relational database servers. Malicious code or a hacker can use an insecure web form to send SQL commands to the web server. The application firewall HTML SQL Injection check provides special defenses against injection of unauthorized SQL code that might break security. If the application firewall detects unauthorized SQL code in a user request, it either transforms the request, to render the SQL code inactive, or blocks the request. The application firewall examines the request payload for injected SQL code in three locations: 1) POST body, 2) headers, and 3) cookies.

A default set of keywords and special characters provides known keywords and special characters that are commonly used to launch SQL attacks. You can add new patterns, and you can edit the default set to customize the SQL check inspection. The application firewall offers various action options for implementing SQL Injection protection. In addition to the **Block**, **Log**, **Stats** and **Learn** actions, the application firewall profile also offers the option to **transform SQL special characters** to render an attack harmless.

In addition to actions, there are several parameters that can be configured for SQL injection processing. You can check for **SQL wildcard characters**. You can change the SQL Injection type and select one of the 4 options (**SQLKeyword**, **SQLSplChar**, **SQLSplCharANDKeyword**, **SQLSplCharORKeyword**) to indicate how to evaluate the SQL keywords and SQL special characters when processing the payload. The **SQL Comments Handling parameter** gives you an option to specify the type of comments that need to be inspected or exempted during SQL Injection detection.

You can deploy relaxations to avoid false positives. The application firewall learning engine can provide recommendations for configuring relaxation rules.

Following options are available for configuring an optimized SQL Injection protection for your application:

Block—If you enable block, the block action is triggered only if the input matches the SQL injection type specification. For example, if **SQLSplCharANDKeyword** is configured as the SQL injection type, a request is not blocked if it contains no key words, even if SQL special characters are detected in the input. Such a request is blocked if the SQL injection type is set to either **SQLSplChar**, or **SQLSplCharORKeyword**.

Log—If you enable the log feature, the SQL Injection check generates log messages indicating the actions that it takes. If block is disabled, a separate log message is generated for each input field in which the SQL violation was detected. However, only one message is generated when the request is blocked. Similarly, one log message per request is generated for the transform operation, even when SQL special characters are transformed in multiple fields. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.

Stats—If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.

Learn—If you are not sure which SQL relaxation rules might be ideally suited for your application, you can use the learn feature to generate recommendations based on the learned data. The application firewall learning engine monitors the traffic and provides SQL learning recommendations based on the observed values. To get optimal benefit without compromising performance, you might want to enable the learn option for a short time to get a representative sample of the rules, and then deploy the rules and disable learning.

Transform SQL special characters—The application firewall considers three characters, Single straight quote ('), Backslash (\), and Semicolon (;) as special characters for SQL security check processing. The SQL Transformation feature modifies the SQL Injection code in an HTML request to ensure that the request is rendered harmless. The modified HTML request is then sent to the server. All default transformation rules are specified in the /netscaler/default_custom_settings.xml file.

The transform operation renders the SQL code inactive by making the following changes to the request:

- Single straight quote (') to double straight quote (").
- Backslash (\) to double backslash (\\).
- Semicolon (;) is dropped completely.

These three characters (special strings) are necessary to issue commands to an SQL server. Unless an SQL command is prefaced with a special string, most SQL servers ignore that command. Therefore, the changes that the application firewall performs when transformation is enabled prevent an attacker from injecting active SQL. After these changes are made, the request can safely be forwarded to your protected web site. When web forms on your protected web site can legitimately contain SQL special strings, but the web forms do not rely on the special strings to operate correctly, you can disable blocking and enable transformation to prevent blocking of legitimate web form data without reducing the protection that the application firewall provides to your protected web sites.

The transform operation works independently of the SQL Injection Type setting. If transform is enabled and the SQL Injection type is specified as SQL keyword, SQL special characters are transformed even if the request does not contain any keywords.

> Tip

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> You normally enable either transformation or blocking, but not both. If the block action is enabled, it takes precedence over the transform action. If you have blocking enabled, enabling transformation is redundant.

Check for SQL Wildcard Characters—Wild card characters can be used to broaden the selections of a structured query language (SQL-SELECT) statement. These wild card operators can be used in conjunction with **LIKE** and **NOT LIKE** operators to compare a value to similar values. The percent (%), and underscore (_) characters are frequently used as wild cards. The percent sign is analogous to the asterisk (*) wildcard character used with MS-DOS and to match zero, one, or multiple characters in a field. The underscore is similar to the MS-DOS question mark (?) wildcard character. It matches a single number or character in an expression.

For example, you can use the following query to do a string search to find all customers whose names contain the D character.

SELECT * from customer WHERE name like "%D%"

The following example combines the operators to find any salary values that have 0 in the second and third place.

SELECT * from customer WHERE salary like '_00%'

Different DBMS vendors have extended the wildcard characters by adding extra operators. The NetScaler application firewall can protect against attacks that are launched by injecting these wildcard characters. The 5 default Wildcard characters are percent (%), underscore (_), caret (^), opening square bracket ([), and closing square bracket (]). This protection applies to both HTML and XML profiles.

The default wildcard chars are a list of literals specified in the ***Default Signatures**:

- <wildchar type="LITERAL">%</wildchar>
- <wildchar type="LITERAL">_</wildchar>
- <wildchar type="LITERAL">^</wildchar>
- <wildchar type="LITERAL">[</wildchar>
- <wildchar type="LITERAL">]</wildchar>

Wildcard characters in an attack can be PCRE, like [^A-F]. The application firewall also supports PCRE wildcards, but the literal wildcard chars above are sufficient to block most attacks.

> **Note**
>
> The SQL wildcard character check is different from the SQL special character check. This option must be used with caution to avoid false positives.

Check Request Containing SQL Injection Type—The application firewall provides 4 options to implement the desired level of strictness for SQL Injection inspection, based on the individual need of the application. The request is checked against the injection type specification for detecting SQL violations. The 4 SQL injection type options are:

- **SQL Special Character and Keyword**—Both an SQL keyword and an SQL special character must be present in the input to trigger SQL violation. This least restrictive setting is also the default setting.
- **SQL Special Character**—At least one of the special characters must be present in the input to trigger SQL violation.
- **SQL key word**—At least one of the specified SQL keywords must be present in the input to trigger an SQL violation. Do not select this option without due consideration. To avoid false positives, make sure that none of the keywords are expected in the inputs.
- **SQL Special Character or Keyword**—Either the key word or the special character string must be present in the input to trigger the security check violation.

> **Tip**
>
> If you configure the application firewall to check for inputs that contain an SQL special character, the application firewall skips web form fields that do not contain any special characters. Since most SQL servers do not process SQL commands that are not preceded by a special character, enabling this option can significantly reduce the load on the application firewall and speed up processing without placing your protected web sites at risk.

SQL comments handling—By default, the application firewall checks all SQL comments for injected SQL commands. Many SQL servers ignore anything in a comment, however, even if preceded by an SQL special character. For faster processing, if your SQL server ignores comments, you can configure the application firewall to skip comments when examining requests for injected SQL. The SQL comments handling options are:

- **ANSI**—Skip ANSI-format SQL comments, which are normally used by UNIX-based SQL databases.
- **Nested**—Skip nested SQL comments, which are normally used by Microsoft SQL Server.
- **ANSI/Nested**—Skip comments that adhere to both the ANSI and nested SQL comment standards. Comments that match only the ANSI standard, or only the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nested standard, are still checked for injected SQL.

- **Check all Comments**— Check the entire request for injected SQL, without skipping anything. This is the default setting.

> **Tip**
>
> In most cases, you should not choose the Nested or the ANSI/Nested option unless your back-end database runs on Microsoft SQL Server. Most other types of SQL server software do not recognize nested comments. If nested comments appear in a request directed to another type of SQL server, they might indicate an attempt to breach security on that server.

Check Request headers— Enable this option if, in addition to examining the input in the form fields, you want to examine the request headers for HTML SQL Injection attacks. If you use the configuration utility, you can enable this parameter in the **Advanced Settings** -> **Profile Settings** pane of the application firewall profile.

> **Note**
>
> If you enable the Check Request header flag, you might have to configure relaxation rule for the **User-Agent** header. Presence of the SQL keyword **like** and SQL special character semi-colon (**;**) might trigger false positive and block requests that contain this header.

> **Warning**
>
> If you enable both request header checking and transformation, any SQL special characters found in headers are also transformed. The Accept, Accept-Charset, Accept-Encoding, Accept-Language, Expect, and User-Agent headers normally contain semicolons (;). Enabling both Request header checking and transformation simultaneously might cause errors.

SQL Fine grained Relaxations

The application firewall gives you an option to exempt a specific form field, header, or Cookie from SQL Injection inspection check. You can completely bypass the inspection for one or more of these fields by configuring relaxation rules for the SQL Injection check.

The application firewall allows you to implement tighter security by fine tuning the relaxation rules. An application might require the flexibility to allow specific patterns, but configuring a relaxation rule to bypass the security inspection might make the application vulnerable to attacks, because the target field is exempted from inspection for any SQL attack patterns. SQL fine grained relaxation provides the option to allow specific patterns and block the rest. For example, the application firewall currently has a default set of more than 100 SQL keywords. Because hackers can use these keywords in SQL Injection attacks, the application firewall flags them as potential threats. You can relax one or more keyword(s) that are considered safe for the specific location. The rest of the potentially dangerous SQL keywords are still checked for the target location and continue to trigger the security check violations. You now have much tighter control.

The commands used in relaxations have optional parameters for **Value Type** and **Value Expression**. You can specify whether the value expression is a regular expression or a literal string. The value type can be left blank or you have an option to select **Keyword** or **SpecialString** or **WildChar**.

> **Warning**
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the HTML SQL Injection check would otherwise have blocked.

Points to Consider:

- Value expression is an optional argument. A field name might not have any value expression.
- A field name can be bound to multiple value expressions.
- Value expressions should be assigned a value type. The SQL value type can be: 1) Keyword, 2) SpecialString, or 3) WildChar.
- You can have multiple relaxation rules per field name/URL combination.

Using the Command Line to Configure the SQL Injection Check

To configure SQL Injection actions and other parameters by using the command line

In the command line interface, you can use either the **set appfw profile** command or the **add appfw profile** command to configure the SQL Injection

© 1999-2017 Citrix Systems, Inc. All rights reserved.

protections. You can enable the block, learn, log, stats action(s) and specify whether you want to transform the special characters used in SQL Injection attack strings to disable the attack. Select the type of SQL attack pattern (key words, wildcard characters, special strings) you want to detect in the payloads, and indicate whether you want the application firewall to also inspect the request Headers for SQL Injection violations. Use the **unset appfw profile** command to revert the configured settings back to their defaults. Each of the following commands sets only one parameter, but you can include multiple parameters in a single command:

- **set appfw profile** <name> -**SQLInjectionAction** (([**block**] [**learn**] [**log**] [**stats**]) | [**none**])
- **set appfw profile** <name> -**SQLInjectionTransformSpecialChars** (**ON** | **OFF**)
- **set appfw profile** <name> -**SQLInjectionCheckSQLWildChars** (**ON** |**OFF**)
- **set appfw profile** <name> -**SQLInjectionType** ([**SQLKeyword**] | [**SQLSplChar**] | [**SQLSplCharANDKeyword**] | [**SQLSplCharORKeyword**])
- **set appfw profile** <name> -**SQLInjectionParseComments** ([**checkall**] | [**ansi|nested**] | [**ansinested**])
- **set appfw profile** <name> -**CheckRequestHeaders** (**ON** | **OFF**)

To configure a SQL Injection relaxation rule by using the command line

Use the bind or unbind command to add or delete binding, as follows:

- **bind appfw profile** <name> -**SQLInjection** <String> [**isRegex** (**REGEX** | **NOTREGEX**)] <formActionURL> [-**location** <location>] [-**valueType** (**Keyword**|**SpecialString**|**Wildchar**) [<valueExpression>][-**isValueRegex** (**REGEX** | **NOTREGEX**)]]

- **unbind appfw profile** <name> -**SQLInjection** <String> <formActionURL> [-**location** <location>] [-**valueType** (**Keyword**|**SpecialString**|**Wildchar**) [<valueExpression>]]

Using the Configuration Utility to Configure the SQL Injection Security Check

In the configuration utility, you can configure the SQL Injection security check in the pane for the profile associated with your application.

To configure or modify the SQL Injection check by using the configuration utility

1. Navigate to **Application Firewall > Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

   a. If you just want to enable or disable Block, Log, Stats, and Learn actions for HTML SQL Injection, you can select or clear check boxes in the table, click OK, and then click Save and Close to close the Security Check pane.

   b. If you want to configure additional options for this security check, double click HTML SQL Injection, or select the row and click Action Settings, to display the following options:

   Transform SQL Special character—Transform any SQL Special characters in the request.

   Check for SQL Wildcard Characters—Consider SQL Wildcard characters in the payload to be attack patterns.

   Check Request Containing—Type of SQL injection (SQLKeyword, SQLSplChar, SQLSplCharANDKeyword, or SQLSplCharORKeyword) to check.

   SQL Comments Handling—Type of comments (Check All Comments, ANSI, Nested, or ANSI/Nested) to check.

   After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

   3. To enable or disable the **Check request Header** setting, in the Advanced Settings pane, click **Profile Settings**. In **Common Settings**, Select or clear the **Check Request Headers** check box. Click **OK.** You can either use the X icon at the top right hand side of the Profile Settings pane to close this section or, if you have finished configuring this profile, you can click **Done** to return to the **Application Firewall > Profile**.

To configure an SQL Injection relaxation rule by using the configuration utility

- Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
- In the **Advanced Settings** pane, click **Relaxation Rules**.
- In the Relaxation Rules table, double-click the **HTML SQL Injection** entry, or select it and click **Edit**.
- In the **HTML SQL Injection Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

> **Note**
>
> When you add a new rule, the **Value Expression** field is not displayed unless you select **Keyword** or **SpecialString** or **WildChar** option in the **Value Type** Field.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To manage SQL Injection relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **HTML SQL Injection** row and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the SQL Injection patterns by using the configuration utility

You can use the configuration utility to view or customize the SQL patterns.

The default SQL patterns are specified in Application Firewall > Signatures > Default Signatures. If you do not bind any signature object to your profile, the default SQL patterns specified in the Default Signatures object will be used by the profile for the SQL Injection security check processing. The rules and patterns, specified in the default signatures object, are read-only. You cannot edit or modify them. If you want to modify or change these patterns, make a copy of the Default Signatures object to create a User-Defined signature object. Make changes in the SQL patterns in the new User-defined signature object and use this signature object in your profile that is processing the traffic for which you want to use these customized SQL patterns.

For more information, see Signatures.

1. **To view default SQL patterns:**

   a. Navigate to **Application firewall > Signatures**, select ***Default Signatures**, and click **Edit**.

   Then click **Manage SQL/XSS patterns**.

   The **Manage SQL/XSS paths** table shows following four rows pertaining to SQL Injection:

   - Injection (not_alphanum, SQL)/ Keyword

   - Injection (not_alphanum, SQL)/ specialstring

   - Injection (not_alphanum, SQL)/ transformrules/transform

   -Injection (not_alphanum, SQL)/ wildchar

   b. Select a row and click **Manage Elements** to display the corresponding SQL patterns (keywords, special strings, transformation rules or the wildcard characters) used by the application firewall SQL injection check.

2. **To customize SQL patterns:** You can edit the User-Defined signature object to customize the SQL key words, special strings, and wildcard characters. You can add new entries or remove the existing ones. You can modify the transformation rules for the SQL special strings.

   a. **Navigate to Application firewall > Signatures**, highlight the target User-defined signature, and click **Edit**. Click **Manage SQL/XSS patterns** to display the **Manage SQL/XSS paths** table.

   b. Select the target SQL Injection row.

      i. Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding SQL element.

      ii. Click **Remove** to remove the selected row.

> **Warning**
>
> You must be very careful before you remove or modify any default SQL element, or delete the SQL path to remove the entire row. The signature rules as well as the SQL Inject security check rely on these elements for detecting SQL Injection attacks to protect your applications. Customizing the SQL patterns can make your application vulnerable to SQL attacks if the required pattern is removed during editing.

Using the Learn Feature with the SQL Injection Check

When the learn action is enabled, the application firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, you can deploy the learned rule as an SQL Injection relaxation rule.

SQL Injection Learning enhancement—An application firewall learning enhancement was introduced in release 11.0 of the NetScaler software. To deploy fine grained SQL Injection relaxation, the application firewall offers fine grained SQL Injection learning. The learning engine makes recommendations regarding the observed Value Type (keyword, SpecialString, Wildchar) and the corresponding Value expression observed in the input fields. In addition to checking the blocked requests to determine whether the current rule is too restrictive and needs to be relaxed, you can review the rules generated by the learning engine to determine which value type and value expressions are triggering violations and need to be addressed in relaxation rules.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Important**
>
> The application firewall's learning engine can distinguish only the first 128 bytes of the name. If a form has multiple fields with names that match for the first 128 bytes, the learning engine might not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields from SQL Injection inspection.

To view or use learned data by using the command line interface

At the command prompt, type one of the following commands:

- **show appfw learningdata** <profilename> **SQLInjection**
- **rm appfw learningdata** <profilename> **-SQLInjection** <string> <formActionURL> [<location>] [<valueType> <valueExpression>]
- **export appfw learningdata** <profilename> **SQLInjection**

To view or use learned data by using the configuration utility

1. Navigate to **Application Firewall > Profiles**, highlight the target profile, and click **Edit**.

2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the **HTML SQL Injection** entry in the Learned Rules table and double-click it to access the learned rules. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

You also have the option to show a summarized view of the learned relaxations by selecting the **HTML SQL Injection** entry in the Learned Rules table and clicking **Visualizer** to get a consolidated view of all the learned violations. The visualizer makes it very easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the Log Feature with the SQL Injection Check

When the log action is enabled, the HTML SQL Injection security check violations are logged in the audit log as **APPFW_SQL** violations. The application firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the SQL Injection violations:

> Shell

 # tail -f /var/log/ns.log | grep APPFW_SQL

Example of a HTML SQL Injection log message when the request is transformed

Jun 26 21:08:41 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|**APPFW_SQL**|6|src=10.217.253.62 geolocation=Unknown spt=54001 method=GET request=http://aaron.stratum8.net/FFC/login.php?login_name\=%27+or&passwd\=and+%3B&drinking_pref\=on&text_area\=select+*+from+%5C+%3B&loginButton\=ClickToLogin&as_sfid\=AAAAAAXjnGN5gLH-hvhTOpIySEIqES7BjFRs5Mq0fwPp-3ZHDi5yWlRWByj0cVbMyy-Ens2vaaiULKOcUri4OD4kbXWwSY5s7I3QkDsrvIgCYMC9BMvBwY2wbNcSqCwk52lfE0k%3D&as_fid\=feeec8758b41740eedeeb6b35b85dfd3d5def30c msg=Special characters seen in fields cn1=74 cn2=762 cs1=pr_ffc cs2=PPE1 cs3=9ztIlf9p1H7p6Xtzn6NMygTv/QM0002 cs4=ALERT cs5=2015 **act=transformed**

Example of a HTML SQL Injection log message when the post request is blocked

Jun 26 21:30:34 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|**APPFW_SQL**|6|src=10.217.253.62 geolocation=Unknown spt=9459 method=POST request=http://aaron.stratum8.net/FFC/login_post.php msg=SQL Keyword check failed for field text_area\="(')" cn1=78 cn2=834 cs1=pr_ffc cs2=PPE1 cs3=eVJMMPtZ2XgylGrHjkx3rZLfBCI0002 cs4=ALERT cs5=2015 **act=blocked**

> **Note**
>
> As part of the streaming changes in 10.5.e build (enhancement builds) as well as 11.0 build onwards, we now process the input data in blocks. RegEx pattern matching is now restricted to 4K for contiguous character string matching. With this change, the SQL violation log messages might include different information compared to the earlier builds. The keyword and special character in the input could be separated by a large number of bytes. We now keep track of the SQL keywords and special strings when processing the data, instead of buffering the entire input value. In addition to the field name, the log message now includes the SQL keyword, or the SQL special character, or both the SQL keyword and the SQL special character, as determined by the configured setting. The rest of the input is no longer included in the log message, as shown in the following example:
>
> **Example:**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> In 10.5, when the application firewall detects the SQL violation, the entire input string might be included in the log message, as shown below:
>
> SQL Keyword check failed for field **text=\"select a name from testbed1\;\\(\;\\)\".*\<blocked\>**
>
> In enhancement builds of 10.5.e that support request side streaming as well as 11.0 build onwards, we log only the field name, keyword and special character (if applicable) in the log message, as shown below:
>
> SQL Keyword check failed for field **text="select(;)" <blocked>**
>
> This change is applicable to requests that contain **application/x-www-form-urlencoded**, or **multipart/form-data**, or **text/x-gwt-rpc** content-types. Log messages generated during processing of **JSON** or **XML** payloads are not affected by this change.

To access the log messages by using the configuration utility

The Citrix configuration utility includes a useful tool (**Syslog Viewer**) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **HTML SQL Injection** row and click **Logs**. When you access the logs directly from the HTML SQL Injection check of the profile, the configuration utility filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **HTML SQL Injection** check, filter by selecting **APPFW** in the dropdown options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_SQL** check box and click the **Apply** button, only log messages pertaining to the **SQL Injection** security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, **Client IP** etc. appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Click to Deploy functionality is available only in the configuration utility. You can use the Syslog Viewer to not only view the logs but also to deploy HTML SQL Injection relaxation rules based on the log messages for the application firewall security check violations. The log messages must be in CEF log format for this operation. Click to deploy functionality is available only for log messages that are generated by the block (or not block) action. You cannot deploy a relaxation rule for a log message about the transform operation.

To deploy a relaxation rule from the Syslog Viewer, select the log message. A check box appears in the upper right corner of the Syslog Viewer box of the selected row. Select the check box, and then select an option from the Action list to deploy the relaxation rule. **Edit & Deploy**, **Deploy**, and **Deploy All** are available as **Action** options.

The SQL Injection rules that are deployed by using the Click to Deploy option do not include the fine grain relaxation recommendations.

To use Click to Deploy functionality in the configuration utility

1. In the Syslog Viewer, select **APPFW** in the **Module** options.
2. Select the **APP_SQL** as the **Event Type** to filter corresponding log messages.
3. Select the check box to identify the rule to deploy.
4. Use the **Action** drop-down list of options to deploy the relaxation rule.
5. Verify that the rule appears in the corresponding relaxation rule section.

Statistics for the SQL Injection violations

When the stats action is enabled, the counter for the SQL Injection check is incremented when the application firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, the request for a page that contains 3 SQL Injection violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display SQL Injection check statistics by using the command line

At the command prompt, type:

> **sh appfw stats**

To display stats for a specific profile, use the following command:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

> **stat appfw profile** <profile name>

To display HTML SQL Injection statistics by using the configuration utility

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about HTML SQL Injection violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

Note the following points about the SQL Injection check:

- **Built-in Support for SQL Injection Protection**—The NetScaler application firewall protects against SQL Injection by monitoring a combination of SQL keywords and special characters in the form parameters. All SQL keywords, special characters, wildcard characters, and default transformation rules are specified in the /netscaler/default_custom_settings.xml file.
- **Customization**—You can change the default key words, special characters, wildcard characters and transformation rules to customize the SQL security check inspection for the specific needs of your application. Make a copy of the default signature object, modify existing entries, or add new ones. Bind this signature object to your profile to make use of the customized configuration.
- **Hybrid Security Model**—Both signatures and deep security protections use the SQL/XSS patterns specified in the signature object that is bound to the profile. If no signature object is bound to the profile, the SQL/XSS patterns present in the default signature object are used.
- **Transform**—Note the following about the transform operation:
  - The transform operation works independently of the other SQL Injection action settings. If transform is enabled and block, log, stats, and learn are all disabled, SQL special characters will be transformed.
  - When SQL Transformation is enabled, user requests are sent to the backend servers after the SQL special characters are transformed in non-block mode. If the block action is enabled, it takes precedence over the transform action. If the injection type is specified as SQL special character and block is enabled, the request is blocked despite the transform action.
- **Fine Grained Relaxation and Learning**—Fine tune the relaxation rule to relax a subset of SQL elements from security check inspection but detect the rest. The learning engine recommends a specific value type and value expressions based on the observed data.
- **Click to Deploy**—Select one, or multiple SQL violation log messages in the syslog viewer and deploy them as relaxation rules.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Buffer Overflow Check

Jan 18, 2016

The Buffer Overflow check detects attempts to cause a buffer overflow on the web server. If the application firewall detects that the URL, cookies, or header are longer than the specified maximum length in a request, it blocks that request because it might be an attempt to cause a buffer overflow.

The Buffer Overflow check prevents attacks against insecure operating-system or web-server software that can crash or behave unpredictably when it receives a data string that is larger than it can handle. Proper programming techniques prevent buffer overflows by checking incoming data and either rejecting or truncating overlong strings. Many programs, however, do not check all incoming data and are therefore vulnerable to buffer overflows. This issue especially affects older versions of web-server software and operating systems, many of which are still in use.

The Buffer Overflow security check allows you to configure the **Block**, **Log**, and **Stats** actions. In addition, you can also configure the following parameters:

- **Maximum URL Length.** The maximum length the application firewall allows in a requested URL. Requests with longer URLs are blocked. **Possible Values**: 0-65535. **Default**: 1024
- **Maximum Cookie Length.** The maximum length the application firewall allows for all cookies in a request. Requests with longer cookies trigger the violations. **Possible Values**: 0-65535. **Default**: 4096
- **Maximum Header Length.** The maximum length the application firewall allows for HTTP headers. Requests with longer headers are blocked. **Possible Values**: 0-65535. **Default**: 4096

Using the Command Line to Configure the Buffer Overflow Security Check

To configure Buffer Overflow security check actions and other parameters by using the command line

If you use the command-line interface, you can add the following Buffer Overflow Check arguments to the set appfw profile <profileName> command:

- -bufferOverflowAction [[**block**] [**log**] [**stats**]] | [**none**]
- -bufferOverflowMaxURLLength <positiveInteger>
- -bufferOverflowMaxCookieLength <positiveInteger>
- -bufferOverflowMaxHeaderLength <positiveInteger>

Using the Configuration Utility to Configure the Buffer Overflow Security Check

In the configuration utility, you can configure the Buffer Overflow security check in the pane for the profile associated with your application.

To configure or modify the Buffer Overflow security check by using the configuration utility

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

   a. If you just want to enable or disable **Block**, **Log**, and **Stats** actions for **Buffer Overflow**, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the Security Check pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

b. If you want to configure additional options for this security check, double click **Buffer Overflow**, or select the row and click **Action Settings**, to display the following options:

Maximum URL Length.

Maximum Cookie Length.

Maximum Header Length.

After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

Using the Log Feature with the Buffer Overflow Security Check

When the log action is enabled, the Buffer Overflow security check violations are logged in the audit log as **APPFW_BUFFEROVERFLOW_URL**, **APPFW_BUFFEROVERFLOW_COOKIE**, and **APPFW_BUFFEROVERFLOW_HDR** violations. The application firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

If you use the configuration utility to review the logs, you can use the click-to-deploy feature to apply relaxations indicated by the logs.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the Buffer overflow violations:

> **Shell**

> **tail -f /var/log/ns.log | grep APPFW_BUFFEROVERFLOW**

Example of a CEF log message showing bufferOverflowMaxCookieLength violation in non-block mode

Oct 22 17:35:20 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|**APPFW_BUFFEROVERFLOW_COOKIE**|6|src=10.217.253.62 geolocation=Unknown spt=41198 method=GET request=http://aaron.stratum8.net/FFC/sc11.html **msg=Cookie header length(43) is greater than maximum allowed(16).** cn1=119 cn2=465 cs1=owa_profile cs2=PPE1 cs3=wvOOOb+cJ2ZRbstZpyeNXIqLj7Y0001 cs4=ALERT cs5=2015 **act=not blocked**

Example of a CEF log message showing bufferOverflowMaxURLLength violation in non-block mode

Oct 22 18:39:56 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|**APPFW_BUFFEROVERFLOW_URL**|6|src=10.217.253.62 geolocation=Unknown spt=19171 method=GET request=http://aaron.stratum8.net/FFC/sc11.html **msg=URL length(39) is greater than maximum allowed(20).** cn1=707 cn2=402 cs1=owa_profile cs2=PPE0 cs3=kW49GcKbnwKByByi3+jeNzfgWa80000 cs4=ALERT cs5=2015 **act=not blocked**

Example of a Native Format Log message showing bufferOverflowMaxHeaderLength violation in block mode

Oct 22 18:44:00 <local0.info> 10.217.31.98 10/22/2015:18:44:00 GMT ns 0-PPE-2 : default APPFW **APPFW_BUFFEROVERFLOW_HDR** 155 0 : 10.217.253.62 374-PPE2 khhBEeY4DB8V2D3H2sMLkXmfWnA0002 owa_profile **Header(User-Agent) length(82) is greater than maximum allowed(10)** : http://aaron.stratum8.net/ **<blocked>**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To access the log messages by using the configuration utility

The Citrix configuration utility includes a useful tool (**Syslog Viewer**) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **Buffer Overflow** row and click **Logs**. When you access the logs directly from the Buffer Overflow Security Check of the profile, the configuration utility filters out the log messages and displays only the logs pertaining to these security check violations.

- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.

- Navigate to **Application Firewall** > **policies** > **Auditing**. In the **Audit Messages** section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The XML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **Buffer Overflow** check, filter by selecting **APPFW** in the dropdown options for **Module**. The **Event Type** list offers three options, **APPFW_BUFFEROVERFLOW_URL**, **APPFW_BUFFEROVERFLOW_COOKIE**, and A**PPFW_BUFFEROVERFLOW_HDR**, to view all the log messages pertaining to buffer overflow security check. You can select one or more options to further refine your selection. For example, if you select the **APPFW_BUFFEROVERFLOW_COOKIE** check box and click the **Apply** button, only log messages pertaining to the Buffer Overflow security check violations for the Cookie header appear in the Syslog Viewer. If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, and **Client IP**, appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Click-to-Deploy: The configuration utility provides click-to-deploy functionality, which is currently supported only for the buffer overflow log messages pertaining to the **URL Length** violations. You can use the Syslog Viewer to not only view the triggered violations, but also execute informed decisions based on the observed lengths of the blocked messages. If the current value is too restrictive and is triggering false positives, you can select a message and deploy it to replace the current value with the URL length value seen in the message. The log messages must be in CEF log format for this operation. If the relaxation can be deployed for a log message, a check box appears at the right edge of the Syslog Viewer box in the row. Select the check box, and then select an option from the **Action** list to deploy the relaxation. **Edit & Deploy**, **Deploy**, and **Deploy All** are available as **Action** options. You can use the **APPFW_BUFFEROVERFLOW_URL** filter to isolate all the log messages pertaining to the configured URL length violations.

If you select an individual log message, all three action options **Edit & Deploy**, **Deploy**, and **Deploy All** are available. If you select **Edit & Deploy**, the **Buffer Overflow settings** dialogue is displayed. The new URL length that was observed in the request is inserted into the Maximum URL length input field. If you click **Close** without any edits, the current configured values remain unchanged. If you click the **OK** button, the new value of the Maximum URL length replaces the previous value.

> ## Note
>
> The **block**, **log** and **stats** action check boxes are unchecked in the displayed **Buffer Overflow settings** dialogue, and need to be reconfigured if you select the **Edit & Deploy** option. Make sure to enable these check boxes before clicking **OK**, otherwise the new URL length will get configured but the actions will be set to **none**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If you select the check boxes for multiple log messages, you can use the **Deploy** or **Deploy All** option. If the deployed log messages have different URL lengths, the configured value gets replaced by the highest URL Length value observed in the selected messages. Deploying the rule results only in changing the **bufferOverflowMaxURLLength** value. Configured actions are retained and remain unchanged.

To use Click-to-Deploy functionality in the configuration utility

1. In the Syslog Viewer, select **APPFW** in the **Module** options.
2. Enable the **APPFW_BUFFEROVERFLOW_URL** check box as the **Event Type** to filter corresponding log messages.
3. Enable the check box to select the rule.
4. Use the **Action** drop-down list of options to deploy the relaxation.
5. Navigate to **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks** to access the **Buffer Overflow** settings pane to verify that the **Maximum URL Length** value is updated.

Statistics for the Buffer Overflow violations

When the stats action is enabled, the counter for the Buffer Overflow Security Check is incremented when the application firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, a request for a page that contains three Buffer Overflow violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display Buffer Overflow Security Check statistics by using the command line

At the command prompt, type:

> **sh appfw stats**

To display stats for a specific profile, use the following command:

> **stat appfw profile** <profile name>

To display Buffer Overflow statistics by using the configuration utility

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about Buffer Overflow violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

- The buffer overflow security check allows you to configure limits to enforce the maximum length of allowed URLs, Cookies and Headers.
- **Block**, **Log** and **Stats** actions enable you to monitor the traffic and configure optimal protection for your application.
- Syslog viewer enables you to filter and view all the log messages pertaining to buffer overflow violations.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Click-to-Deploy** functionality is supported for the **bufferOverflowMaxURLLength** violations. You can select and deploy an individual rule, or you can select multiple log messages to tweak and relax the current configured value of the maximum allowed length of the URL. The highest value of the URL from the selected group is set as the new value, to allow all these requests that are currently flagged as violations.
- The application firewall now evaluates individual cookies when inspecting the incoming request. If length of any one cookie received in the Cookie header exceeds the configured **BufferOverflowMaxCookieLength**, the Buffer Overflow violation is triggered.

Important

In release 10.5.e (in a few interim enhancement builds prior to 59.13xx.e build) as well as in the 11.0 release (in builds prior to 65.x), application firewall processing of the Cookie header was changed. In those releases, every cookie is evaluated individually, and if the length of any one cookie received in the Cookie header exceeds the configured BufferOverflowMaxCookieLength, the Buffer Overflow violation is triggered. As a result of this change, requests that were blocked in 10.5 and earlier release builds might be allowed, because the length of the entire cookie header is not calculated for determining the cookie length. In some situations, the total cookie size forwarded to the server might be larger than the accepted value, and the server might respond with "400 Bad Request".

Note that this change has been reverted. The behavior in the 10.5.e ->59.13xx.e and subsequent 10.5.e enhancement builds as well as in the 11.0 release 65.x and subsequent builds is now similar to that of the non-enhancement builds of release 10.5. The entire raw Cookie header is now considered when calculating the length of the cookie. Surrounding spaces and the semicolon (;) characters separating the name-value pairs are also included in determining the cookie length.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie Consistency Check

May 23, 2017

The Cookie Consistency check examines cookies returned by users, to verify that they match the cookies that your web site set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the web server. You can also configure the Cookie Consistency check to transform all of the server cookies that it processes, by encrypting the cookies, proxying the cookies, or adding flags to the cookies. This check applies to requests and responses.

An attacker would normally modify a cookie to gain access to sensitive private information by posing as a previously authenticated user, or to cause a buffer overflow. The Buffer Overflow check protects against attempts to cause a buffer overflow by using a very long cookie. The Cookie Consistency check focuses on the first scenario.

If you use the wizard or the configuration utility, in the Modify Cookie Consistency Check dialog box, on the General tab you can enable or disable the following actions:

- Block
- Log
- Learn
- Statistics
- Transform. If enabled, the Transform action modifies all cookies as specified in the following settings:
  - **Encrypt Server Cookies.** Encrypt cookies set by your web server, except for any listed in the Cookie Consistency check relaxation list, before forwarding the response to the client. Encrypted cookies are decrypted when the client sends a subsequent request, and the decrypted cookies are reinserted into the request before it is forwarded to the protected web server. Specify one of the following types of encryption:
    - **None.** Do not encrypt or decrypt cookies. The default.
    - **Decrypt only.** Decrypt encrypted cookies only. Do not encrypt cookies.
    - **Encrypt session only.** Encrypt session cookies only. Do not encrypt persistent cookies. Decrypt any encrypted cookies.
    - **Encrypt all.** Encrypt both session and persistent cookies. Decrypt any encrypted cookies.
      Note: When encrypting cookies, the application firewall adds the **HttpOnly** flag to the cookie. This flag prevents scripts from accessing and parsing the cookie. The flag therefore prevents a script-based virus or trojan from accessing a decrypted cookie and using that information to breach security. This is done regardless of the Flags to Add in Cookies parameter settings, which are handled independently of the Encrypt Server Cookies parameter settings.
- **Proxy Server Cookies.** Proxy all non-persistent (session) cookies set by your web server, except for any listed in the Cookie Consistency check relaxation list. Cookies are proxied by using the existing application firewall session cookie. The application firewall strips session cookies set by the protected web server and saves them locally before forwarding the response to the client. When the client sends a subsequent request, the application firewall reinserts the session cookies into the request before forwarding it to the protected web server. Specify one of the following settings:
  - **None.** Do not proxy cookies. The default.
  - **Session only.** Proxy session cookies only. Do not proxy persistent cookies.
    Note: If you disable cookie proxying after having enabled it (set this value to None after it was set to Session only), cookie proxying is maintained for sessions that were established before you disabled it. You can therefore safely disable this feature while the application firewall is processing user sessions.
- **Flags to Add in Cookies**. Add flags to cookies during transformation. Specify one of the following settings:
  - **None.** Do not add flags to cookies. The default.
  - **HTTP only.** Add the HttpOnly flag to all cookies. Browsers that support the HttpOnly flag do not allow scripts to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

access cookies that have this flag set.

- **Secure**. Add the Secure flag to cookies that are to be sent only over an SSL connection. Browsers that support the Secure flag do not send the flagged cookies over an insecure connection.
- **All.** Add the HttpOnly flag to all cookies, and the Secure flag to cookies that are to be sent only over an SSL connection.

If you use the command-line interface, you can enter the following commands to configure the Cookie Consistency Check:

- set appfw profile <name> -cookieConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]
- set appfw profile <name> -cookieTransforms ([**ON**] | [**OFF**])
- set appfw profile <name> -cookieEncryption ([**none**] | [**decryptOnly**] | [**encryptSession**] | [**encryptAll**])
- set appfw profile <name> -cookieProxying ([**none**] | [**sessionOnly**])
- set appfw profile <name> -addCookieFlags ([**none**] | [**httpOnly**] | [**secure**] | [**all**])

To specify relaxations for the Cookie Consistency check, you must use the configuration utility. On the Checks tab of the Modify Cookie Consistency Check dialog box, click Add to open the Add Cookie Consistency Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Cookie Consistency Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

Following are examples of Cookie Consistency check relaxations:

- **Logon Fields.** The following expression exempts all cookie names beginning with the string logon_ followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

  ^logon_[0-9A-Za-z]{2,15}$
- **Logon Fields (special characters).** The following expression exempts all cookie names beginning with the string türkçe-logon_ followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

  ^t\xC3\xBCrk\xC3\xA7e-logon_[0-9A-Za-z]{2,15}$
- **Arbitrary strings.** Allow cookies that contain the string sc-item_, followed by the ID of an item that the user has added to his shopping cart ([0-9A-Za-z]+), a second underscore (_), and finally the number of these items he wants ([1-9][0-9]?), to be user-modifiable:

  ^sc-item_[0-9A-Za-z]+_[1-9][0-9]?$

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the Cookie Consistency check would otherwise have blocked.

Important

In release 10.5.e (in a few interim enhancement builds prior to 59.13xx.e build) as well as in the 11.0 release (in builds prior to 65.x), application firewall processing of the Cookie header was changed. In those releases, every cookie is evaluated individually, and if the length of any one cookie received in the Cookie header exceeds the configured BufferOverflowMaxCookieLength, the Buffer Overflow violation is triggered. As a result of this change, requests that were blocked in 10.5 and earlier release builds might be allowed, because the length of the entire cookie header is not calculated for determining the cookie length. In some situations, the total cookie size forwarded to the server might be larger than the accepted value, and the server might respond with "400 Bad Request".

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Note that this change has been reverted. The behavior in the 10.5.e ->59.13xx.e and subsequent 10.5.e enhancement builds as well as in the 11.0 release 65.x and subsequent builds is now similar to that of the non-enhancement builds of release 10.5. The entire raw Cookie header is now considered when calculating the length of the cookie. Surrounding spaces and the semicolon (;) characters separating the name-value pairs are also included in determining the cookie length.

Note

Sessionless Cookie Consistency: The cookie consistency behavior has changed in release 11.0. In earlier releases, the cookie consistency check invokes sessionization. The cookies are stored in the session and signed. A "wlt_" suffix is appended to transient cookies and a "wlf_" suffix is appended to the persistent cookies before they are forwarded to the client. Even if the client does not return these signed wlf/wlt cookies, the application firewall uses the cookies stored in the session to perform the cookie consistency check.

In release 11.0, the cookie consistency check is sessionless. The application firewall now adds a cookie that is a hash of all the cookies tracked by the application firewall. If this hash cookie or any other tracked cookie is missing or tampered with, the application firewall strips the cookies before forwarding the request to the back end server and triggers a cookie-consistency violation. The server treats the request as a new request and sends new Set-Cookie header(s).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Firewall Support for Google Web Toolkit

Oct 09, 2016

Note: This feature is available in NetScaler release 10.5.e.

Web servers following Google Web Toolkit (GWT) Remote Procedure Call (RPC) mechanisms can be secured by the NetScaler application firewall without a need for any specific configuration to enable the GWT support.

What is GWT

The GWT is used for building and optimizing complex high-performance web applications by people who do not have expertise in XMLHttpRequest, and JavaScript. This open source, free development toolkit is used extensively for developing small and large scale applications and is quite frequently used for displaying browser based data such as search results for flights, hotels, and so on. The GWT provides a core set of Java APIs and widgets for writing optimized JavaScript scripts that can run on most browsers and mobile devices. The GWT RPC framework makes it easy for the client and server components of the web application to exchange Java objects over HTTP. GWT RPC services are not the same as web services based on SOAP or REST. They are simply a lightweight method for transferring data between the server and the GWT application on the client. GWT handles serialization of the Java objects exchanging the arguments in the method calls and the return value.

For popular websites that use GWT, see

https://www.quora.com/What-web-applications-use-Google-Web-Toolkit-%28GWT%29

How a GWT Request Works

The GWT RPC request is pipe delimited and has variable number of arguments. It is carried as a payload of HTTP POST and has the following values:

1. Content-type = text/x-gwt-rpc. Charset can be any value.
2. Method = POST.

The following example shows a valid payload for a GWT request:

5|0|8|http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com.test.client.TestService|testMethod|java.lang.String|java.lang.Integer| myInput1|java.lang.Integer/3438268394|1|2|3|4|2|5|6|7|8|1|

The request can be divided into three parts:

a) Header: 5|0|8|

The first 3 digits "5|0|8|" in the above request, represent “version, subversion, and size of table”, respectively. These must be positive integers.

b) String Table:

http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com.test.client.TestService|testMethod|java.lang.String|java.lang.Integer|myInput1| java.lang.Integer/3438268394|

The members of the above pipe delimited string table contain the user-provided inputs. These inputs are parsed for the application firewall checks and are identified as follows:

- 1st : http://localhost:8080/test/
  This is the Request URL. It should not contain the query part, because the GET method is not allowed.
- 2nd : 16878339F02B83818D264AE430C20468
  Unique HEX identifier. A request is considered malformed if this string has non-hex characters.
- 3rd : com.test.client.TestService
  Service Class name
- 4th : testMethod
  Service method name
- 5th onwards: java.lang.String|java.lang.Integer|myInput1|java.lang.Integer/3438268394
  Data-types and data. Non-primitive data-types are specified as

  <container>.<sub-cntnr>.name/<integer_identifier>

c) Payload: 1|2|3|4|2|5|6|7|8|1|

The payload consists of references to the elements in the string table. These integer values cannot be larger than the number of elements in the string table.

Application firewall protection for GWT applications

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The application firewall understands and interprets GWT RPC requests, inspects the payload for security check violations, and takes specified actions.

The application firewall headers and cookies checks for GWT requests are similar to those for other request formats. After appropriate URL decoding and charset conversion, all the parameters in the string table are inspected. The GWT request body does not contain field names, just the field values. The input values can be validated against the specified format by using the application firewall Field Format check, which can also be used to control the length of the input. The **Cross-site Scripting** and **SQL Injection** attacks in the inputs can be easily detected and thwarted by the application firewall.

Learning and relaxation rules: Learning and deployment of relaxation rules are supported for GWT requests. Application firewall rules are in the form of <actionURL> <fieldName> mapping. The GWT request format does not have the field names and thus requires special handling. The application firewall inserts dummy field names in the learned rules that can be deployed as relaxation rules. The -isRegex flag works as it does for non-GWT rules.

Action URL:

Multiple services responding to an RPC can be configured on the same web server. The HTTP request has the URL of the web server, not of the actual service handling the RPC. Therefore, relaxation is not applied on the basis of the HTTP request URL, because that would relax all the services on that URL for the target field. For GWT requests, the application firewall uses the URL of the actual service found in the GWT payload, in the fourth field in the string table.

Field name:

Since the GWT request body contains only field values, the application firewall inserts dummy field names such as 1, 2, and so on when recommending learned rules.

Example of a GWT learned rule

```
POST /abcd/def/gh HTTP/1.1
Content-type: text/x-gwt-rpc
Host: 10.217.222.75
Content-length: 157

5|0|8|http://localhost:8080/acdtest/|16878339F02Baf83818D264AE430C20468|
com.test.client.TestService|testMethod|java.lang.String%3b|java.lang.Integer|onblur|

  The learn data will be as follows:
> sh learningdata pr1 crossSiteScripting
Profile: pr1 SecurityCheck: crossSiteScripting
1) Url:    http://localhost:8080/acdtest/  >> From GWT Payload.
 Field:  10
 Hits:   1
 Done
```

Example of a GWT relaxation rule

```
bind appfw profile pr1 -crossSiteScripting 1 abcd -isregex NOTREGEX
```

Log Messages: The application firewall generates log messages for the security check violations that are detected in the GWT requests. A log message generated by a malformed GWT request contains the string “GWT” for easy identification.

Example of a Log message for malformed GWT request

```
Dec 5 21:48:02 <local0.notice> 10.217.31.247 12/05/2014:21:48:02 GMT ns 0-PPE-0 : APPFW Message 696 0 : "GWT RPC request with malformed payload. <blocked>”
```

Difference in processing of GWT vs non-GWT requests

The same payload can trigger different application firewall security check violations for different Content-types. Consider the following example:

```
5|0|8|http://localhost:8080/acdtest/|16878339F02Baf83818D264AE430C20468|com.test.client.TestService|testMethod|java.lang.String%3b|java.lang.Integer|select|
```

Content-type: application/x-www-form-urlencoded

A request sent with this content type results in a SQL violation if the SQL Injection Type is configured to use any of the four available options: SQLSplCharANDKeyword, SQLSplCharORKeyword, SQLKeyword, or SQLSplChar. The application firewall considers '&' to be the field separator and '=' to be the name-value separator when processing the above payload. Since neither of these characters appears anywhere in the post body, the entire content is treated as a single field name. The field name in this request contains both an SQL special character (;) and an SQL Keyword (select). Therefore violations are caught for all four SQL Injection type options.

Content-type: text/x-gwt-rpc

A request sent with this content type triggers an SQL violation only if the SQL injection type is set to one of the following three options:SQLSplCharORKeyword,SQLKeyword, or SQLSplChar. No violation is triggered if the SQL injection type is set to SQLSplCharANDKeyword, which is the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

default option. The application firewall considers the vertical bar ' | ' to be the field separator for the above payload in the GWT request. Therefore, the post body is divided into various form-field values, and form-field names are added (in accordance with the convention described earlier). Because of this splitting, the SQL special character and SQL keyword become parts of separate form fields.

Form field 8: java.lang.String%3b -> %3b is the (;) char

Form Field 10: select

As a result, when SQL Injection Type is set to **SQLSplChar**, field 8 indicates the SQL violation. For **SQLKeyword**, field 10 indicates the violation. Either of these two fields can indicate a violation if the SQL Inject type is configured with the **SQLSplCharORKeyword** option, which looks for the presence of either a keyword or a special character. No violation is caught for the default **SQLSplCharANDKeyword** option, because there is no single field that has a value that contains both **SQLSplChar** and **SQLKeyword** together.

Tips:

- No special application firewall configuration is needed to enable GWT support.
- The Content-type must be text/x-gwt-rpc.
- Learning and deploying of the relaxation rules for all the pertinent application firewall security checks applied to GWT payload works the same as it does for the other supported content-types.
- Only POST requests are considered valid for GWT. All other request methods are blocked if the content-type is text/x-gwt-rpc.
- GWT requests are subject to the configured POST body limit of the profile.
- The sessionless setting for the security checks is not applicable and will be ignored.
- CEF log format is supported for the GWT log messages.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Data Leak Prevention Checks

Mar 28, 2012

The data-leak-prevention checks filter responses to prevent leaks of sensitive information, such as credit card numbers and social security numbers, to unauthorized recipients.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Credit Card Check

Feb 03, 2014

If you have an application that accepts credit cards, or your websites have access to database servers that store credit card numbers, you must use Data Leak Prevention (DLP) measures and configure protection for each type of credit card that you accept.

The NetScaler application firewall Credit Card check prevents attackers from exploiting Data Leak Prevention flaws to obtain credit card numbers of your customers. By following simple configuration steps, you can enforce protection of one or more of the following credit cards: 1) Visa, 2) Master Card, 3) Discover, 4) American Express (Amex), 5) JCB, and 6) Diners Club.

The Credit Card security check examines server responses to identify instances of the target credit card numbers, and applies a specified action when such a number is found. The action can be to transform the response by X'ing out all but the last group of digits in the credit card number, or to block the response if it contains more than a specified number of credit card numbers. If you specify both, the block action takes precedence. The Maximum credit cards allowed per page setting determines when the block action is invoked. The default setting, 0 (no credit card numbers allowed on the page), is the safest, but you can allow up to 255. Depending on where the violation is detected in the response and the block action gets triggered, you might get fewer than the maximum allowed number of credit cards in the response.

To avoid false positives, you can apply relaxations to exempt specific numbers from the Credit Card check. For example, a social security number, purchase order number, or Google account number might be similar to a credit card number. You can specify individual numbers or use a regular expression to indicate the string of digits to be bypassed when processing the response URL for credit card inspection.

If you're not sure which credit card numbers to exempt, you can use the learn feature to generate recommendations based on the learned data. To get optimal benefit without compromising performance, you might want to enable this option for a short time to get a representative sample of the rules, and then deploy the relaxations and disable learning.

If you enable the log feature, the Credit Card check generates log messages indicating the actions that it takes. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate thwarted attempts to gain access. By default, the doSecureCreditCardLogging parameter is ON, so the credit card number is not included in the log message generated by the safe commerce (Credit Card) violation.

The stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack.

To configure the Credit Card security check for protecting your application, configure the profile that governs inspecting the traffic to and from this application.

Note: A website that does not access a SQL database usually does not have access to sensitive private information such as credit card numbers.

Using the Command Line to Configure the Credit Card Check

In the command line interface, you can use either the set appfw profile command or the add appfw profile command to activate credit-card checking and specify which actions to perform. You can use the unset appfw profile command to revert back to the default settings. To specify relaxations, use the bind appfw command to bind credit card numbers to the profile.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To configure a Credit Card check by using the command line

Use either the set appfw profile command or the add appfw profile command, as follows:

- set appfw profile <name> -creditCardAction (([**block**] [**learn**] [**log**] [**stats**]) | [**none**])
- set appfw profile <name> -creditCard (**VISA** | **MASTERCARD** | **DISCOVER** | **AMEX** | **JCB** | **DINERSCLUB**)
- set appfw profile <name> -creditCardMaxAllowed <integer>
- set appfw profile <name> -creditCardXOut ([**ON**] | [**OFF**])
- set appfw profile <name> -doSecureCreditCardLogging ([**ON**] | [**OFF**])

To configure a Credit Card relaxation rule by using the command line

Use the bind command to bind the credit card number to the profile. To remove a credit card number from a profile, use the unbind command, with the same arguments that you used for the bind command. You can use the show command to display the credit card numbers bound to a profile.

To bind a credit card number a profile

bind appfw profile <profile-name> -creditCardNumber <any number/regex> “<url>”

Example: bind appfw profile test_profile -creditCardNumber 378282246310005 “http://www.example.com/credit_card_test.html”

To unbind a credit card number from a profile

unbind appfw profile <profile-name> -creditCardNumber <credit card number / regex> “<url>”

To show the list of credit card numbers bound to a profile.

show appfw profile <profile>

Using the Configuration Utility to Configure the Credit Card Check

In the configuration utility, you configure the Credit Card security check in the pane for the profile associated with your application.

To add or modify the Credit Card security check by using the Configuration Utility

1. Navigate to Application Firewall > Profiles, highlight the target profile, and click Edit.
2. In the Advanced Settings pane, click Security Checks.
   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

   1. If you just want to enable or disable Block, Log, Stats, and Learn actions for Credit Card, you can select or clear check boxes in the table, Click OK, and then click Save and Close to close the Security Check pane.
   2. If you want to configure additional options for this security check, double click Credit Card, or select the row and click Action Settings to display additional options as follows:
      - X-Out—Mask any credit card number detected in a response by replacing each digit, except the digits in the final group, with the letter “X.”
      - Maximum credit cards allowed per page—Specify the number of credit cards that can be forwarded to the client without triggering a block action.
      - Protected Credit Cards. Select or clear a check box to enable or disable protection for each type of credit card.
      - You can also edit the Block, Log, Stats and Learn actions in the Credit Card Settings pane.
        After making any of the above changes, click OK to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click OK to save all the changes you have made in the Security Checks section and then click Save and Close to close the Security Check pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Advanced Settings pane, click Profile settings. To enable or disable secure logging of credit card Numbers, select or clear the Secure Credit Card Logging check box. (By default, it is selected.)
   Click OK to save the changes.

To configure a Credit Card relaxation rule by using the Configuration Utility

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click Relaxation Rules. The Relaxation Rules table has a Credit Card entry. You can double click, or select this row and click the Edit button to access the Credit Card Relaxation Rules dialogue. You can perform Add, Edit, Delete, Enable, or Disable operations for relaxation rules.

Using the Learn Feature with the Credit Card Check

When the learn action is enabled, the application firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, if you want to exempt a specific string of digits from the Credit Card security check, you can by deploy the learned rule as a relaxation rule.

To view or use learned data by using the command line interface

show appfw learningdata <profilename> creditCardNumber
rm appfw learningdata <profilename> -creditcardNumber <credit card number> “<url>”

export appfw learningdata <profilename> creditCardNumber

To view or use learned data by using the configuration utility

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Learned Rules**. You can select the Credit Card entry in the Learned Rules table and double-click it to access the learned rules. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

You also have the option to show a summarized view of the learned relaxations by selecting the Credit Card entry in the Learned Rules table and clicking Visualizer to get a consolidated view of all the learned violations. The visualizer makes it very easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the Log Feature with the Credit Card Check

When the log action is enabled, the Credit Card security check violations are logged in the audit log as APPFW_SAFECOMMERCE or APPFW_SAFECOMMERCE_XFORM violations. The application firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

The default setting for doSecureCreditCardLogging is ON. If you change it to OFF, both credit card number and type are included in the log message.

Depending on the settings configured for the Credit Card checks, the application-firewall generated log messages might include the following information:

- Response was blocked or not blocked.
- Credit card numbers were transformed (X’d out). A separate log message is generated for each transformed credit card

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

number, so multiple log messages might be generated during processing of a single response.
- Response contained the maximum number of potential credit card numbers.
- Credit card numbers and their corresponding types.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the Credit Card violations:

- Shell
- tail -f /var/log/ns.log | grep SAFECOMMERCE

To access the log messages by using the configuration utility

1. The Citrix configuration utility includes a very useful tool (Syslog Viewer) for analyzing the log messages. You have a couple of options for accessing the Syslog Viewer: Navigate to the **target profile** > **Security Checks**. Highlight the Credit Card row and click Logs. When you access the logs directly from the Credit Card security check of the profile, it filters out the log messages and displays only the logs pertaining to these security check violations.
2. You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
   The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To access Credit Card security check violation log messages, filter by selecting APPFW in the dropdown options for Module. The Event Type displays a rich set of options to further refine your selection. For example, if you select the APPFW_SAFECOMMERCE and APPFW_SAFECOMMERCE_XFORM check boxes and click the Apply button, only log messages pertaining to the Credit Card security check violations appear in the Syslog Viewer.

   If you place the cursor in the row for a specific log message, multiple options, such as Module and EventType, appear below the log message. You can select any of these options to highlight the corresponding information in the logs.

Example of a Native format log message when the response is not blocked

```
May 29 01:26:31 <local0.info> 10.217.31.98 05/29/2015:01:26:31 GMT ns 0-PPE-0 :
default APPFW APPFW_SAFECOMMERCE 2181 0 :  10.217.253.62 1098-PPE0
4erNfkaHy0IeGP+nv2S9Rsdu77I0000 pr_ffc http://aaron.stratum8.net/FFC/CreditCardMind.html
Maximum number of potential credit card numbers seen <not blocked>
```

Example of a CEF format log message when the response is transformed

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE_XFORM|6|src=10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/CreditCardMind.html
msg=Transformed (xout) potential credit card numbers seen in server response
cn1=66 cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVI80002
cs4=ALERT cs5=2015 act=transformed
```

Example of a CEF format log message when the response is blocked. The credit card number and type can be seen in the log, because the doSecureCreditCardLogging parameter is disabled.

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE|6|src=10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/CreditCardMind.html
msg=Credit Card number 4505050504030302 of type Visa is seen in response cn1=68
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVI80002 cs4=ALERT cs5=2015
act=blocked

Statistics for the Credit Card violations

When the stats action is enabled, the corresponding counter for the Credit Card check is incremented when the application firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled and the Max Allowed credit card setting is 0, the request for a page that contains 20 credit card numbers increments the stats counter by one when the page is blocked as soon as the first credit card number is detected. However, if block is disabled and transform is enabled, processing the same request increments the statistics counter for logs by 20, because each credit card transformation generates a separate log message.

To display Credit Card statistics by using command line

At the command prompt, type:
sh appfw stats

To display stats for a specific profile, use the following command:

stat appfw profile <profile name>

To display Credit Card statistics by using configuration utility

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about Credit Card violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

Note the following points about the Credit Card security check:

- The application firewall enables you to protect credit card information and detect any attempts to access this sensitive data.
- To use the Credit Card protection check, you must specify at least one type of credit card and an action. The check is then applied to HTML, XML, and Web 2.0 profiles.
- You can pipe the output of sh appfw profile command and grep for CreditCard to see all the Credit Card specific configuration. For example, sh appfw profile my_profile | grep CreditCard displays the configured settings of various parameters as well as the relaxation rules pertaining to the Credit Card check for the application firewall profile named my_profile.
- You can exclude specific numbers from Credit Card inspection without bypassing the security check inspection for the rest of the credit card numbers.
- Relaxation is available for all application firewall protected credit card patterns. In the configuration utility, you can use the visualizer to specify Add, Edit, Delete, Enable, or Disable operations on relaxation rules.
- The application firewall learning engine can monitor the outgoing traffic to recommend rules based on observed violations. Visualizer support is also available for managing the learned credit card rules in the configuration utility. You can edit and deploy the learned rules, or skip them after careful inspection.
- The setting for number of allowed credit cards applies to each response. It does not pertain to the cumulative total of credit card numbers observed during the entire user session.
- The number of X'd out digits depends on the length of the credit card numbers. Ten digits are X'd out for credit cards that have 13 through 15 digits. Twelve digits are X'd out for credit cards that have 16 digits. If your application does not require sending the entire credit card number in the response, Citrix recommends that you enable this action to mask the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

digits in the credit card numbers.

- The X-out operation transforms all the credit cards and works independently of the configured settings for the maximum number of allowed credit cards. For example, if there are 4 credit cards in the response and the creditCardMaxAllowed parameter is set to 10, all 4 credit cards are X'd-out, but they are not blocked. If the credit card numbers are spread out in the document, a partial response with X'd-out numbers might be sent to the client before the response is blocked.
- Do not disable the doSecureCreditCardLogging parameter before due consideration. When this parameter is turned off, the credit card numbers are displayed and are accessible in the log messages. These numbers are not masked in the logs, even if the X-out action is enabled. If you are sending logs to a remote syslog server, and the logs are compromised, the credit card numbers can be exposed.
- When the response page is blocked because of a Credit Card violation, the application firewall does not redirect to the error page.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Safe Object Check

Feb 03, 2014

The Safe Object check provides user-configurable protection for sensitive business information, such as customer numbers, order numbers, and country-specific or region-specific telephone numbers or postal codes. A user-defined regular expression or custom plug-in tells the application firewall the format of this information and defines the rules to be used to protect it. If a string in a user request matches a safe object definition, the application firewall either blocks the response, masks the protected information, or removes the protected information from the response before sending it to the user, depending on how you configured that particular safe object rule.

The Safe Object check prevents attackers from exploiting a security flaw in your web server software or on your web site to obtain sensitive private information, such as company credit card numbers or social security numbers. If your web sites do not have access to these types of information, you do not need to configure this check. If you have a shopping cart or other application that can access such information, or your web sites have access to database servers that contain such information, you should configure protection for each type of sensitive private information that you handle and store.

Note: A web site that does not access an SQL database usually does not have access to sensitive private information.
The Safe Object Check dialog box is unlike that for any other check. Each safe object expression that you create is the equivalent of a separate security check, similar to the Credit Card check, for that type of information. If you use the wizard or the configuration utility, you add a new expression by clicking Add and configuring the expression in the Add Safe Object dialog box. You modify an existing expression by selecting it, then clicking Open, and then configuring the expression in the Modify Safe Object dialog box.

In the Safe Object dialog box for each safe object expression, you can configure the following:

- **Safe Object Name.** A name for your new safe object. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 255 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.
- **Actions.** Enable or disable the Block, Log, and Statistics actions, and the following actions:
  - **X-Out.** Mask any information that matches the safe object expression with the letter “X”.
  - **Remove.** Remove any information that matches the safe object expression.
- **Regular Expression.** Enter a PCRE-compatible regular expression that defines the safe object. You can create the regular expression in one of three ways: by typing the regular expression directly into the text box, by using the **Regex Tokens** menu to enter regular expression elements and symbols directly into the text box, or by opening the Regular Expressions Editor and using it to construct the expression. The regular expression must consist of ASCII characters only. Do not cut and paste characters that are not part of the basic 128-character ASCII set. If you want to include non-ASCII characters, you must manually type those characters in PCRE hexadecimal character encoding format.
  Note: Do not use start anchors (^) at the beginning of Safe Object expressions, or end anchors ($) at the end of Safe Object expressions. These PCRE entities are not supported in Safe Object expressions, and if used, will cause your expression not to match what it was intended to match.
- **Maximum Match Length.** Enter a positive integer that represents the maximum length of the string that you want to match. For example, if you want to match U.S. social security numbers, enter the number eleven (11) in this field. That allows your regular expression to match a string with nine numerals and two hyphens. If you want to match California driver’s license numbers, enter the number eight (8).
  Caution: If you do not enter a maximum match length in this field, the application firewall uses a default value of one (1) when filtering for strings that match your safe object expressions. As a result, most safe object expressions fail to match their target strings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

You cannot use the command-line interface to configure the Safe Object check. You must configure it by using either the application firewall wizard or the configuration utility.

Following are examples of Safe Object check regular expressions:

- Look for strings that appear to be U.S. social security numbers, which consist of three numerals (the first of which must not be zero), followed by a hyphen, followed by two more numerals, followed by a second hyphen, and ending with a string of four more numerals:
  [1-9][0-9]{2,2}-[0-9]{2,2}-[0-9]{4,4}
- Look for strings that appear to be California driver's license IDs, which start with a letter and are followed by a string of exactly seven numerals:
  [A-Za-z][0-9]{7,7}
- Look for strings that appear to be Example Manufacturing customer IDs which, consist of a string of five hexadecimal characters (all the numerals and the letters A through F), followed by a hyphen, followed by a three-letter code, followed by a second hyphen, and ending with a string of ten numerals:
  [0-9A-Fa-f]{5,5}-[A-Za-z]{3,3}-[0-9]{10,10}

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write to ensure that they define exactly the type of string you want to add as a safe object definition, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you did not want or expect, such as blocking access to web content that you did not intend to block.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Advanced Form Protection Checks

Mar 28, 2012

The advanced Form Protection checks examine web form data to prevent attackers from compromising your system by modifying the web forms on your web sites or sending unexpected types and quantities of data to your web site in a form.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Field Formats Check

Sep 28, 2017

The Field Formats check verifies the data that users send to your web sites in web forms. It examines both the length and type of data to ensure that it is appropriate for the form field in which it appears. If the application firewall detects inappropriate web form data in a user request, it blocks the request.

By preventing an attacker from sending inappropriate web form data to your web site, the Field Formats check prevents certain types of attacks on your web site and database servers. For example, if a particular field expects the user to enter a phone number, the Field Formats check examines the user-submitted input to ensure that the data matches the format of a phone number. If a particular field expects a first name, the Field Formats check ensures that the data in that field is of a type and length appropriate for a first name. It does the same thing for each form field that you configure it to protect.

This check applies to HTML requests only. It does not apply to XML requests. You can configure Field Format Checks in HTML profiles or Web 2.0 profiles to inspect HTML payload for protecting your applications. The application firewall also supports Field Format Check protection for Google Web Toolkit (GWT) applications.

> ## Note
>
> Field format rules are tightening rules. Adding them to relaxation list from learned data acts as a blocking rule.
>
> To relax field format rules, please remove particular "fieldname" from the fieldformat relaxations list.

The Field Formats check requires that you enable one or more actions. The application firewall examines the submitted inputs and applies the specified actions.

You have the option to set the default field formats to specify Field Type and the minimum and maximum length of data expected in each form field on each web form that you want to protect. You can deploy relaxation rules to configure a Field Format for an individual field of a specific form. Multiple rules can be added to specify the field name, the action URL, and Field Formats. Specify Field Formats to accept different types of inputs in different form fields. The learning feature can provide recommendations for the relaxation rules.

Field Format Actions—You can enable Block, Log, Stats, and Learn actions. At least one of these actions must be enabled to engage the Field Format Check protection.

- **Block**. If you enable block, the block action is triggered if the input does not conform to the specified Field Format. If a rule was configured for the target field, the input is checked against the specified rule. Otherwise, it is checked against the default field format specification. Any mismatch in the Field Type or min/max length specification results in blocking the request.
- **Log**. If you enable the log feature, the Field Format check generates log messages indicating the actions that it takes. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate malicious attempts to launch an attack.
- **Stats**. If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack, or you might have to revisit the configuration to see if the specified field format is too restrictive.
- **Learn**. If you are not sure which Field Types or minimum and maximum length values might be ideally suited for your application, you can use the learn feature to generate recommendations based on the learned data. The application

© 1999-2017 Citrix Systems, Inc. All rights reserved.

firewall learning engine monitors the traffic and provides field format recommendations based on the observed values. To get optimal benefit without compromising performance, you might want to enable the learn option for a short time to get a representative sample of the rules, and then deploy the rules and disable learning.

Note: The application firewall's learning engine can distinguish only the first 128 bytes of the name. If a form has multiple fields with names that match for the first 128 bytes, the learning engine might not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields.

Default Field Format—In addition to configuring the actions, you can configure the default Field Format to specify the type of data expected in all the form fields for your application. A Field Type can be selected as the Field Format type. Minimum length and Maximum length parameters can be used to specify the length of the allowed inputs. As an alternative to Field Types, you can use Character Maps to specify what's allowed in a field (except in cluster deployments).

- **Field Type**—Field Types are named expression to which you assign assigned priority values. Field Type expressions specify the allowed inputs and are matched against the submitted data to determine whether the received values are consistent with the allowed values. The Field Types are checked in the order of their priority numbers. A lower number indicates a higher priority. The application firewall gives you the option to add your own Field Types and assign them the priorities you want. The priority value can range from 0 through 64000. The following built-in Field Types are provided to help simplify the configuration process:

```
> sh appfw fieldtype
1)    Name: integer        Regex: "^[+-]?[0-9]+$"
      Priority: 30         Comment: Integer
      Builtin: IMMUTABLE
2)    Name: alpha          Regex: "^[a-zA-Z]+$"
      Priority: 40         Comment: "Alpha characters"
      Builtin: IMMUTABLE
3)    Name: alphanum       Regex: "^[a-zA-Z0-9]+$"
      Priority: 50         Comment: "Alpha-numeric characters"
      Builtin: IMMUTABLE
4)    Name: nohtml         Regex: "^[^&<>]*$"
      Priority: 60         Comment: "Not HTML"
      Builtin: IMMUTABLE
5)    Name: any            Regex: "^.*$"
      Priority: 70         Comment: Anything
      Builtin: IMMUTABLE
 Done
>
```

  Note: The built-in Field Types are IMMUTABLE. They cannot be modified or removed. Any Field Types that you add are MODIFIABLE. You can edit them or remove them.

  Configuring a Field Type as a default Field Format might be useful when you have a PCRE expression that can identify the valid inputs in all or most of the form fields for your application and exclude the invalid inputs. For example, if all the inputs in your application forms are expected to contain only numbers and letters, you might want to use the built-in Field Type alphanum as the default Field Type. Any non-alphanumeric character such as a backslash (\) or semicolon (;) in the input will trigger a violation. You can also add your own customized Field Types and use them to configure default Field Formats. For example, if you want to make the lowercase "x", "y", and "z" the only allowed alpha characters, you can configure a customized Field Type with regular expression "^[x-z]+$". You could assign it a higher priority (lower priority number) then the built-in Field Types and use it as the default Field Type.

- **Minimum Length**—The default minimum data length assigned to form fields in web forms that do not have an explicit

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

setting. This parameter is set to 0 by default, which allows the user to leave the field blank. Any higher setting forces users to fill in the field.
Caution: If the minimum length value is 0 but the Field Type is integer, alpha, or alphanum, a request is blocked if any input field is left empty, despite the minimum length setting. That is because the regEx for these Field Types contains a + character, which means one or more characters. Distinguishing an integer from an alpha character requires at least one character.

- **Maximum Length**—The default maximum data length assigned to form fields in web forms that do not have an explicit setting. This parameter is set to 65535 by default.
  Note: Characters vs bytes. The minimum and the maximum lengths for the field formats represent the number of bytes, not the number of characters. Languages that have greater than one-byte character representation can cause the limit to be exceeded with fewer characters than the number configured for the maximum value. For example, with double-byte character representation, the maximum value of 9 allows no more than 4 characters.

  Tip: The configuration utility allows you to cut and paste UTF-8 characters directly into the GUI without having to convert them to hex.

- **Character Maps**: In addition to recommending the Field Types, the application firewall learning engine offers you an additional option, Use Character Maps, to deploy the Format Check rules. A Character Map is a set of all the characters allowed in a particular form field. You can fine tune the Field format specification to allow or disallow specific characters by using Character Maps. A separate Character Map is generated for each form field. The alpha and numeric characters are treated differently in Character Maps. If any alpha character is seen in the input, all alpha characters [A-Za-z] will be allowed by the recommended PCRE expression in the Character Map. Similarly, if any digit is included, all digits [0-9] will be allowed. Non-printable characters are specified by using the \x construct. Only single Byte characters with values between 0-255 are considered for Character Map recommendations.
  A Character Map can be more specific than the corresponding Field Type recommendation. In some situations, Character Maps might be a better option, because they give you tighter control over the set of characters allowed as inputs. The deployed character maps are displayed as strings that start with prefix “CM” followed by digits. The priority for the Character Maps starts at 10000. As with user-added Field Types, you can add, edit or remove a Character Map. Character Maps that are currently used in deployed rules cannot be modified or removed.

  Note: Character Maps are not supported in cluster deployments.

Using the Command Line to Configure the Field Format Check

In the command line interface, you can use the add appfw fieldtype command to add a new Field Type. You can use either the set appfw profile command or the add appfw profile command to configure the Field Format check and specify which actions to perform. You can use the unset appfw profile command to revert the configured settings back to their defaults. To specify a Field Format rule, use the bind appfw command to bind a Field Type to a form Field and the action URL, along with the minimum and maximum length specifications.

To add, remove or view a Field Type by using the command line

Use the add command to add a Field Type. You must specify the Name, Regular expression and Priority when adding a new Field Type. You also have the option to add a Comment. You can use the show command to display the configured Field Types. You can also delete a Field Type by using the remove command, which requires only the Name of the Field Type.

add [appfw] fieldType <name> <regex> <priority> [-comment <string>]where:

<regex> is a reqular expression

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

<priority> is a positive_integer

Example: add fieldtype "Cust_Zipcode" "^[0-9]{5}[-][0-9]{4}$" 4

- show [appfw] fieldType [<name>]
  Example: sh fieldtype

  sh appfw fieldtype

  sh appfw fieldtype cust_zipcode

- rm [appfw] fieldType <name>
  Example: rm fieldtype cusT_ziPcode

  rm appfw fieldtype cusT_ziPcode

Note: As shown above, use of “appfw” in the command is optional. For example, “Add FieldType” or “Add appfw fieldType” are both valid options. The names of the Field Types are case insensitive due to normalization. As shown in the above examples, Cust_Zipcode, cust_zipcode, and cUsT_ziPcode refer to the same Field Type.
To configure a Field Format check by using the command line

Use either the set appfw profile command or the add appfw profile command, as follows:
- set appfw profile <name> -fieldFormatAction (([block] [learn] [log] [stats]) | [none])
- set appfw profile <name> -defaultFieldFormatType <string>
- set appfw profile <name> -defaultFieldFormatMinLength <integer>
- set appfw profile <name> -defaultFieldFormatMaxLength <integer>

To configure a Field Format relaxation rule by using the command line

bind appfw profile <name> (-fieldFormat <string> <formActionURL> <fieldType>
[-fieldFormatMinLength <positive_integer>] [-fieldFormatMaxLength <positive_integer>]
[-isRegex (REGEX | NOTREGEX)])
Example:

bind appfw profile pr_ffc -fieldFormat "login_name" ".*/login.php" integer -fieldformatMinLength 3 -FieldformatMaxlength 6

Using the Configuration Utility to Configure the Field Formats Security Check

In the configuration utility, you can manage the Field Types. You can also configure the Field Formats security check in the pane for the profile associated with your application.

To add, modify or remove a Field Type using the Configuration Utility
1. Navigate to application firewall node. In the Settings, click **Manage Field Types** to display the Configure Application Firewall Field Type dialogue box.
2. Click **Add** to add a new Field Type. Follow the instructions in this pane and click Create. You can also edit or delete any user-added Field Type if it is currently not being used by a deployed rule.

To add or modify the Field Formats security check by using the Configuration Utility
1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Security Checks**.
   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. If you just want to enable or disable **Block**, **Log**, **Stats**, and **Learn** actions for Field Formats, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the Security Check pane.
2. If you want to configure additional options for this security check, double click Field Formats, or select the row and click Action Settings, to display the following options for **Default Field Format**:
   - **Field Type**—Select the Field Type that you want to configure as the default Field Type. You can select the built-in and user-defined Field Types. The deployed Character Maps are also included in the list and can be selected.
   - **Minimum Length**—Specify the minimum number of characters that must be in each field. Possible values: 0-65535.
   - **Maximum Length**— Specify the maximum number of characters that must be in each field. Possible values: 1-65535.
     You can also edit the **Block**, **Log**, **Stats** and **Learn** actions in the Field Formats Settings pane.

After making any of the above changes, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click OK to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

To configure a Field Formats relaxation rule by using the Configuration Utility

1. Navigate to **Application Firewall** > Profiles, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**. The Relaxation Rules table has a Field Formats entry. You can double click, or select this row and click the Edit button, to access the Field Formats Relaxation Rules dialogue. You can perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.
   For a consolidated view of all the relaxation rules, you can highlight the Field Formats row and click Visualizer. The visualizer for deployed relaxations offers you the option to Add a new rule or Edit an existing one. You can also Enable or Disable a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

Using the Learn Feature with the Field Formats Check

When the learn action is enabled, the application firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, you can deploy the learned rule as a Field Format relaxation rule.

Field Formats Learning enhancement—An application firewall learning enhancement was introduced in release 11.0. In the previous releases, once the learned field format recommendation is deployed, the application firewall learning engine stops monitoring the valid requests for the purpose of recommending new rules on the basis of the new data points. This limits the configured security protection, because the learning database does not include any representations of the new data seen in the valid requests processed by the security check.

Violations are no longer coupled with learning. The learning engine learns and makes recommendations for the field formats regardless of the violations. In addition to checking the blocked requests to determine whether the current field format is too restrictive and needs to be relaxed, the learning engine also monitors the allowed requests to determine whether the current field format is too permissive, and allows elevating the security by deploying a more restrictive rule.

Following is a summary of the Field Formats learning behavior:

No Field format is bound—The behavior remains unchanged in this scenario. All learn data is sent to the aslearn engine. The learning engine suggests a field format rule based on the data set.

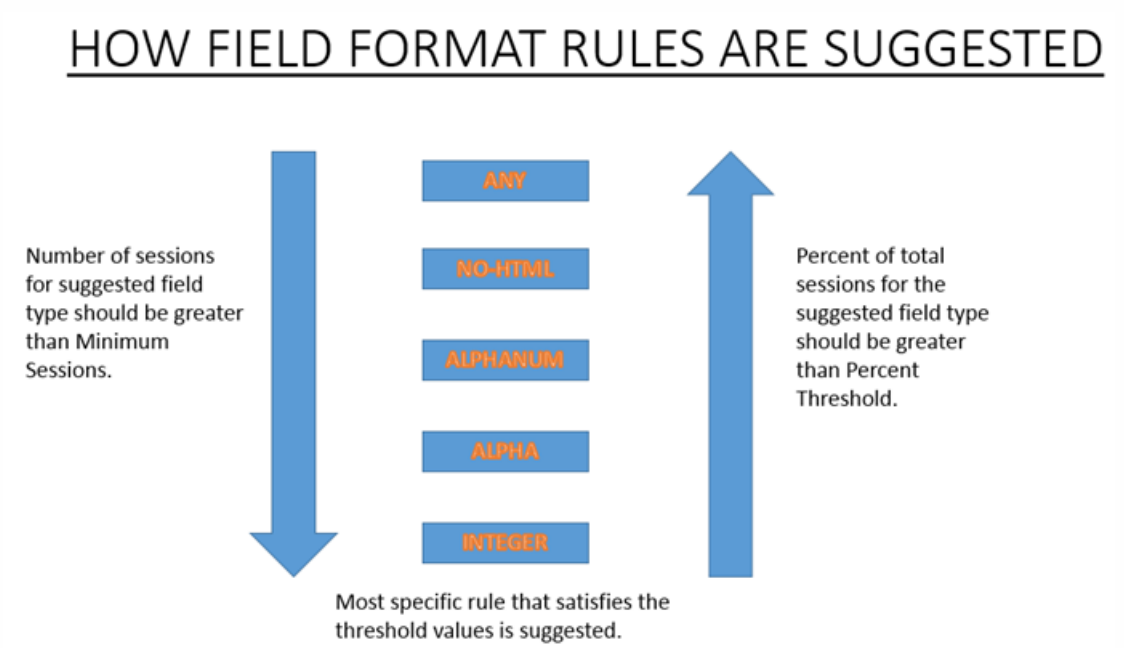
Field format is bound: In the previous releases, observed data is sent to the aslearn engine only in the case of a violation. The learning engine suggests a field format rule based on the data set. In the 11.0 release, all data is sent to aslearn engine even if no violation is triggered. The learning engine suggests a field format rule based on the entire data set of all received

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

inputs.

Use Case for learning enhancement:

If the initial field format learned rules are based on a small sample of data, a few non typical values might result in a recommendation that is too lenient for the target field. The ongoing learning allows the application firewall to observe data points from every request to collect a representative sample for the learned recommendations. This is helpful in further tightening the security to deploy the optimal input format with an adequate range value.



The field format learning makes use of the priority of the Field Types as well as the configured settings of the following learning thresholds:

- **FieldFormatMinThreshold**— Minimum number of times a specific form field must be observed before a learned relaxation is generated. Default: 1.
- **FieldFormatPercentThreshold**— Percentage of times a form field matched a particular Field Type, before a learned relaxation is generated. Default: 0.

The field format rule recommendations are based on the following criteria:

- **Field Type recommendations**— The Field Type recommendations are determined by the assigned priorities of the existing Field Types and the specified Field Format thresholds. The priorities determine the order in which the Field Types are matched against the inputs. A lower number specifies a higher priority. For example, Field Type integer has the higher priority (30) and is therefore evaluated before Field Type alphanum (50). The thresholds determine the number of inputs evaluated to collect a representative sample for the data point. Assigning the right priority to the configured Field Types, and configuring an appropriate **learningsetting** value for the **fieldFormatPercentThreshold** and **fieldFormatMinThreshold** parameters, is essential for getting the correct Field Format recommendation. The Field Type with the highest priority, based on the configured thresholds, is matched first against the inputs. If there is a match, this Field Type is suggested without considering the other Field Types. For example, three default Field Types, integer, alphanum, and any will match if all the inputs contain only numbers. However, integer will be recommended since it has the highest priority.
- **Minimum and Maximum length recommendations**— Calculations for the minimum and maximum lengths for the Field Format are done independently of the determination for the Field Type. The field format length calculations are based on the average length of all the observed inputs. Half of this calculated average is suggested as the min value, and twice

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

the value of this average is suggested as the max value. The range for the Minimum Length is 0-65535 and the range for the Maximum Length is 1-65535. The configured value for the minimum length cannot exceed the maximum length.

- **Handling of space character**—The Field Format check counts every space character when checking for the Field Formats length. Leading or trailing spaces are not stripped, and multiple consecutive spaces in the middle of the input string are no longer consolidated to a single space during input processing.

Example to illustrate the Field Format recommendations:
Total requests: 100
Number of Req with Field Type:
Int : 22 (22 int values) – 22%
Alpha : 44 (44 alpha values) – 44%
Alphanum: 14 (14 + 44 + 22 = 80 alphanum values) = 80%
noHTML: 10 (80 + 10 = 90 noHTML values) = 90%
any : 10 (90 + 10 = 100 any values) = 100%

% threshold	Suggested Field Type
0-22	int
23-44	alpha
45-80	alphanum
81-90	noHTML
91-100	any

To view or use learned data by using the command line interface
show appfw learningdata <profilename> FieldFormat
rm appfw learningdata <profilename> -fieldFormat <string> <formActionURL>
export appfw learningdata <profilename> FieldFormat
To view or use learned data by using the configuration utility

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click Edit.
2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the Field Formats entry in the Learned Rules table and double-click it to access the learned rules. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.
   You also have the option to show a summarized view of the learned relaxations by selecting the Field Formats entry in the Learned Rules table and clicking Visualizer to get a consolidated view of all the learned violations. The visualizer makes it very easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the Log Feature with the Field Formats Check

When the log action is enabled, the Field Formats security check violations are logged in the audit log as APPFW_FIELDFORMAT violations. The application firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the Field Formats

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

violations:

- Shell
- tail -f /var/log/ns.log | grep APPFW_FIELDFORMAT

To access the log messages by using the configuration utility

The Citrix configuration utility includes a very useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the Application Firewall > Profiles, select the target profile, and click Security Checks. Highlight the Field Formats row and click Logs. When you access the logs directly from the Field Formats security check of the profile, it filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to NetScaler > System > Auditing. In the Audit Messages section, click the Syslog messages link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to Application Firewall > policies > Auditing. In the Audit Messages section, click the Syslog messages link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.
  The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To access Field Formats security check violation log messages, filter by selecting APPFW in the dropdown options for Module. The Event Type displays a rich set of options to further refine your selection. For example, if you select the APPFW_FIELDFORMAT check box and click the Apply button, only log messages pertaining to the Field Formats security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as Module and EventType, appear below the log message. You can select any of these options to highlight the corresponding information in the logs.

Example of a Native format log message when the request is not blocked
Jun 10 22:32:26 <local0.info> 10.217.31.98 06/10/2015:22:32:26 GMT ns 0-PPE-0 :
default APPFW APPFW_FIELDFORMAT 97 0 : 10.217.253.62 562-PPE0
x1MV+YnNGzQFM3Bsy2wti4bhXio0001 pr_ffc http://aaron.stratum8.net/FFC/login_post.php
Field format check failed for field passwd="65568888sz-*_" <not blocked>
Example of a CEF format log message when the request is blocked
Jun 11 00:03:51 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_FIELDFORMAT|6|src=10.217.253.62 spt=27076
method=POST requet=http://aaron.stratum8.net/FFC/maxlen_post.php msg=Field format check
failed for field text_area\="" cn1=108 cn2=644 cs1=pr_ffc cs2=PPE0
cs3=GaUROfl1Nx1jJTvja5twH5BBql0000 cs4=ALERT cs5=2015 act=blocked

Statistics for the Field Formats violations

When the stats action is enabled, the corresponding counter for the Field Formats check is incremented when the application firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, the request for a page that contains 3 Field Format violations increments the stats counter by one, because the page is blocked as soon as the first Field Formats violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by 3, because each Field Formats violation generates a separate log message.

To display Field Formats statistics by using command line

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type:

```
sh appfw stats
```

To display stats for a specific profile, use the following command:

```
stat appfw profile <profile name>
```

To display Field Formats statistics by using configuration utility

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the Statistics Link.
3. Use the scroll bar to view the statistics about Field Formats violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Deployment Tip

- Enable Field format actions log, learn and stats.
- After running a representative sample of the traffic to your application, review the learned recommendations.
- If a Field Type is recommended by most of the learned rules, configure that Field Type as the Default Field Type. For minimum and maximum lengths, use the widest range suggested by these rules.
- Deploy rules for other fields for which different Field Types or different minimum/maximum lengths are better suited.
- Enable blocking and disable learning.
- Monitor stats and logs. If a significant number of violations are still being triggered, you might want to review the log messages to confirm that the violations represent malicious requests that should have been blocked. If valid requests are being flagged as violations, you can either edit the configured Field Format rule to further relax it or enable learning again to get recommendations based on the new data points.

Note: You can fine tune your configuration by getting new learning recommendations.

Highlights

Note the following points about the Field Format security check:

- **Protection**—By configuring optimal field format rules, you can protect against many attacks. For example, if you specify that a field can only have integers, hackers will not be able to launch SQL Injection or XSS attacks by using this field, because the inputs required to launch such attacks will not meet the configured field format requirement.
- **Performance**—You can limit the minimum and the maximum allowed length for the inputs in the field format rules. This can prevent a malicious user from entering excessively large input strings in an attempt to add processing overhead to the server, or worse, cause the server to dump core because of stack overflow. By limiting the input size, you can shorten the time required for processing legitimate requests.
- **Configuring Field Formats**—You must enable one of the actions (block, log, stats, learn) to engage the field Format protection. You can also specify the Field format rules to identify the allowed inputs in your form fields.
- **Selecting Character Maps vs. Field Types**—Both Character Maps and Field Types use regular expressions. However, a Character Map provides a more specific expression by narrowing down the list of allowed characters. For example, for an input such as janedoe@citrix.com, the learning engine might recommend the Field Type nohtml but the Character Map [.@-Za-z] might be more specific, because it narrows down the allowed set of non-alpha characters. The Character Map option allows, in addition to alpha characters, only two non-alpha characters: period (.) and at (@).
- **Ongoing Learning**—The application firewall monitors and takes into account all the incoming data (violations as well as allowed inputs) to build a learning table for recommending rules. The rules are revised and updated as new incoming data arrives. New field format rules are suggested for a field even if it already has a bound field format rule. If the configured Field Formats are too restrictive and are blocking the valid requests, you can deploy a more relaxed Field Format. Similarly,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

if the current Field Formats are too generic, you can further refine and tighten the security by deploying a more restrictive Field Format.

- **Overwriting Rules**—If a rule has already been deployed for a field/URL combination, the GUI allows the user to update the field format. A dialog box asks for confirmation to replace the existing rule. If you are using the command line interface, you have to explicitly unbind the previous binding and then bind the new rule.
- **Multiple match**—If multiple field formats match a given field name and its action URL, the application firewall arbitrarily selects one of them to apply.
- **Buffer boundary**—If a field value extends across multiple streaming buffers, and the format for these two parts of the field value is different, a field format corresponding to “any” is sent to the learn database.
- **Field Format vs. Field Consistency Check**—Both the Field Format check and the Field Consistency check are form-based protection checks. The Field Formats check provides a different type of protection than does the Form Field Consistency check. The Form Field Consistency check verifies that the structure of the web forms returned by users is intact, that data format restrictions configured in the HTML are respected, and that data in hidden fields has not been modified. It can do this without any specific knowledge about your web forms other than what it derives from the web form itself. The Field Formats check verifies that the data in each form field matches the specific formatting restrictions that you configured manually, or that the learning feature generated and you approved. In other words, the Form Field Consistency check enforces general web form security, while the Field Formats check enforces the specific rules for the allowed inputs for your web forms.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Form Field Consistency Check

Feb 05, 2014

The Form Field Consistency check examines the web forms returned by users of your web site, and verifies that web forms were not modified inappropriately by the client. This check applies only to HTML requests that contain a web form, with or without data. It does not apply to XML requests.

The Form Field Consistency check prevents clients from making unauthorized changes to the structure of the web forms on your web site when they fill out and submit a form. It also ensures that the data a user submits meets the HTML restrictions for length and type, and that data in hidden fields is not modified. This prevents an attacker from tampering with a web form and using the modified form to gain unauthorized access to web site, redirect the output of a contact form that uses an insecure script and thereby send unsolicited bulk email, or exploit a vulnerability in your web server software to gain control of the web server or the underlying operating system. Web forms are a weak link on many web sites and attract a wide range of attacks.

The Form Field Consistency check verifies all of the following:

- If a field is sent to the user, the check ensures that it is returned by the user.
- The check enforces HTML field lengths and types.
  Note: The Form Field Consistency check enforces HTML restrictions on data type and length but does not otherwise validate the data in web forms. You can use the Field Formats check to set up rules that validate data returned in specific form fields on your web forms.
- If your web server does not send a field to the user, the check does not allow the user to add that field and return data in it.
- If a field is a read-only or hidden field, the check verifies that the data has not changed.
- If a field is a list box or radio button field, the check verifies that the data in the response corresponds to one of the values in that field.

If a web form returned by a user violates one or more of the Form Field consistency checks, and you have not configured the application firewall to allow that web form to violate the Form Field Consistency checks, the request is blocked.

If you use the wizard or the configuration utility, in the Modify Form Field Consistency Check dialog box, on the General tab you can enable or disable the Block, Log, Learn, and Statistics actions.

You also configure Sessionless Field Consistency in the General tab. If Sessionless Field Consistency is enabled, the application firewall checks only the web form structure, dispensing with those parts of the Form Field Consistency check that depend upon maintaining session information. This can speed the Form Field Consistency check with little security penalty for web sites that use many forms. To use Sessionless Field Consistency on all web forms, select On. To use it only for forms submitted with the HTTP POST method, select postOnly

If you use the command-line interface, you can enter the following command to configure the Form Field Consistency Check:

- set appfw profile <name> -fieldConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]

To specify relaxations for the Form Field Consistency check, you must use the configuration utility. On the Checks tab of the Modify Form Field Consistency Check dialog box, click Add to open the Add Form Field Consistency Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Form Field Consistency Check Relaxation

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dialog box. Either dialog box provides the same options for configuring a relaxation, as described in "Manual Configuration By Using the Configuration Utility."

Following are examples of Form Field Consistency check relaxations:

Form Field Names

- Choose form fields with the name UserType:

  ^UserType$
- Choose form fields with names that begin with UserType_ and are followed by a string that begins with a letter or number and consists of from one to twenty-one letters, numbers, or the apostrophe or hyphen symbol:

  ^UserType_[0-9A-Za-z][0-9A-Za-z'-]{0,20}$
- Choose form fields with names that begin with Turkish-UserType_ and are otherwise the same as the previous expression, except that they can contain Turkish special characters throughout:

  ^T\xC3\xBCrk\xC3\xA7e-UserType_([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])+$
  Note: See "PCRE Character Encoding Format" for a complete description of supported special characters and how to encode them properly.
- Choose form field names that begin with a letter or number, consist of a combination of letters and/or numbers only, and that contain the string Num anywhere in the string:

  ^[0-9A-Za-z]*Num[0-9A-Za-z]*$

Form Field Action URLs

- Choose URLs beginning with http://www.example.com/search.pl? and containing any string after the query except for a new query:

  ^http://www[.]example[.]com/search[.]pl\?[^?]*$
- Choose URLs that begin with http://www.example-español.com and have paths and filenames that consist of upper-case and lower-case letters, numbers, non-ASCII special characters, and selected symbols in the path. The ñ character and any other special characters are represented as encoded UTF-8 strings containing the hexadecimal code assigned to each special character in the UTF-8 charset:

  ^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
  ([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
  ([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
- Choose all URLs that contain the string /search.cgi?:

  ^[^?<>]*/search[.]cgi\?[^?<>]*$

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the Cookie Consistency check would otherwise have blocked.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

CSRF Form Tagging Check

Feb 05, 2014

The Cross Site Request Forgery (CSRF) Form Tagging check tags each web form sent by a protected web site to users with a unique and unpredictable FormID, and then examines the web forms returned by users to ensure that the supplied FormID is correct. This check protects against cross-site request forgery attacks. This check applies only to HTML requests that contain a web form, with or without data. It does not apply to XML requests.

The CSRF Form Tagging check prevents attackers from using their own web forms to send high volume form responses with data to your protected web sites. This check requires relatively little CPU processing capacity compared to certain other security checks that analyze web forms in depth. It is therefore able to handle high volume attacks without seriously degrading the performance of the protected web site or the application firewall itself.

Before you enable the CSRF Form Tagging check, you should be aware of the following:

- You need to enable form tagging. The CSRF check depends on form tagging and does not work without it.
- You should disable the Citrix NetScaler Integrated Caching feature for all web pages containing forms that are protected by that profile. The Integrated Caching feature and CSRF form tagging are not compatible.
- You should consider enabling Referer checking. Referer checking is part of the Start URL check, but it prevents cross-site request forgeries, not Start URL violations. Referer checking also puts less load on the CPU than does the CSRF Form Tagging check. If a request violates Referer checking, it is immediately blocked, so the CSRF Form Tagging check is not invoked.
- The CSRF Form Tagging check does not work with web forms that use different domains in the form-origin URL and form-action URL. For example, CSRF Form Tagging cannot protect a web form with a form-origin URL of http://www.example.com/ and a form action URL of http://www.example.org/form.pl, because example.com and example.org are different domains.

If you use the wizard or the configuration utility, in the Modify CSRF Form Tagging Check dialog box, on the General tab you can enable or disable the Block, Log, Learn and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the CSRF Form Tagging Check:

- set appfw profile <name> -fieldConsistencyAction [**block**] [**log**] [**learn**] [**stats**] [**none**]

To specify relaxations for the CSRF Form Tagging check, you must use the configuration utility. On the Checks tab of the Modify CSRF Form Tagging Check dialog box, click Add to open the Add CSRF Form Tagging Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify CSRF Form Tagging Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

Following are examples of CSRF Form Tagging check relaxations:

Note: The following expressions are URL expressions that can be used in both the Form Origin URL and Form Action URL roles.

- Choose URLs beginning with http://www.example.com/search.pl? and containing any string after the query, except for a new query:

  ^http://www[.]example[.]com/search[.]pl\?[^?]*$
- Choose URLs that begin with http://www.example-español.com and have paths and filenames that consist of upper-case and lower-case letters, numbers, non-ASCII special characters, and selected symbols in the path. The ñ character

© 1999-2017 Citrix Systems, Inc. All rights reserved.

and any other special characters are represented as encoded UTF-8 strings containing the hexadecimal code assigned to each special character in the UTF-8 charset:

```
^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
```

- Choose all URLs that contain the string /search.cgi?:

```
^[^?<>]*/search[.]cgi\?[^?<>]*$
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the check would otherwise have blocked.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing CSRF Form Tagging Check Relaxations

Sep 03, 2013

You configure an exception (or relaxation) to the CSRF Form Tagging security check in the Add Cross-Site Request Forgery Tagging Check Relaxation dialog box or the Modify Cross-Site Request Forgery Tagging Check Relaxation dialog box.

To configure a CSRF Form Tagging check relaxation by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the Profiles pane, select the profile you want to configure, and then click Open.
3. In the Configure Application Firewall Profile dialog box, click the Security Checks tab. The Security Checks tab contains the list of application firewall security checks.
4. In the Security Checks window, click CSRF Form Tagging, and then click Open. The Modify Cross-Site Request Forgery Tagging Check dialog box is displayed, with the Checks tab selected. The Checks tab contains a list of existing CSRF relaxations. The list might be empty if you have not either manually added any relaxations or approved any relaxations that were recommended by the learning engine. Beneath the list is a row of buttons that allow you to add, modify, delete, enable, or disable the relaxations on the list.
5. To add or modify a CSRF relaxation, do one of the following:
   - To add a new relaxation, click Add.
   - To modify an existing relaxation, select the relaxation that you want to modify, and then click Open.

   The Add Cross-Site Request Forgery Tagging Check Relaxation or Modify Cross-Site Request Forgery Tagging Check Relaxation dialog box is displayed. Except for the title, these dialog boxes are identical.
6. Fill in the dialog box as described below.
   - **Enabled check box**—Select to place this relaxation or rule in active use; clear to deactivate it.
   - **Form Origin URL**—In the text area, enter a PCRE-format regular expression that defines the URL that hosts the form.
   - **Form Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the form is delivered.
   - **Comments**—In the text area, type a comment. Optional.

   Note: For any element that requires a regular expression, you can type the regular expression, use the Regex Tokens menu to insert regular expression elements and symbols directly into the text box, or click Regex Editor to open the Add Regular Expression dialog box, and use it to construct the expression.
7. Click OK. The Add Cross-Site Request Forgery Tagging Check Relaxation or Modify Cross-Site Request Forgery Tagging Check Relaxation dialog box closes and you return to the Modify Cross-Site Request Forgery Tagging Check dialog box.
8. To remove a relaxation or rule, select it, and then click Remove.
9. To enable a relaxation or rule, select it, and then click Enable.
10. To disable a relaxation or rule, select it, and then click Disable.
11. To configure the settings and relationships of all existing relaxations in an integrated interactive graphic display, click Visualizer, and use the display tools.
12. To review and configure learned rules for the CSRF check, click Learning and perform the steps in "To configure and use the Learning feature."
13. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL Protection Checks

Mar 28, 2012

The URL Protection checks examine request URLs to prevent attackers from aggressively attempting to access multiple URLs (forceful browsing) or using a URL to trigger a known security vulnerability in web server software or web site scripts.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Start URL Check

Feb 03, 2014

The Start URL check examines the URLs in incoming requests and blocks the connection attempt if the URL does not meet the specified criteria. To meet the criteria, the URL must match an entry in the Start URL list, unless the Enforce URL Closure parameter is enabled. If you enable this parameter, a user who clicks a link on your Web site is connected to the target of that link.

The primary purpose of the Start URL check is to prevent repeated attempts to access random URLs on a Web site, (forceful browsing) through bookmarks, external links, or jumping to pages by manually typing in the URLs to skip the pages required to reach that part of the website. Forceful browsing can be used to trigger a buffer overflow, find content that users were not intended to access directly, or find a back door into secure areas of your Web server. The application firewall enforces a website's given traversal or logic path by allowing access to only the URL's that are configured as start URLs.

If you use the wizard or the configuration utility, in the Modify Start URL Check dialog box, on the General tab you can enable or disable Block, Log, Statistics, Learn actions, and the following parameters:

- **Enforce URL Closure.** Allow users to access any web page on your web site by clicking a hyperlink on any other page on your web site. Users can navigate to any page on your web site that can be reached from the home page or any designated start page by clicking hyperlinks.
  Note: The URL closure feature allows any query string to be appended to and sent with the action URL of a web form submitted by using the HTTP GET method. If your protected web sites use forms to access an SQL database, make sure that you have the SQL injection check enabled and properly configured.
- **Sessionless URL Closure.** From the client's point of view, this type of URL closure functions in exactly the same way as standard, session-aware URL Closure, but uses a token embedded in the URL instead of a cookie to track the user's activity, which consumes considerably fewer resources. When sessionless URL closure is enabled, the application firewall appends a "as_url_id" tag to all the URL's that are in URL closure.
  Note: When enabling sessionless (Sessionless URL Closure), you must also enable regular URL closure (Enforce URL Closure) or sessionless URL closure does not work.
- **Validate Referer Header.** Verify that the Referer header in a request that contains web form data from your protected web site instead of another web site. This action verifies that your web site, not an outside attacker, is the source of the web form. Doing so protects against cross-site request forgeries (CSRF) without requiring form tagging, which is more CPU-intensive than header checks. The application firewall can handle the HTTP Referer header in one of the following four ways, depending on which option you select in the drop-down list:
  - **Off**—Do not validate the Referer header.
  - **If-Present**—Validate the Referer header if a Referer header exists. If an invalid Referer header is found, the request generates a referer-header violation. If no Referer header exists, the request does not generate a referer-header violation. This option enables the application firewall to perform Referer header validation on requests that contain a Referer header, but not block requests from users whose browsers do not set the Referer header or who use web proxies or filters that remove that header.
  - **Always Except Start URLs**—Always validate the Referer header. If there is no Referer header and the requested URL is not exempted by the startURL relaxation rule, the request generates a referer-header violation. If the Referer header is present but it is invalid, the request generates a referer-header violation.
  - **Always Except First Request**—Always validate the referer header. If there is no referer header, only the URL that is accessed first is allowed. All other URL's are blocked without a valid referer header. If the Referer header is present but it is invalid, the request generates a referer-header violation.

One Start URL setting, **Exempt Closure URLs from Security Checks**, is not configured in the Modify Start URL Check dialog

© 1999-2017 Citrix Systems, Inc. All rights reserved.

box, but is configured in the Settings tab of the Profile. If enabled, this setting directs the application firewall not to run further form based checks (such as Cross-Site Scripting and SQL Injection inspection) on URLs that meet the URL Closure criteria.

> **Note**
>
> Although the referer header check and Start URL security check share the same action settings, it is possible to violate the referer header check without violating the Start URL check. The difference is visible in the logs, which log referer header check violations separately from Start URL check violations.

The Referer header settings (OFF, if-Present, AlwaysExceptStartURLs, and AlwaysExceptFirstRequest) are arranged in order of least restrictive to most restrictive and work as follows:

OFF:

- Referer Header Not checked.

If-Present:

- Request has no referer header -> Request is allowed.
- Request has referer header and the referer URL is in URL closure -> Request is allowed.
- Request has referer header and the referer URL is **not** in URL closure -> Request is blocked.

AlwaysExceptStartURLs:

- Request has no referer header and the request URL is a start URL -> Request is allowed.
- Request has no referer header and the request URL is not a start URL ->Request is blocked.
- Request has referer header and the referer URL is in URL closure -> Request is allowed.
- Request has referer header and the referer URL is **not** in URL closure -> Request is blocked.

AlwaysExceptFirstRequest:

- Request has no referer header and is the first request URL of the session -> Request is allowed.
- Request has no referer header and is **not** the first request URL of the session -> Request is blocked.
- Request has referer header and is either the first request URL of the session or is in URL closure -> Request is allowed.
- Request has referer header and is neither the first request URL of the session nor is in URL closure -> Request is blocked.

If you use the command-line interface, you can enter the following commands to configure the Start URL Check:

- set appfw profile <name> -startURLAction [**block**] [**learn**] [**log**] [**stats**] [**none**]
- set appfw profile <name> -startURLClosure ([**ON**] | [**OFF**])
- set appfw profile <name> -sessionlessURLClosure ([**ON**] | [**OFF**])
- set appfw profile <name> -exemptClosureURLsFromSecurityChecks ([**ON**] | [**OFF**])
- set appfw profile <name> -RefererHeaderCheck ([**OFF**] | [**if-present**] | [**AlwaysExceptStartURLs**] | [**AlwaysExceptFirstRequest**])

To specify relaxations for the Start URL check, you must use the configuration utility. On the Checks tab of the Modify Start URL Check dialog box, click Add to open the Add Start URL Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Start URL Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Following are examples of Start URL check relaxations:

- Allow users to access the home page at www.example.com:

  ^http://www[.]example[.]com$
- Allow users to access all static HTML (.htm and .html), server-parsed HTML (.htp and .shtml), PHP (.php), and Microsoft ASP (.asp) format web pages at www.example.com:

  ^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
  [0-9A-Za-z][0-9A-Za-z_.-]*[.](asp|htp|php|s?html?)$
- Allow users to access web pages with pathnames or file names that contain non-ASCII characters at www.example-español.com:

  ^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*
  ([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
  Note: In the above expression, each character class has been grouped with the string \\x[0-9A-Fa-f][0-9A-Fa-f], which matches all properly-constructed character encoding strings but does not allow stray backslash characters that are not associated with a UTF-8 character encoding string. The double backslash (\\) is an escaped backslash, which tells the application firewall to interpret it as a literal backslash. If you included only one backslash, the application firewall would instead interpret the following left square bracket ([) as a literal character instead of the opening of a character class, which would break the expression.
- Allow users to access all GIF (.gif), JPEG (.jpg and .jpeg), and PNG (.png) format graphics at www.example.com:

  ^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
  [0-9A-Za-z][0-9A-Za-z_.-]*[.](gif|jpe?g|png)$
- Allow users to access CGI (.cgi) and PERL (.pl) scripts, but only in the CGI-BIN directory:

  ^http://www[.]example[.]com/CGI-BIN/[0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$
- Allow users to access Microsoft Office and other document files in the docsarchive directory:

  ^http://www[.]example[.]com/docsarchive/[0-9A-Za-z][0-9A-Za-z_-.]*[.](doc|xls|pdf|ppt)$

> **Note**
>
> By default, all application firewall URLs are considered to be regular expressions.

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions that you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the Start URL check would otherwise have blocked.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Deny URL Check

Aug 16, 2017

The Deny URL check examines and blocks connections to URLs that are commonly accessed by hackers and malicious code. This check contains a list of URLs that are common targets of hackers or malicious code and that rarely if ever appear in legitimate requests. You can also add URLs or URL patterns to the list. The Deny URL check prevents attacks against various security weaknesses known to exist in web server software or on many web sites.

The Deny URL check takes priority over the Start URL check, and thus denies malicious connection attempts even when a Start URL relaxation would normally allow a request to proceed.

In the **Modify Deny URL Check** dialog box, on the **General** tab you can **enable** or **disable** the Block, Log, and Statistics actions.
Also, there are built-in Deny URL rules that are disabled, by default. These built-in rules can be enabled.

If you use the command-line interface, you can enter the following command to configure the Deny URL Check:

- set appfw profile <name> -denyURLAction [**block**] [**log**] [**stats**] [**none**]

To create and configure your own deny URLs, you must use the configuration utility. On the Checks tab of the Modify Deny URL Check dialog box, click Add to open the Add Deny URL dialog box, or select an existing user-defined deny URL and click Open to open the Modify Deny URLdialog box. Either dialog box provides the same options for creating and configuring a deny URL.

Following are examples of Deny URL expressions:

- Do not allow users to access the image server at images.example.com directly:

  ```
  ^http://images[.]example[.]com$
  ```
- Do not allow users to access CGI (.cgi) or PERL (.pl) scripts directly:

  ```
  ^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
  [0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$
  ```
- Here is the same deny URL, modified to support non-ASCII characters:

  ```
  ^http://www[.]example[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
  ([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
  ([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](cgi|pl)$
  ```

Important

Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL or pattern that you want to block, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Built-in Deny URL rules are available that are disabled. These built-in rules can be enabled based on your requirement.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XML Protection Checks

Jul 25, 2015

The XML Protection checks examine requests for XML-based attacks of all types.

Caution: The XML security checks apply only to content that is sent with an HTTP content-type header of text/xml. If the content-type header is missing, or is set to a different value, all XML security checks are bypassed. If you plan to protect XML or Web 2.0 web applications, the webmasters of each web server that hosts those applications should ensure that the proper HTTP content-type header is sent.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XML Format Check

Jul 25, 2015

The XML Format check examines the XML format of incoming requests and blocks those requests that are not well formed or that do not meet the criteria in the XML specification for properly-formed XML documents. Some of those criteria are:

- An XML document must contain only properly-encoded Unicode characters that match the Unicode specification.
- No special XML syntax characters—such as < , > and &—can be included in the document except when used in XML markup.
- All begin, end, and empty-element tags must be correctly nested, with none missing or overlapping.
- XML element tags are case-sensitive. All beginning and end tags must match exactly.
- A single root element must contain all the other elements in the XML document.

A document that does not meet the criteria for well-formed XML does not meet the definition of an XML document. Strictly speaking, it is not XML. However, not all XML applications and web services enforce the XML well-formed standard, and not all handle poorly-formed or invalid XML correctly. Inappropriate handling of a poorly-formed XML document can cause security breaches. The purpose of the XML Format check is to prevent a malicious user from using a poorly-formed XML request to breach security on your XML application or web service.

If you use the wizard or the configuration utility, in the Modify XML Format Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the XML Format Check:

- set appfw profile <name> -xmlFormatAction [**block**] [**log**] [**stats**] [**none**]

You cannot configure exceptions to the XML Format check. You can only enable or disable it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XML Denial-of-Service Check

Jul 25, 2015

The XML Denial of Service (XML DoS or XDoS) check examines incoming XML requests to determine whether they match the characteristics of a denial-of-service (DoS) attack, and blocks those requests that do. The purpose of the XML DoS check is to prevent an attacker from using XML requests to launch a denial-of-service attack on your web server or web site.

If you use the wizard or the configuration utility, in the Modify XML Denial-of-Service Check dialog box, on the General tab you can enable or disable the Block, Log, Statistics, and Learn actions:

If you use the command-line interface, you can enter the following command to configure the XML Denial-of-Service check:

- set appfw profile <name> -xmlDoSAction [**block**] [**log**] [**learn**] [**stats**] [**none**]

To configure individual XML Denial-of-Service rules, you must use the configuration utility. On the Checks tab of the Modify XML Denial-of-Service Check dialog box, select a rule and click Open to open the Modify XML Denial-of-Service dialog box for that rule. The individual dialog boxes differ for the different rules but are extremely simple. Some only allow you to enable or disable the rule; others allow you to modify a number by typing a new value in a text box.

The individual XML Denial-of-Service rules are:

Maximum Element Depth

Restrict the maximum number of nested levels in each individual element to 256. If this rule is enabled, and the application firewall detects an XML request with an element that has more than the maximum number of allowed levels, it blocks the request. You can modify the maximum number of levels to any value from one (1) to 65,535.

Maximum Element Name Length

Restrict the maximum length of each element name to 128 characters. This includes the name within the expanded namespace, which includes the XML path and element name in the following format:

{http://prefix.example.com/path/}target_page.xml

The user can modify the maximum name length to any value between one (1) character and 65,535.

Maximum # Elements

Restrict the maximum number of any one type of element per XML document to 65,535. You can modify the maximum number of elements to any value between one (1) and 65,535.

Maximum # Element Children

Restrict the maximum number of children (including other elements, character information, and comments) each individual element is allowed to have to 65,535. You can modify the maximum number of element children to any value between one (1) and 65,535.

Maximum # Attributes

Restrict the maximum number of attributes each individual element is allowed to have to 256. You can modify the maximum number of attributes to any value between one (1) and 256.

Maximum Attribute Name Length

Restrict the maximum length of each attribute name to 128 characters. You can modify the maximum attribute name

© 1999-2017 Citrix Systems, Inc. All rights reserved.

length to any value between one (1) and 2,048.

Maximum Attribute Value Length

Restrict the maximum length of each attribute value to 2048 characters. You can modify the maximum attribute name length to any value between one (1) and 2,048.

Maximum Character Data Length

Restrict the maximum character data length for each element to 65,535. You can modify the length to any value between one (1) and 65,535.

Maximum File Size

Restrict the size of each file to 20 MB. You can modify the maximum file size to any value.

Minimum File Size

Require that each file be at least 9 bytes in length. You can modify the minimum file size to any positive integer representing a number of bytes.

Maximum # Entity Expansions

Limit the number of entity expansions allowed to the specified number. Default: 1024.

Maximum Entity Expansion Depth

Restrict the maximum number of nested entity expansions to no more than the specified number. Default: 32.

Maximum # Namespaces

Limit the number of namespace declarations in an XML document to no more than the specified number. Default: 16.

Maximum Namespace URI Length

Limit the URL length of each namespace declaration to no more than the specified number of characters. Default: 256.

Block Processing Instructions

Block any special processing instructions included in the request. This rule has no user-modifiable values.

Block DTD

Block any document type definitions (DTD) included with the request. This rule has no user-modifiable values.

Block External Entities

Block all references to external entities in the request. This rule has no user-modifiable values.

SOAP Array Check

Enable or disable the following SOAP array checks:

- **Maximum SOAP Array Size.** The maximum total size of all SOAP arrays in an XML request before the connection is blocked. You can modify this value. Default: 20000000.
- **Maximum SOAP Array Rank.** The maximum rank or dimensions of any single SOAP array in an XML request before the connection is blocked. You can modify this value. Default: 16.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XML Cross-Site Scripting Check

Oct 13, 2015

The XML Cross-Site Scripting check examines the user requests for possible cross-site scripting attacks in the XML payload. If it finds a possible cross-site scripting attack, it blocks the request.

To prevent misuse of the scripts on your protected web services to breach security on your web services, the XML Cross-Site Scripting check blocks scripts that violate the same origin rule, which states that scripts should not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called cross-site scripting. The reason cross-site scripting is a security issue is that a web server that allows cross-site scripting can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

The application firewall offers various action options for implementing XML Cross-Site Scripting protection. You have the option to configure **Block**, **Log**, and **Stats** actions.

The application firewall XML XSS check is performed on the payload of the incoming requests and attack strings are identified even if they are spread over multiple lines. The check looks for XSS attack strings in the **element** and the **attribute** values. You can apply relaxations to bypass security check inspection under specified conditions. The logs and statistics can help you identify needed relaxations.

The CDATA section of the XML payload might be an attractive area of focus for the hackers because the scripts are not executable outside the CDATA section. A CDATA section is used for content that is to be treated entirely as character data. HTML mark up tag delimiters "**<**", "**>**", and "**/>**" will not cause the parser to interpret the code as HTML elements. The following example shows a CDATA Section with XSS attack string:

command COPY

```
<![CDATA[\r\n

<script language="Javascript" type="text/javascript">alert ("Got you")</script>\r\n

]]>
```

Action Options:

An action is applied when the XML Cross-Site Scripting check detects an XSS attack in the request. The following options are available for optimizing XML Cross-Site Scripting protection for your application:

- **Block**— Block action is triggered if the XSS tags are detected in the request.
- **Log**— Generate log messages indicating the actions taken by the XML Cross-Site Scripting check. If block is disabled, a separate log message is generated for each location (ELEMENT, ATTRIBUTE) in which the XSS violation is detected. However, only one message is generated when the request is blocked. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attempts to launch an attack.

- **Stats**—Gather statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.

Relaxation Rules

If your application requires you to bypass the Cross-Site Scripting check for a specific ELEMENT or ATTRIBUTE in the XML payload, you can configure a relaxation rule. The XML Cross-Site Scripting check relaxation rules have the following parameters:

- **Name**—You can use literal strings or regular expressions to configure the name of the ELEMENT or the Attribute. The following expression exempts all ELEMENTS beginning with the string name_ followed by a string of uppercase or lowercase letters, or numbers, that is at least two and no more than fifteen characters long:

  ^name_[0-9A-Za-z]{2,15}$

> ## Note
>
> The names are case sensitive. Duplicate entries are not allowed, but you can use capitalization of the names and differences in location to create similar entries. For example, each of the following relaxation rules is unique:
>
> 1) XMLXSS: ABC IsRegex: NOTREGEX
>
> Location: ATTRIBUTE State: ENABLED
>
> 2) XMLXSS: ABC IsRegex: NOTREGEX
>
> Location: ELEMENT State: ENABLED
>
> 3) XMLXSS: abc IsRegex: NOTREGEX
>
> Location: ELEMENT State: ENABLED
>
> 4) XMLXSS: abc IsRegex: NOTREGEX
>
> Location: ATTRIBUTE State: ENABLED

- **Location**—You can specify the Location of the Cross-site Scripting Check exception in your XML payload. The option ELEMENT is selected by default. You can change it to ATTRIBUTE.
- **Comment**—This is an optional field. You can use up to a 255 character string to describe the purpose of this relaxation Rule.

> ## Warning
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the name that you want to add as an exception, and nothing else. Careless use of Regular Expressions can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the XML Cross-Site Scripting check would otherwise have blocked.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Using the Command Line to Configure the XML Cross-Site Scripting check

To configure XML Cross-Site Scripting check actions and other parameters by using the command line

If you use the command-line interface, you can enter the following commands to configure the XML Cross-Site Scripting Check:

> **set appfw profile** <name> -**XMLXSSAction** (([**block**] [**log**] [**stats**]) | [**none**])

To configure a XML Cross-Site Scripting check relaxation rule by using the command line

You can add relaxation rules to bypass inspection of XSS script attack inspection in a specific location. Use the bind or unbind command to add or delete the relaxation rule binding, as follows:

> **bind appfw profile** <name> -**XMLXSS** <string> [**isRegex** (**REGEX** | **NOTREGEX**)] [-**location** (**ELEMENT** | **ATTRIBUTE**)] –**comment** <string> [-**state** (**ENABLED** | **DISABLED**)]

> **unbind appfw profile** <name> -**XMLXSS** <String>

Example

> **bind appfw profile test_pr -XMLXSS ABC**

After executing the above command, the following relaxation rule is configured. The rule is enabled, the name is treated as a literal (NOTREGEX), and ELEMENT is selected as the default location:

1) XMLXSS: ABC IsRegex: NOTREGEX

Location: ELEMENT State: ENABLED

> **unbind appfw profile test_pr -XMLXSS abc**

ERROR: No such XMLXSS check

> **unbind appfw profile test_pr -XMLXSS ABC**

Done

Using the Configuration Utility to Configure the XML Cross-Site Scripting Check

In the configuration utility, you can configure the XML Cross-Site Scripting check in the pane for the profile associated with your application.

To configure or modify the XML Cross-Site Scripting check by using the configuration utility

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

   a) If you just want to enable or disable **Block**, **Log**, and **Stats** actions for the **XML Cross-Site Scripting check**, you can select or clear check boxes in the table, click **OK**, and then click Save and Close to close the Security Check pane.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

b) You can double click **XML Cross-Site Scripting**, or select the row and click **Action Settings**, to display the action options. After changing any of the action settings, click **OK** to save the changes and return to the Security Checks table.

You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

To configure a XML Cross-Site Scripting relaxation rule by using the configuration utility

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**.
3. In the Relaxation Rules table, double-click the **XML Cross-Site Scripting** entry, or select it and click **Edit**.
4. In the **XML Cross-Site Scripting Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

To manage XML Cross-Site Scripting relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **XML Cross-Site Scripting** row in the Relaxation Rules table, and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the Cross-Site Scripting patterns by using the configuration utility

You can use the configuration utility to view or customize the default list of XSS allowed attributes or allowed tags. You can also view or customize the default list of XSS denied Patterns.

The default lists are specified in **Application Firewall** > **Signatures** > **Default Signatures**. If you do not bind any signature object to your profile, the default XSS Allowed and Denied list specified in the Default Signatures object will be used by the profile for the Cross-Site Scripting security check processing. The Tags, Attributes, and Patterns, specified in the default signatures object, are read-only. You cannot edit or modify them. If you want to modify or change these, make a copy of the Default Signatures object to create a User-Defined signature object. Make changes in the Allowed or Denied lists in the new user-defined signature object and use this signature object in the profile that is processing the traffic for which you want to use these customized allowed and denied lists.

For more information about signatures, see http://support.citrix.com/proddocs/topic/ns-security-10-map/appfw-signatures-con.html.

To view default XSS patterns:

1. Navigate to **Application firewall** > **Signatures**, select ***Default Signatures**, and click **Edit**. Then click **Manage SQL/XSS Patterns**.

   The **Manage SQL/XSS Paths** table shows following three rows pertaining to XSS :

   xss/allowed/attribute

   xss/allowed/tag

   xss/denied/pattern

   Select a row and click **Manage Elements** to display the corresponding XSS Elements (Tag, Attribute, Pattern) used by

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

the application firewall **Cross-Site Scripting** check.

To customize XSS Elements: You can edit the user-defined signature object to customize the allowed Tag, allowed Attributes and denied Patterns. You can add new entries or remove the existing ones.

1. **Navigate to Application firewall** > **Signatures**, highlight the target user-defined signature, and click **Edit**. Click **Manage SQL/XSS Patterns** to display the **Manage SQL/XSS paths** table.
2. Select the target XSS row.

   a) Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding XSS element.

   b) Click **Remove** to remove the selected row.

> ## Warning
>
> Be very careful when you remove or modify any default XSS element, or delete the XSS path to remove the entire row. The signatures, HTML Cross-Site Scripting security check, and XML Cross-Site Scripting security check rely on these Elements for detecting attacks to protect your applications. Customizing the XSS Elements can make your application vulnerable to Cross-Site Scripting attacks if the required pattern is removed during editing.

Using the Log Feature with the XML Cross-Site Scripting Check

When the log action is enabled, the XML Cross-Site Scripting security check violations are logged in the audit log as **APPFW_XML_XSS** violations. The application firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the XML Cross-Site Scripting violations:

> **Shell**

> **tail -f /var/log/ns.log | grep APPFW_XML_XSS**

Example of a XML Cross-Site Scripting security check violation log message in Native log format showing <blocked> action

Oct 7 01:44:34 <local0.warn> 10.217.31.98 10/07/2015:01:44:34 GMT ns 0-PPE-1 : default APPFW APPFW_XML_XSS 1154 0 : 10.217.253.69 3466-PPE1 - owa_profile http://10.217.31.101/FFC/login.html Cross-site script check failed for field script="Bad tag: script" <**blocked**>

Example of a XML Cross-Site Scripting security check violation log message in CEF log format showing <not blocked> action

Oct 7 01:46:52 <local0.warn> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_XML_XSS|4|src=10.217.30.17 geolocation=Unknown spt=33141 method=GET request=http://10.217.31.101/FFC/login.html msg=Cross-site script check failed for field script\="Bad tag: script" cn1=1607 cn2=3538 cs1=owa_profile cs2=PPE0 cs4=ERROR cs5=2015 act=**not blocked**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To access the log messages by using the configuration utility

The Citrix configuration utility includes a useful tool (**Syslog Viewer**) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **XML Cross-Site Scripting** row and click **Logs**. When you access the logs directly from the XML Cross-Site Scripting check of the profile, the configuration utility filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the Syslog messages link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the **Audit Messages** section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The XML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **XML Cross-Site Scripting** check, filter by selecting **APPFW** in the dropdown options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_XML_XSS** check box and click the **Apply** button, only log messages pertaining to the XML Cross-Site Scripting security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, **Client IP** etc. appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Statistics for the XML Cross-Site Scripting violations

When the stats action is enabled, the counter for the XML Cross-Site Scripting check is incremented when the application firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, a request for a page that contains three XML Cross-Site Scripting violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display XML Cross-Site Scripting check statistics by using the command line

At the command prompt, type:

> **sh appfw stats**

To display stats for a specific profile, use the following command:

> **stat appfw profile** <profile name>

To display XML Cross-Site Scripting statistics by using the configuration utility

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the **Statistics** Link.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

3. Use the scroll bar to view the statistics about XML Cross-Site Scripting violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XML SQL Injection Check

Jan 27, 2016

The XML SQL Injection check examines the user requests for possible XML SQL Injection attacks. If it finds injected SQL in XML payloads, it blocks the requests.

A XML SQL attack can inject source code into a web application such that it can be interpreted and executed as a valid SQL query to perform a database operation with malicious intent. For example, XML SQL attacks can be launched to gain unauthorized access to the contents of a database or to manipulate the stored data. XML SQL Injection attacks are not only common, but can also be very harmful and costly.

Compartmentalizing the privileges of the database users can assist in protecting the database to some extent. All database users should be given only the required privileges to complete their intended tasks, so that they cannot execute SQL queries to perform other tasks. For example, a read-only user should not be allowed to write or manipulate data tables. The application firewall XML SQL Injection check inspects all XML requests to provide special defenses against injection of unauthorized SQL code that might break security. If the application firewall detects unauthorized SQL code in any XML request of any user, it can block the request.

The NetScaler application firewall inspects the presence of SQL keywords and special characters to identify the XML SQL Injection attack. A default set of keywords and special characters provides known keywords and special characters that are commonly used to launch XML SQL attacks. The application firewall considers three characters, single straight quote ('), backslash (\), and semicolon (;) as special characters for SQL security check processing. You can add new patterns, and you can edit the default set to customize the XML SQL check inspection.

The application firewall offers various action options for implementing XML SQL Injection protection. You can **Block** the request, **Log** a message in the ns.log file with details regarding the observed violations, and collect **Stats** to keep track of the number of observed attacks.

In addition to actions, there are several parameters that can be configured for XML SQL injection processing. You can check for **SQL wildcard characters**. You can change the XML SQL Injection type and select one of the 4 options (**SQLKeyword**, **SQLSplChar**, **SQLSplCharANDKeyword**, **SQLSplCharORKeyword**) to indicate how to evaluate the SQL keywords and SQL special characters when processing the XML payload. The XML **SQL Comments Handling** parameter gives you an option to specify the type of comments that need to be inspected or exempted during XML SQL Injection detection.

You can deploy relaxations to avoid false positives. The application firewall XML SQL check is performed on the payload of the incoming requests, and attack strings are identified even if they are spread over multiple lines. The check looks for SQL Injection strings in the **element** and the **attribute** values. You can apply relaxations to bypass security check inspection under specified conditions. The logs and statistics can help you identify needed relaxations.

Action Options

An action is applied when the XML SQL Injection check detects an SQL Injection attack string in the request. The following actions are available for configuring an optimized XML SQL Injection protection for your application:

Block—If you enable block, the block action is triggered only if the input matches the XML SQL injection type specification. For example, if **SQLSplCharANDKeyword** is configured as the XML SQL injection type, a request is not blocked if it does not contain any key words, even if SQL special characters are detected in the payload. Such a request is blocked if the XML SQL injection type is set to either **SQLSplChar**, or **SQLSplCharORKeyword**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Log—If you enable the log feature, the XML SQL Injection check generates log messages indicating the actions that it takes. If block is disabled, a separate log message is generated for each location (**ELEMENT**, **ATTRIBUTE**) in which the XML SQL violation was detected. However, only one message is generated when the request is blocked. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.

Stats—If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.

XML SQL Parameters

In addition to the block, log and stats actions, you can configure the following parameters for XML SQL Injection check:

Check for XML SQL Wildcard Characters—Wild card characters can be used to broaden the selections of a structured query language (SQL-SELECT) statement. These wild card operators can be used in conjunction with **LIKE** and **NOT LIKE** operators to compare a value to similar values. The percent (%), and underscore (_) characters are frequently used as wild cards. The percent sign is analogous to the asterisk (*) wildcard character used with MS-DOS and to match zero, one, or multiple characters in a field. The underscore is similar to the MS-DOS question mark (?) wildcard character. It matches a single number or character in an expression.

For example, you can use the following query to do a string search to find all customers whose names contain the D character.

SELECT * from customer WHERE name like "%D%"

The following example combines the operators to find any salary values that have 0 as the second and third character.

SELECT * from customer WHERE salary like '_00%'

Different DBMS vendors have extended the wildcard characters by adding extra operators. The NetScaler application firewall can protect against attacks that are launched by injecting these wildcard characters. The 5 default Wildcard characters are percent (%), underscore (_), caret (^), opening square bracket ([), and closing square bracket (]). This protection applies to both HTML and XML profiles.

The default wildcard chars are a list of literals specified in the ***Default Signatures**:

- <wildchar type="LITERAL">%</wildchar>
- <wildchar type="LITERAL">_</wildchar>
- <wildchar type="LITERAL">^</wildchar>
- <wildchar type="LITERAL">[</wildchar>
- <wildchar type="LITERAL">]</wildchar>

Wildcard characters in an attack can be PCRE, like [^A-F]. The application firewall also supports PCRE wildcards, but the literal wildcard chars above are sufficient to block most attacks.

> **Note**
>
> The XML SQL **wildcard character** check is different from the XML SQL **special character** check. This option must be used with caution to avoid false positives.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Check Request Containing SQL Injection Type—The application firewall provides 4 options to implement the desired level of strictness for SQL Injection inspection, based on the individual need of the application. The request is checked against the injection type specification for detecting SQL violations. The 4 SQL injection type options are:

- **SQL Special Character and Keyword**—Both an SQL keyword and an SQL special character must be present in the inspected location to trigger SQL violation. This least restrictive setting is also the default setting.
- **SQL Special Character**—At least one of the special characters must be present in the processed payload string to trigger SQL violation.
- **SQL keyword**—At least one of the specified SQL keywords must be present in the processed payload string to trigger an SQL violation. Do not select this option without due consideration. To avoid false positives, make sure that none of the keywords are expected in the inputs.
- **SQL Special Character or Keyword**—Either the keyword or the special character string must be present in the payload to trigger the security check violation.

> **Tip**
>
> If you select the SQL Special Character option, the application firewall skips strings that do not contain any special characters. Since most SQL servers do not process SQL commands that are not preceded by a special character, enabling this option can significantly reduce the load on the application firewall and speed up processing without placing your protected web sites at risk.

SQL comments handling—By default, the application firewall parses and checks all comments in XML data for injected SQL commands. Many SQL servers ignore anything in a comment, even if preceded by an SQL special character. For faster processing, if your XML SQL server ignores comments, you can configure the application firewall to skip comments when examining requests for injected SQL. The XML SQL comments handling options are:

- **ANSI**—Skip ANSI-format SQL comments, which are normally used by UNIX-based SQL databases.
- **Nested**—Skip nested SQL comments, which are normally used by Microsoft SQL Server.
- **ANSI/Nested**—Skip comments that adhere to both the ANSI and nested SQL comment standards. Comments that match only the ANSI standard, or only the nested standard, are still checked for injected SQL.
- **Check all Comments**—Check the entire request for injected SQL, without skipping anything. This is the default setting.

> **Tip**
>
> In most cases, you should not choose the Nested or the ANSI/Nested option unless your back-end database runs on Microsoft SQL Server. Most other types of SQL server software do not recognize nested comments. If nested comments appear in a request directed to another type of SQL server, they might indicate an attempt to breach security on that server.

Relaxation rules

If your application requires you to bypass the XML SQL Injection inspection for a specific ELEMENT or ATTRIBUTE in the XML payload, you can configure a relaxation rule. The XML SQL Injection inspection relaxation rules have the following parameters:

- **Name:** You can use literal strings or regular expressions to configure the name of the **ELEMENT** or the **ATTRIBUTE**. The following expression exempts all **ELEMENTS** beginning with the string **PurchaseOrder_** followed by a string of numbers

© 1999-2017 Citrix Systems, Inc. All rights reserved.

that is at least two and no more than ten characters long:

Comment: "Exempt XML SQL Check for Purchase Order Elements"

XMLSQLInjection: "PurchaseOrder_[0-9A-Za-z]{2,10}"

IsRegex: REGEX Location: ELEMENT

State: ENABLED

Note: The names are case sensitive. Duplicate entries are not allowed, but you can use capitalization of the names and differences in location to create similar entries. For example, each of the following relaxation rules is unique:

1) XMLSQLInjection: XYZ IsRegex: NOTREGEX

   Location: ELEMENT State: ENABLED

2) XMLSQLInjection: xyz IsRegex: NOTREGEX

   Location: ELEMENT State: ENABLED

3) XMLSQLInjection: xyz IsRegex: NOTREGEX

   Location: ATTRIBUTE State: ENABLED

4) XMLSQLInjection: XYZ IsRegex: NOTREGEX

   Location: ATTRIBUTE State: ENABLED

- **Location**: You can specify the Location of the XML SQL Inspection exception in your XML payload. The option **ELEMENT** is selected by default. You can change it to **ATTRIBUTE**.
- **Comment**: This is an optional field. You can use up to a 255 character string to describe the purpose of this relaxation Rule.

> ## Warning
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the name that you want to add as an exception, and nothing else. Careless use of Regular Expressions can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the XML SQL Injection inspection would otherwise have blocked.

Using the Command Line to Configure the XML SQL Injection Check

To configure XML SQL Injection actions and other parameters by using the command line

In the command line interface, you can use either the **set appfw profile** command or the **add appfw profile** command to configure the XML SQL Injection protections. You can enable the block, log, and stats action(s). Select the type of SQL attack pattern (key words, wildcard characters, special strings) you want to detect in the payloads. Use the **unset appfw profile** command to revert the configured settings back to their defaults. Each of the following commands sets only one parameter, but you can include multiple parameters in a single command:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **set appfw profile** <name> **-XMLSQLInjectionAction** (([**block**] [**log**] [**stats**]) | [**none**])
- **set appfw profile** <name> **-XMLSQLInjectionCheckSQLWildChars** (**ON** | **OFF**)
- **set appfw profile** <name> **-XMLSQLInjectionType** ([**SQLKeyword**] | [**SQLSplChar**] | [**SQLSplCharANDKeyword**] | [**SQLSplCharORKeyword**])
- **set appfw profile** <name> **-XMLSQLInjectionParseComments** ([**checkall**] | [**ansi**|**nested**] | [**ansinested**])

To configure a SQL Injection relaxation rule by using the command line

Use the bind or unbind command to add or delete relaxation rules, as follows:

- **bind appfw profile** <name> **-XMLSQLInjection** <string> [**isRegex** (**REGEX** | **NOTREGEX**)] [-**location** (**ELEMENT** | **ATTRIBUTE**)] –**comment** <string> [-**state** (**ENABLED** | **DISABLED**)]
- **unbind appfw profile** <name> -**XMLSQLInjection** <String>

Example:

> bind appfw profile test_profile -XMLSQLInjection "PurchaseOrder_[0-9A-Za-z]{2,15}" -isregex REGEX -location ATTRIBUTE

> unbind appfw profile test_profile –XMLSQLInjection "PurchaseOrder_[0-9A-Za-z]{2,15}" -location ATTRIBUTE

Using the Configuration Utility to Configure the XMLSQL Injection Security Check

In the configuration utility, you can configure the XML SQL Injection security check in the pane for the profile associated with your application.

To configure or modify the XML SQL Injection check by using the configuration utility

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

   a. If you just want to enable or disable **Block**, **Log**, and **Stats** actions for **XML SQL Injection**, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the Security Check pane.

   b. If you want to configure additional options for this security check, double click **XML SQL Injection**, or select the row and click **Action Settings**, to display the following options:

   Check for SQL Wildcard Characters—Consider SQL Wildcard characters in the payload to be attack patterns.

   Check Request Containing—Type of SQL injection (**SQLKeyword**, **SQLSplChar**, **SQLSplCharANDKeyword**, or **SQLSplCharORKeyword**) to check.

   SQL Comments Handling—Type of comments (**Check All Comments**, **ANSI**, **Nested**, or **ANSI/Nested**) to check.

After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

To configure a XML SQL Injection relaxation rule by using the configuration utility

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**.
3. In the Relaxation Rules table, double-click the **XML SQL Injection** entry, or select it and click **Edit**.
4. In the **XML SQL Injection Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

To manage XML SQL Injection relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **XML SQL Injection** row in the Relaxation Rules table, and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the SQL Injection patterns by using the configuration utility

You can use the configuration utility to view or customize the SQL patterns.

The default SQL patterns are specified in **Application Firewall** > **Signatures** > ***Default Signatures**. If you do not bind any signature object to your profile, the default SQL patterns specified in the Default Signatures object will be used by the profile for XML SQL Injection security check processing. The rules and patterns in the Default Signatures object are read-only. You cannot edit or modify them. If you want to modify or change these patterns, create a user-defined signature object by making a copy of the Default Signatures object and changing the SQL patterns. Use the user-defined signature object in the profile that processes the traffic for which you want to use these customized SQL patterns.

For more information, see Signatures.

To view default SQL patterns:

a. Navigate to **Application firewall** > **Signatures**, select ***Default Signatures**, and click **Edit**. Then click **Manage SQL/XSS Patterns**.

The Manage SQL/XSS Paths table shows following four rows pertaining to SQL Injection:

Injection (not_alphanum, SQL)/ Keyword

Injection (not_alphanum, SQL)/ specialstring

Injection (not_alphanum, SQL)/ transformrules/transform

Injection (not_alphanum, SQL)/ wildchar

b. Select a row and click **Manage Elements** to display the corresponding SQL patterns (keywords, special strings, transformation rules or the wildcard characters) used by the application firewall SQL injection check.

To customize SQL Patterns: You can edit a user-defined signature object to customize the SQL key words, special strings, and wildcard characters. You can add new entries or remove the existing ones. You can modify the transformation rules for the SQL special strings.

a. Navigate to **Application firewall** > **Signatures**, highlight the target user-defined signature, and click **Edit**. Click **Manage SQL/XSS Patterns** to display the **Manage SQL/XSS paths** table.

b. Select the target SQL row.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

i. Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding SQL element.

ii. Click **Remove** to remove the selected row.

> **Warning**
>
> You must be very careful when removing or modifying any default SQL element, or deleting the SQL path to remove the entire row. The signature rules as well as the XML SQL Injection security check rely on these elements for detecting SQL Injection attacks to protect your applications. Customizing the SQL patterns can make your application vulnerable to XML SQL attacks if the required pattern is removed during editing.

Using the Log Feature with the XML SQL Injection Check

When the log action is enabled, the **XML SQL Injection** security check violations are logged in the audit log as **APPFW_XML_SQL** violations. The application firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the XML Cross-Site Scripting violations:

> **Shell**

> **tail -f /var/log/ns.log | grep APPFW_XML_SQL**

To access the log messages by using the configuration utility

The Citrix configuration utility includes a useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **XML SQL Injection** row and click **Logs**. When you access the logs directly from the XML SQL Injection check of the profile, the configuration utility filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **Policies** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the **Syslog Viewer**, which displays all log messages, including other security check violation logs.

The XML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **XML SQL Injection** check, filter by selecting **APPFW** in the dropdown options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_XML_SQL** check box and click the **Apply** button, only log messages pertaining to the **XML SQL Injection** security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, and

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Client IP appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Statistics for the XML SQL Injection violations

When the stats action is enabled, the counter for the **XML SQL Injection** check is incremented when the application firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, a request for a page that contains three **XML SQL Injection** violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display XML SQL Injection check statistics by using the command line

At the command prompt, type:

> **sh appfw stats**

To display stats for a specific profile, use the following command:

> **stat appfw profile** <profile name>

To display XML SQL Injection statistics by using the configuration utilitY

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about **XML SQL Injection** violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XML Attachment Check

Jul 25, 2015

The XML Attachment check examines incoming requests for malicious attachments, and it blocks those requests that contain attachments that might breach applications security. The purpose of the XML Attachment check is to prevent an attacker from using an XML attachment to breach security on your server.

If you use the wizard or the configuration utility, in the Modify XML Attachment Check dialog box, on the General tab you can enable or disable the Block, Learn, Log, Statistics, and Learn actions:

If you use the command-line interface, you can enter the following command to configure the XML Attachment Check:

- set appfw profile <name> -xmlAttachmentAction [**block**] [**learn**] [**log**] [**stats**] [**none**]

You must configure the other XML Attachment check settings in the configuration utility. In the Modify XML Attachment Check dialog box, on the Checks tab, you can configure the following settings:

- **Maximum Attachment Size.** Allow attachments that are no larger than the maximum attachment size you specify. To enable this option, first select the Enabled check box, and then type the maximum attachment size in bytes in the Size text box.
- **Attachment Content Type.** Allow attachments of the specified content type. To enable this option, first select the Enabled check box, and then enter a regular expression that matches the Content-Type attribute of the attachments that you want to allow.
  - You can type the URL expression directly in the text window. If you do so, you can use the Regex Tokens menu to enter a number of useful regular expressions at the cursor instead of typing them manually.
  - You can click Regex Editor to open the Add Regular Expression dialog box and use it to construct the URL expression.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Web Services Interoperability Check

Jul 25, 2015

The Web Services Interoperability (WS-I) check examines both requests and responses for adherence to the WS-I standard, and blocks those requests and responses that do not adhere to this standard. The purpose of the WS-I check is to block requests that might not interact with other XML appropriately. An attacker can use inconsistencies in interoperability to launch an attack on your XML application.

If you use the wizard or the configuration utility, in the Modify Web Services Interoperability Check dialog box, on the General tab you can enable or disable the Block, Log, Statistics, and Learn actions.

If you use the command-line interface, you can enter the following command to configure the Web Services Interoperability check:

- set appfw profile <name> -xmlWSIAction [**block**] [**log**] [**learn**] [**stats**] [**none**]

To configure individual Web Services Interoperability rules, you must use the configuration utility. On the Checks tab of the Modify Web Services Interoperability Check dialog box, select a rule and click Enable or Disable to enable or disable the rule. You can also click Open to open the Web Services Interoperability Detail message box for that rule. The message box displays read-only information about the rule. You cannot modify or make other configuration changes to any of these rules.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XML Message Validation Check

Jul 25, 2015

The XML Message Validation check examines requests that contain XML messages to ensure that they are valid. If a request contains an invalid XML message, the application firewall blocks the request. The purpose of the XML Validation check is to prevent an attacker from using specially constructed invalid XML messages to breach the security of your application.

If you use the wizard or the configuration utility, in the Modify XML Message Validation Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the XML Message Validation Check:

- set appfw profile <name> -xmlValidationAction [**block**] [**log**] [**stats**] [**none**]

You must use the configuration utility to configure the other XML Validation check settings. In the Modify XML Message Validation Check dialog box, on the Checks tab, you can configure the following settings:

- **XML Message Validation.** Use one of the following options to validate the XML message:
  - **SOAP Envelope.** Validate only the SOAP envelope of XML messages.
  - **WSDL.** Validate XML messages by using an XML SOAP WSDL. If you choose WSDL validation, in the WSDL Object drop-down list you must choose a WSDL. If you want to validate against a WSDL that has not already been imported to the application firewall, you can click the Import button to open the Manage WSDL Imports dialog box and import your WSDL. See "WSDL" for more information.
    - If you want to validate the entire URL, leave the Absolute radio button in the End Point Check button array selected. If you want to validate only the portion of the URL after the host, select the Relative radio button.
    - If you want the application firewall to enforce the WSDL strictly, and not allow any additional XML headers not defined in the WSDL, you must clear the Allow additional headers not defined in the WSDL check box.
      Caution: If you uncheck the Allow Additional Headers not defined in the WSDL check box, and your WSDL does not define all XML headers that your protected XML application or Web 2.0 application expects or that a client sends, you may block legitimate access to your protected service.
  - **XML Schema.** Validate XML messages by using an XML schema. If you choose XML schema validation, in the XML Schema Object drop-down list you must choose an XML schema. If you want to validate against an XML schema that has not already been imported to the application firewall, you can click the Import button to open the Manage XML Schema Imports dialog box and import your WSDL. See "WSDL" for more information.
- **Response Validation.** By default, the application firewall does not attempt to validate responses. If you want to validate responses from your protected application or Web 2.0 site, select the Validate Response check box. When you do, the Reuse the XML Schema specified in request validation check box and the XML Schema Object drop-down list are activated.
  - Check the Reuse XML Schema check box to use the schema you specified for request validation to do response validation as well.
    Note: If you check this check box, the XML Schema Object drop-down list is grayed out.
  - If you want to use a different XML schema for response validation, use the XML Schema Object drop-down list to select or upload that XML schema .

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XML SOAP Fault Filtering Check

Jul 25, 2015

The XML SOAP Fault Filtering check examines responses from your protected web services and filters out XML SOAP faults. This prevents leaking of sensitive information to attackers.

If you use the wizard or the configuration utility, in the Modify XML SOAP Fault Filtering Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions, and the Remove action, which removes SOAP faults before forwarding the response to the user.

If you use the command-line interface, you can enter the following command to configure the XML SOAP Fault Filtering Check:

set appfw profile <name> -XMLSOAPFaultAction [**block**] [**log**] [**stats**] [**none**]

You cannot configure exceptions to the XML SOAP Fault Filtering check. You can only enable or disable it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing Content Types

Jun 23, 2017

Web servers usually add a Content-Type header that contains a MIME/type definition for the type of content in each file that the web server serves to users. Web servers serve many different types of content. For example, standard HTML is assigned the "text/html" MIME type. JPG images are assigned the "image/jpeg" or "image/jpg" content type. A normal web server can serve dozens or hundreds of different types of content, all defined in the Content Type header by an assigned MIME/type.

Many application firewall filtering rules are designed to filter specific types of content. Because filtering rules that apply to one type of content (such as HTML) are often inappropriate when filtering a different type of content (such as images), the application firewall attempts to determine the content type of requests and responses before it filters them. When a web server or browser does not add a Content-Type header to a request or response, the application firewall applies a default content type to the connection and filters the content accordingly.

The default content type is normally "application/octet-stream", the most generic MIME/type definition.This MIME/type is appropriate for any type of content that a web server is likely to serve, but also does not provide much information to the application firewall to allow it to choose appropriate filtering. If a protected web server on your network is configure to add accurate content type headers to the content it serves, or serves only one type of content, you can create a profile for that web server and assign a different default content type to it to improve both the speed and the accuracy of filtering.

You can also configure a list of allowed response content types for a specific profile. When this feature is configured, if the application firewall filters a response that does not match one of the allowed content types, it blocks the response. After upgrade from release 10.5 to 11.0, unknown content-types which are not in the default allowed content-type list do not bind. You can add other content-types which you want to be allowed to the relaxed rules.

Requests must always be of either the "application/x-www-form-urlencoded", "multipart/form-data", or "text/x-gwt-rpc" types. The application firewall blocks any request that has any other content type designated.

> ## Note
>
> You cannot include the "application/x-www-form-urlencoded" or "multipart/form-data" content types on the allowed response content types list.

To set the default request content type by using the command line interface

At the command prompt, type the following commands:

- set appfw profile <name> -requestContentType <type>
- save ns config

Example

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following example sets the "text/html" content type as the default for the specified profile:

```
set appfw profile profile1 -requestContentType "text/html"
save ns config
```

To remove the user-defined default request content type by using the command line interface

At the command prompt, type the following commands:

- unset appfw profile <name> -requestContentType <type>
- save ns config

Example

The following example unsets the default content type of "text/html" for the specified profile, allowing the type to revert to "application/octet-stream":

```
unset appfw profile profile1 -requestContentType "text/html"
save ns config
```

To set the default response content type by using the command line interface

At the command prompt, type the following commands:

- set appfw profile <name> -responseContentType <type>
- save ns config

Example

The following example sets the "text/html" content type as the default for the specified profile:

```
set appfw profile profile1 -responseContentType "text/html"
save ns config
```

To remove the user-defined default response content type by using the command line interface

At the command prompt, type the following commands:

- unset appfw profile <name> -responseContentType <type>
- save ns config

Example

The following example unsets the default content type of "text/html" for the specified profile, allowing the type to revert to "application/octet-stream":

```
unset appfw profile profile1 -responseContentType "text/html"
save ns config
```

To add a content type to the allowed content types list by using the command line interface

At the command prompt, type the following commands:

- bind appfw profile <name> -ContentType <contentTypeName>
- save ns config

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example

The following example adds the "text/shtml" content type to the allowed content types list for the specified profile:

```
bind appfw profile profile1 -contentType "text/shtml"
save ns config
```

To remove a content type from the allowed content types list by using the command line interface

At the command prompt, type the following commands:

- unbind appfw profile <name> -ContentType <contentTypeName>
- save ns config

Example

The following example removes the "text/shtml" content type from the allowed content types list for the specified profile:

```
unbind appfw profile profile1 -contentType "text/shtml"
save ns config
```

To manage the default and allowed content types by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, select the profile that you want to configure, and then click Edit. The Configure Application Firewall Profile dialog box is displayed.
3. In the Configure Application Firewall Profile dialog box, click the Settings tab
4. On the Settings tab, scroll down about halfway to the Content Type area.
5. In the Content Type area, configure the default request or response content type:
   - To configure the default request content type, type the MIME/type definition of the content type you want to use in the Default Request text box.
   - To configure the default response content type, type the MIME/type definition of the content type you want to use in the Default Response text box.
   - To create a new allowed content type, click Add. The Add Allowed Content Type dialog box is displayed.
   - To edit an existing allowed content type, select that content type, and then click Open. The Modify Allowed Content Type dialog box is displayed.
6. To manage the allowed content types, click Manage Allowed Content Types.
7. To add a new content type or modify an existing content type, click Add or Open, and in the Add Allowed Content Type or Modify Allowed Content Type dialog box, do the following steps.
   1. Select/clear the Enabled check box to include the content type in, or exclude it from, the list of allowed content types.
   2. In the Content Type text box, type a regular expression that describes the content type that you want to add, or change the existing content type regular expression.
      Content types are formatted exactly as MIME type descriptions are.

      Note: You can include any valid MIME type on the allowed contents type list. Since many types of document can contain active content and therefore could potentially contain malicious content, you should exercise caution when adding MIME types to this list.
   3. In the Comments text box, add an optional comment that describes the reason for adding this particular MIME type to the allowed contents type list.
   4. Click Create or OK to save your changes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

8. Click Close to close the Manage Allowed Content Types dialog box and return to the Settings tab.
9. Click OK to save your changes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Profiles

Feb 03, 2014

A profile is a collection of security settings that are used to protect specific types of web content or specific parts of your web site. In a profile, you determine how the application firewall applies each of its filters (or checks) to requests to your web sites, and responses from them. The application firewall supports two types of profile: four built-in (default) profiles that do not require further configuration, and user-defined profiles that do require further configuration.

Built-In Profiles

The four application firewall built-in profiles provide simple protection for applications and web sites that either do not require protection, or that should not be directly accessed by users at all. These profile types are:

- **APPFW_BYPASS.** Skips all application firewall filtering and sends the unmodified traffic to the protected application or web site, or to the client.
- **APPFW_RESET.** Resets the connection, requiring that the client re-establish his or her session by visiting a designated start page.
- **APPFW_DROP.** Drops all traffic to or from the protected application or web site, and sends no response of any kind to the client.
- **APPFW_BLOCK.** Blocks traffic to or from the protected application or web site.

You use the built-in profiles exactly as you do user-defined profiles, by configuring a policy that selects the traffic to which you want to apply the profile and then associating the profile with your policy. Since you do not have to configure a built-in policy, it provides a quick way to allow or block specified types of traffic or traffic that is sent to specific applications or web sites.

User-Defined Profiles

User-defined profiles are profiles that are build and configured by users. Unlike the default profiles, you must configure a user-defined profile before it will be of use filtering traffic to and from your protected applications.

There are three types of user-defined profile:

- **HTML.** Protects HTML-based web pages.
- **XML.** Protects XML-based web services and web sites.
- **Web 2.0.** Protects Web 2.0 content that combines HTML and XML content, such as ATOM feeds, blogs, and RSS feeds.

The application firewall has a number of security checks, all of which can be enabled or disabled, and configured in a number of ways in each profile. Each profile also has a number of settings that control how it handles different types of content. Finally, rather than manually configuring all of the security checks, you can enable and configure the learning feature. This feature observes normal traffic to your protected web sites for a period of time, and uses those observations to provide you with a tailored list of recommended exceptions (*relaxations*) to some security checks, and additional rules for other security checks.

During initial configuration, whether by using the Application Firewall Wizard or manually, you normally create one general purpose profile to protect all content on your web sites that is not covered by a more specific profile. After that, you can create as many specific profiles as you want to protect more specialized content.

The Profiles pane consists of a table that contains the following elements:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name. Displays all the application firewall profiles configured in the appliance.

Bound signature. Displays the signatures object that is bound to the profile in the previous column, if any.

Policies. Displays the application firewall policy that invokes the profile in the leftmost column of that row, if any.

Comments. Displays the comment associated with the profile in the leftmost column of that row, if any.

Profile Type. Displays the type of profile. Types are Built-In, HTML, XML, and Web 2.0.

Above the table is a row of buttons and a drop-down list that allow you to create, configure, delete, and view information about your profiles:

- **Add.** Add a new profile to the list.
- **Edit.** Edit the selected profile.
- **Delete.** Delete the selected profile from the list.
- **Statistics.** View the statistics for the selected profile.
- **Action.** Drop-down list that contains additional commands. Currently allows you to import a profile that was exported from another application firewall configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating Application Firewall Profiles

Jun 12, 2014

You can create an application firewall profile in one of two ways: by using the command line, and by using the configuration utility. Creating a profile by using the command line requires that you specify options on the command line. The process is similar to that of configuring an existing profile, and with a few exceptions the two commands take the same parameters.

Creating a profile by using the configuration utility requires that you specify only two options. You specify basic or advanced *defaults*, the default configuration for the various security checks and settings that are part of a profile, and choose the profile *type* to match the type of content that the profile is intended to protect. You can also, optionally, add a comment. After you create the profile, you must then configure it by selecting it in the data pane, and then clicking Edit.

If you plan to use the learning feature or to enable and configure a large number of advanced protections, you should choose advanced defaults. In particular, if you plan to configure either of the SQL injection checks, either of the cross-site scripting checks, any check that provides protection against Web form attacks, or the cookie consistency check, you should plan to use the learning feature. Unless you include the proper exceptions for your protected Web sites when configuring these checks, they can block legitimate traffic. Anticipating all of the necessary exceptions without creating any that are too broad is difficult. The learning feature makes this task much easier. Otherwise, basic defaults are quick and should provide the protection that your web applications need.

There are three profile types:

- **HTML.** Protects standard HTML-based web sites.
- **XML.** Protects XML-based web services and web sites.
- **Web 2.0 (HTML XML).** Protects sites that contain both HTML and XML elements, such as ATOM feeds, blogs, and RSS feeds.

There are also a few restrictions on the name that you can give to a profile. A profile name cannot be the same as the name assigned to any other profile or action in any feature on the NetScaler appliance. Certain action or profile names are assigned to built-in actions or profiles, and can never be used for user profiles. A complete list of disallowed names can be found in the Application Firewall Profile Supplemental Information. If you attempt to create a profile with a name that has already been used for an action or a profile, an error message is displayed and the profile is not created.

To create an application firewall profile by using the command line interface

At the command prompt, type the following commands:

- add appfw profile <name> [-defaults (**basic** | **advanced**)]
- set appfw profile <name> -type (**HTML** | **XML** | **HTML XML**)
- set appfw profile <name> -comment "<comment>"
- save ns config

Example

The following example adds a profile named pr-basic, with basic defaults, and assigns a profile type of HTML. This is the appropriate initial configuration for a profile to protect an HTML Web site.

```
add appfw profile pr-basic -defaults basic -comment "Simple profile for web sites."
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set appfw profile pr-basic -type HTML
save ns config
```

To create an application firewall profile by using the configuration utility

Creating an application firewall profile requires that you specify only a few configuration details.

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, click Add.
3. In the Create Application Firewall Profile dialog box, type a name for your profile.
   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.
4. Choose the profile type from the drop-down list.
5. Click Create, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Application Firewall Profiles

Jun 12, 2014

To configure a user-defined application firewall profile, first configure the security checks, which are called *deep protections* or *advanced protections* in the application firewall wizard. Certain checks require configuration if you are to use them at all. Others have default configurations that are safe but limited in scope; your web sites might need or benefit from a different configuration that takes advantage of additional features of certain security checks.

After you have configured the security checks, you can also configure a number of other settings that control the behavior, not of a single security check, but the application firewall feature. The default configuration is sufficient to protect most web sites, but you should review them to make sure that they are right for your protected web sites.

For more information about the application firewall security checks, see "Advanced Protections."

To configure an application firewall profile by using the command line

At the command prompt, type the following commands:

- set appfw profile <name> <arg1> [<arg2> ...]
  where:
  - <arg1> = a parameter and any associated options.
  - <arg2> = a second parameter and any associated options.
  - ... = additional parameters and options.
  For descriptions of the parameters to use when configuring specific security checks, see "Advanced Protections."
- save ns config

Example

The following example shows how to enable blocking for the HTML SQL Injection and HTML Cross-Site Scripting checks in a profile named pr-basic. This command enables blocking for those actions while making no other changes to the profile.

```
set appfw profile pr-basic -crossSiteScriptingAction block
        -SQLInjectionAction block
```

To configure an application firewall profile by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, select the profile that you want to configure, and then click Edit.
3. In the Configure Application Firewall Profile dialog box, on the Security Checks tab, configure the security checks.
   - To enable or disable an action for a check, in the list, select or clear the check box for that action.
   - To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check.
     You can also select a check and, at the bottom of the dialog box, click Open to display the Configure Relaxation dialog box or Configure Rule dialog box for that check. These dialog boxes also vary from check to check. Most of them include a Checks tab and a General tab. If the check supports relaxations or user-defined rules, the Checks tab includes an Add button, which opens yet another dialog box, in which you can specify a relaxation or rule for the check. (A relaxation is a rule for exempting specified traffic from the check.) If relaxations have already been configured, you can select one and click Open to modify it.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- To review learned exceptions or rules for a check, select the check, and then click Learned Violations. In the Manage Learned Rules dialog box, select each learned exception or rule in turn.
  - To edit the exception or rule, and then add it to the list, click Edit & Deploy.
  - To accept the exception or rule without modification, click Deploy.
  - To remove the exception or rule from the list, click Skip.
- To refresh the list of exceptions or rules to be reviewed, click Refresh.
- To open the Learning Visualizer and use it to review learned rules, click Visualizer.
- To review the log entries for connections that matched a check, select the check, and then click Logs. You can use this information to determine which checks are matching attacks, so that you can enable blocking for those checks. You can also use this information to determine which checks are matching legitimate traffic, so that you can configure an appropriate exemption to allow those legitimate connections. For more information about the logs, see "Logs, Statistics, and Reports."
- To completely disable a check, in the list, clear all of the check boxes to the right of that check.

4. On the Settings tab, configure the profile settings.
   - To associate the profile with the set of signatures that you previously created and configured, under Common Settings, choose that set of signatures in the Signatures drop-down list.
     Note: You may need to use the scroll bar on the right of the dialog box to scroll down to display the Common Settings section.
   - To configure an HTML or XML Error Object, select the object from the appropriate drop-down list.
     Note: You must first upload the error object that you want to use in the Imports pane. For more information about importing error objects, see "Imports."
   - To configure the default XML Content Type, type the content type string directly into the Default Request and Default Response text boxes, or click Manage Allowed Content Types to manage the list of allowed content types. ">>More..."
5. If you want to use the learning feature, click Learning, and configure the learning settings for the profile, as described in "Configuring and Using the Learning Feature".
6. Click OK to save your changes and return to the Profiles pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Changing an Application Firewall Profile Type

Jun 12, 2014

If you chose the wrong profile type for an application firewall profile, or the type of content on the protected web site has changed, you can change the profile type.

Note: When you change the profile type, you lose all configuration settings and learned relaxations or rules for the features that the new profile type does not support. For example, if you change the profile type from Web 2.0 to XML, you lose any configuration options for Start URL, Form Field Consistency Check, and the other HTML-specific security checks. The configuration for any options that is supported by both the old and the new profile types remains unchanged.

To change an application firewall profile type by using the command line interface

At the command prompt, type the following commands:

- set appfw profile <name> -type (**HTML** | **XML** | **HTML XML**)
- save ns config

Example

The following example changes the type of a profile named pr-basic, from HTML to HTML XML, which is equivalent to the Web 2.0 type in the configuration utility.

```
set appfw profile pr-basic -type HTML XML
save ns config
```

To change an application firewall profile type by using the configuration utility.

1. Navigate to Security > Application Firewall > Policies.
2. In the details pane, click Action, and then Change Profile Type.
3. In the Change Application Firewall Profile Type dialog box, Profile Type drop-down list, select a new profile type.
4. Click OK to save your changes and return to the Profiles pane.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exporting and Importing an Application Firewall Profile

Jan 12, 2016

You can replicate the entire configuration of an application firewall profile (including all bound objects, such as HTML error object, XML error object, WSDL or XML schema, signatures, and so on) across multiple appliances. You can select a target profile and export the configuration to save it in your computer's local file system, or you can transfer the archived configuration to store it on a server. Similarly, you can browse your computer's local file system or import the archive from the server to select a previously exported profile and import it into your NetScaler appliance.

The option to export the entire profile configuration and then import it into another appliance can be useful in various use cases. For example, you might want to configure an application firewall profile in a test bed set-up to test and validate that it is working as expected. Once you are satisfied, you can export the profile and import the profile configuration to your production NetScaler appliances. This functionality is also useful for backing up your configuration. You can export the profile before making changes, so that you can easily roll back the configuration to a known state if necessary.

> ## Note
>
> Application firewall profiles that are exported and archived from one build cannot be restored to a system running a different build, because changes introduced in the newer releases can lead to compatibility issues. If you attempt to restore an archived profile to a different build than the one from which it was exported, an error message is logged in ns.log.

The export and import profile functionality is available in both the configuration utility (GUI) and the command line interface (CLI). The configuration utility is recommended, because it offers easy to use **Action** options. With a click of a button you can **Export** or **Import** the entire configuration of a profile.

Exporting Application Firewall Profiles with the CLI

If you use CLI to **export** a profile, you must **archive** the configuration and then **export** it. To **import** a profile, you must **import** the archive into the NetScaler appliance and then execute the **restore** command to extract the configuration. The following set of CLI commands can be used for exporting, importing and managing the profile configurations.

CLI commands to export archives:

- archive appfw profile <name> <archivename> [-comment <string>]
- export appfw archive <name> <target>

CLI commands to import archives:

- import appfw archive <src> <name> [-comment <string>]
- restore appfw profile <archivename>

CLI commands to manage archives:

- show appfw archive
- rm appfw archive <name>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exporting a profile from one appliance and importing it to another requires five steps in CLI. The first 3 steps are performed on the source appliance on which profile configuration is originally created, and the next 2 steps are performed on the target appliance on which the profile configuration is to be replicated.

Export profile from the source NetScaler appliance:

Step 1: Create an archive of the configured profile.

Step 2: Export the archive to the NetScaler file system.

Step 3: Use a file transfer utility such as scp to transfer the exported archive file from NetScaler appliance A to the target NetScaler appliance.

Import profile to the target NetScaler appliance:

Step 4: Execute the import command to import the archived file. You can import the archive from your NetScaler's local file system, or you can use HTTP or HTTPS protocol to import the archive from a server by using the URL.

Step 5: Execute the restore command to restore the profile configuration from the imported archive

To export an application firewall profile by using the command line interface

First, **archive** the profile's configuration, and then **export** the archive to a target location. At the command prompt, type the following commands:

archive appfw profile <profileName> <archiveName>

where:

- <profileName> is the name of the profile to archive.
- <archiveName> is the name of the archive file to create.

Execution of the above command creates 2 instances of the archive file. One in /var/tmp folder and another in the /var/archive/appfw folder.

export appfw archive <archiveName> <target>

where:

- <archiveName> is the name of the archive to export. (The same name as in the previous command.
- <target> is a file path starting with local: as the prefix, followed by <archiveName>.

Execution of the export command saves the exported archive file on the file system of your NetScaler appliance in the /var/tmp folder.

Examples:

> archive appfw profile test_pr archived_test_pr

> export appfw archive archived_test_pr local:dutA_test_pr

After the above two commands are executed, the /var/tmp folder contains the archived_test_pr file and the exported copy, dutA_test_pr, which are identical in size. From the CLI, you can drop into the shell to navigate to the folder to verify that these files are there.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

After exporting the archive file, you can use **scp** or some other such file transfer utility to transfer a copy of the archive file from the NetScaler appliance on which they were created to your target NetScaler appliance.

Importing Application Firewall Profiles with the CLI

After you have successfully scp'd the archived file from the source appliance to the target appliance, you are ready to **import** the profile's archive, and then execute the **restore** command to replicate the profile's configuration on the target appliance.

Log onto the target appliance. Drop into the shell and cd to the /var/tmp folder to verify that the size of the scp'd file on this appliance matches the size of the original archived file on the source appliance. Exit the shell to return to the command line.

To import a profile by using the CLI

At the command prompt, type the following commands:

import appfw archive <src> <name> [-comment <string>]

where

- <src> is the location of the archive file after it has been transferred from the source appliance on which it was created. You can use a local file system and file name. If you have placed the archive on a server, you can use a URL to import the archived file. If the path or file name contains spaces, enclose the URL in straight double quotation marks.
- <name> is the name of the archive file to be imported.
- <string> is an optional description of the purpose of the Archive.

restore appfw profile <archiveName>

Examples:

A. Import from local file followed by restore

> import appfw archive local:dutA_test_pr dut2_test_pr

> restore appfw profile dut2_test_pr

B. Import from URL followed by restore

> import appfw archive http://10.217.30.16/FFC/Profile_ImportExport/dutA_test_pr.tgz my_archive

> restore appfw profile my_archive

This example restores the test_pr profile along with all bound objects (such as signatures, html error page, relaxation rules and so on) on the target NetScaler appliance.

You can use the following CLI commands to access man pages for additional details.

- man archive appfw profile
- man export appfw archive
- man import appfw archive
- man restore appfw profile
- man show appfw archive

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- man rm appfw archive

Exporting and Importing Application Firewall Profiles with the Configuration Utility

The configuration utility (GUI) is easier to use than the CLI. The utility performs both archive and export operations when you click **Export**. Similarly it executes both import and restore when you click **Import**. The configuration utility can access the local file system of the computer from which you access the utility. You can export a copy of the archive and save it on your local computer. You can then import this copy directly in the target appliance without having to manually transfer the archive file from one appliance to the other(s).

To export an application firewall profile by using the configuration utility

1. Navigate to **Configuration** > **Security** > **Application Firewall** > **Profiles**.
2. In the details pane, select a profile to export. Click **Actions** and select **Export** to download and save a copy in your computer's local file system.

To import an application firewall profile by using the configuration utility

1. Navigate to **Configuration** > **Security** > **Application Firewall** > **Profiles**.
2. In the details pane, click **Actions** and select **Import**. In the Import Application firewall Profile pane, the Import From* selection box gives you 2 options:

   URL: You can choose to import an archive by specifying a **URL**. When this option is selected, you must provide an absolute path for the archived file in the **URL** input box.

   File: You can choose to import an archive from the local **File**. When this option is selected, a **Local File** selection field is displayed. You can browse your computer's local files to select the target archive file.

   Click **Create** to import the specified archive. Successful completion of the import operation creates the profile configuration on the target appliance.

Highlights

- You can replicate the entire configuration (including all import objects as well as configured relaxation rules for the profile) on multiple appliances, without needing to repeat configuration steps, by using export and import profile functionality.
- The imported objects, such as signatures, WSDL, Schema, error page and so on, are included in the archived tar file and replicated on the target appliance.
- Customized field types are included in the archived tar file and replicated on the target appliance.
- The policy bindings of the archived profile are not replicated when the configuration is restored. You must configure the policy and bind it to the profile after importing the profile to the appliance.
- The name of the archive file can be up to 31 character long. As with profile names, an archive name must begin with an alphanumeric character or underscore and contain only alphanumeric and underscore (_), number (#), period (.), space ( ), colon (:), at (@), equals (=) or hyphen (-) characters.
- Comments associated with the archive should be descriptive enough to convey the purpose of the archived configuration. The maximum allowed length for a comment is 255 characters.
- The "clear config –force basic" command does not remove the archived profiles.
- The import and export profile functionality is supported in high availability (HA) deployments.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Debugging Tips

- Monitor the /var/log/ns.log during command executions to see if there are any ERROR messages.
- Additional logs (_restore.log, remove.log, import.log) are generated in the /var/tmp/folder. They can help debug issues during the corresponding operations. When these logs reach one MB in size, the log messages are purged to shrink the log file to one fourth of the original size.
- If the import command fails when you are using the URL option instead of the local file system, verify that DNS name server and route settings are accurately configured.
- If you are using the HTTPS protocol to import the archive, the command might fail if the HTTPS server requires client certificate authentication.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring and Using the Learning Feature

Sep 28, 2017

The learning feature is a repetitive pattern filter that observes activity on a web site or application protected by the application firewall, to determine what constitutes normal activity on that web site or application. It then generates a list of up to 2,000 suggested rules or exceptions (relaxations) for each security checks that includes support for the learning feature. Users normally find it easier to configure relaxations by using the learning feature than by entering the necessary relaxations manually.

The security checks that support the learning feature are:

- Start URL check
- Cookie Consistency check
- Form Field Consistency check
- Field Formats check
- CSRF Form Tagging check
- HTML SQL Injection check
- HTML Cross-Site Scripting check
- XML Denial-of-Service check
- XML Attachment check
- Web Services Interoperability check

You perform two different types of activities when using the learning feature. First, you enable and configure the feature to use it. You can use learning on all traffic to your protected web applications, or you can configure a list of IPs (called the *Add Trusted Learning Clients* list) from which the learning feature should generate recommendations. Second, after the feature has been enabled and has processed a certain amount of traffic to your protected web sites, you review the list of suggested rules and relaxations (learned rules) and mark each with one of the following designations:

- **Edit & Deploy.** The rule is pulled into the Edit dialog box so that you can modify it, and the modified form is deployed.
- **Deploy.** The unmodified learned rule is placed on the list of rules or relaxations for this security check.
- **Skip.** The learned rule is placed on a list of rules or relaxations that are not deployed. The learned rule is removed when skipped. However, as they are not added to relaxations, they might get learned again.

Learning is not performed only when relaxations are in place, except for field format rules. When rules are skipped, they are only removed from learned database. As relaxations are not added, they might get learned again. When rules are deployed, they are removed from learned database and also relaxations are added for the rules. As relaxations are added, they would not be learned again. For fieldformat protection, learning is performed irrespective of relaxations.

Although you can use the command line interface for basic configuration of the learning feature, the feature is designed primarily for configuration through the Application Firewall wizard or the configuration utility. You can perform only limited configuration of the learning feature by using the command line.

The wizard integrates configuration of learning features with configuration of the application firewall as a whole, and is therefore the easiest method for configuring this feature on a new NetScaler appliance or when managing a simple application firewall configuration. The configuration utility visualizer and manual interface both provide direct access to all learned rules for all security checks, and are therefore often preferable when you must review learned rules for a large number of security checks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The learning database is limited to 20 MB in size, which is reached after approximately 2,000 learned rules or relaxations are generated per security check for which learning is enabled. If you do not regularly review and either approve or ignore learned rules and this limit is reached, an error is logged to the NetScaler log and no more learned rules are generated until you review the existing learned rules and relaxations.

If learning stops because the database has reached its size limit, you can restart learning either by reviewing the existing learned rules and relaxations or by resetting the learning data. After learned rules or relaxations are approved or ignored, they are removed from the database. After you reset the learning data, all existing learning data is removed from the database and it is reset to its minimum size. When the database falls below 20 MB in size, learning restarts automatically.

To add new trusted network, provide network IP followed by subnet mask prefix length.

Example:

- bind appfw profile pr1 – trustedLearningClients 10.217.0.0/16
- bind appfw profile pr1 – trustedLearningClients 10.217.30.0/24

To configure the learning settings by using the command line interface

Specify the application firewall profile to be configured and, for each security check that you want to include in that profile, specify the minimum threshold or the percent threshold. The minimum threshold is an integer representing the minimum number of user sessions that the application firewall must process before it learns a rule or relaxation (default: 1). The percent threshold is an integer representing the percentage of user sessions in which the application firewall must observe a particular pattern (URL, cookie, field, attachment, or rule violation) before it learns a rule or relaxation (default: 0). Use the following commands:

- set appfw learningsettings <profileName> [-startURLMinThreshold <positive_integer>] [-startURLPercentThreshold <positive_integer>] [-cookieConsistencyMinThreshold <positive_integer>] [-cookieConsistencyPercentThreshold <positive_integer>] [-CSRFtagMinThreshold <positive_integer>] [-CSRFtagPercentThreshold <positive_integer>] [-fieldConsistencyMinThreshold <positive_integer>] [-fieldConsistencyPercentThreshold <positive_integer>] [-crossSiteScriptingMinThreshold <positive_integer>] [-crossSiteScriptingPercentThreshold <positive_integer>] [-SQLInjectionMinThreshold <positive_integer>] [-SQLInjectionPercentThreshold <positive_integer>] [-fieldFormatMinThreshold <positive_integer>] [-fieldFormatPercentThreshold <positive_integer>] [-XMLWSIMinThreshold <positive_integer>] [-XMLWSIPercentThreshold <positive_integer>] [-XMLAttachmentMinThreshold <positive_integer>] [-XMLAttachmentPercentThreshold <positive_integer>]
- save ns config

Example

The following example enables and configures the learning settings in the profile pr-basic for the HTML SQL Injection security check. This is an appropriate initial test bed learning configuration, where you have complete control over the traffic that is sent to the application firewall.

```
set appfw learningsettings pr-basic -SQLInjectionMinThreshold 10
set appfw learningsettings pr-basic -SQLInjectionPercentThreshold 70
save ns config
```

To reset learning settings to their defaults by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To remove any custom configuration of the learning settings for the specified profile and security check, and return the learning settings to their defaults, at the command prompt type the following commands:

- unset appfw learningsettings <profileName> [-startURLMinThreshold] [-startURLPercentThreshold] [-cookieConsistencyMinThreshold] [-cookieConsistencyPercentThreshold] [-CSRFtagMinThreshold] [-CSRFtagPercentThreshold] [-fieldConsistencyMinThreshold] [-fieldConsistencyPercentThreshold] [-crossSiteScriptingMinThreshold] [-crossSiteScriptingPercentThreshold] [-SQLInjectionMinThreshold] [-SQLInjectionPercentThreshold] [-fieldFormatMinThreshold] [-fieldFormatPercentThreshold] [-XMLWSIMinThreshold] [-XMLWSIPercentThreshold] [-XMLAttachmentMinThreshold] [-XMLAttachmentPercentThreshold]
- save ns config

To display the learning settings for a profile by using the command line interface

At the command prompt, type the following command:

show appfw learningsettings <profileName>

To display unreviewed learned rules or relaxations for a profile by using the command line interface

At the command prompt, type the following command:

show appfw learningdata <profileName> <securityCheck>

To remove specific unreviewed learned rules or relaxations from the learning database by using the command line interface

At the command prompt, type the following command:

rm appfw learningdata <profileName> (-startURL <expression> | -cookieConsistency <string> | (-fieldConsistency <string> <formActionURL>) | (-crossSiteScripting <string> <formActionURL>) | (-SQLInjection <string> <formActionURL>) | (-fieldFormat <string><formActionURL>) | (-CSRFTag <expression> <CSRFFormOriginURL>) | -XMLDoSCheck <expression> | -XMLWSICheck <expression> | -XMLAttachmentCheck <expression>) [-TotalXMLRequests]

Example

The following example removes all unreviewed learned relaxations for the pr-basic profile, HTML SQL Injection security check, that apply to the LastName form field.

rm appfw learningdata pr-basic -SQLInjection LastName

To remove all unreviewed learned data by using the command line interface

At the command prompt, type the following command:

reset appfw learningdata

To export learning data by using the command line interface

At the command prompt, type the following command:

export appfw learningdata <profileName> <securitycheck>[-target <string>]

Example

The following example exports learned relaxations for the pr-basic profile and the HTML SQL Injection security check to a comma-separated values (CSV) format file in the /var/learnt_data/ directory under the filename specified in the -target parameter.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
export appfw learningdata pr-basic SQLInjection -target sqli_ld
```

To configure the Learning feature by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the Profiles pane, select the profile, and then click Edit.
3. Click the Learning tab. At the top of the Learning tab is list of the security checks that are available in the current profile and that support the learning feature.
4. To configure the learning thresholds, select a security check, and then type the appropriate values in the following text boxes:
   - **Minimum number threshold.** Depending on which security check's learning settings you are configuring, the minimum number threshold might refer to the minimum number of total user sessions that must be observed, the minimum number of requests that must be observed, or the minimum number of times a specific form field must be observed, before a learned relaxation is generated. Default: 1
   - **Percentage of times threshold.** Depending on which security check's learning settings you are configuring, the percentage of times threshold might refer to the percentage of total observed user sessions that violated the security check, the percentage of requests, or the percentage of times a form field matched a particular field type, before a learned relaxation is generated. Default: 0
5. To remove all learned data and reset the learning feature, so that it must start its observations again from the beginning, click Remove All Learned Data.
   Note: This button removes only learned recommendations that have not been reviewed and either approved or skipped. It does not remove learned relaxations that have been accepted and deployed.
6. To restrict the learning engine to traffic from a specific set of IPs, click Trusted Learning Clients, and add the IP addresses that you want to use to the list.
   1. To add an IP address or IP address range to the Trusted Learning Clients list, click Add.
   2. In the Add Trusted Learning Clients dialog box, Trusted Clients IP list box, type the IP address or an IP address range in CIDR format.
   3. In the Comments text area, type a comment that describes this IP address or range.
   4. Click Create to add your new IP address or range to the list.
   5. To modify an existing IP address or range, click the IP address or range, and then click Open. Except for the name, the dialog box that appears is identical to the Add Trusted Learning Clients dialog box.
   6. To disable or enable an IP address or range, but leave it on the list, click the IP address or range, and then click Disable or Enable, as appropriate.
   7. To remove an IP address or range completely, click the IP address or range, and then click Remove.
7. Click Close to return to the Configure Application Firewall Profile dialog box.
8. Click Close to close the Configure Application Firewall Profile dialog box, and return to the Application Firewall Profile screen.

To review learned rules or relaxations by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. Select the security check for which you want to review learned rules or relaxations, and then click Manage Rules.
3. In the Manage Learned Rules dialog box, choose how you want to review the learned rules.
   - To review the actual learned patterns as displayed in the window, do nothing and proceed to the next step.
   - To review the learned data hierarchically as a branching tree, enabling you to choose general patterns that match many of the learned patterns, click Visualizer.
4. If you have chosen to review actual learned patterns, perform the following steps.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Select the first learned relaxation and choose how to handle it.
   - To modify and then accept the relaxation, click Edit & Deploy, edit the relaxation regular expression, and then click OK.
   - To accept the relaxation without modifications, click Deploy.
   - To remove the relaxation from the list without deploying it, click Skip.
2. Repeat the previous step to review each additional learned relaxation.

5. If you have chosen to use the Learning Visualizer, perform the following steps.
   1. In the branching hierarchical display, select a node that contains a learned pattern, and choose how to handle it.
      The screen area beneath the tree structure, under Regex of Selected Node, displays a generalized expression that matches all of the patterns in that node. If you want to display an expression that matches just one of the branches or just one of the leaves, select that branch or leaf.
      - To modify and then accept the learned relaxation, click Edit & Deploy, edit the relaxation regular expression, and then click OK.
      - To accept the relaxation without modifications, click Deploy.
      - To remove the modification from the list without deploying it, click Skip.
   2. Repeat the previous step to review other portions of the display.
   3. Click Close to return to the Manage Learned Rules dialog box.
6. Click Close to return to the Configure Application Firewall Profile dialog box.
7. Click Close to close the Configure Application Firewall Profile dialog box, and return to the Application Firewall Profile screen.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Supplemental Information about Profiles

Oct 01, 2013

Following is supplemental information about particular aspects of application firewall profiles. This information explains how to include special characters in a security check rule or relaxation, and how to use variables when configuring profiles.

Configuration Variable Support

Instead of using static values, to configure the application firewall's security checks and settings, you can now use standard NetScaler named variables. By creating variables, you can more easily export and then import configurations to new NetScaler appliances, or update existing NetScaler appliances from a single set of configuration files. This simplifies updates when you use a test bed setup to develop a complex application firewall configuration that is tuned for your local network and servers and then transfer that configuration to your production NetScaler appliances.

You create application firewall configuration variables in the same manner as you do any other NetScaler named variables, following standard NetScaler conventions. To create a named expression variable by using the configuration utility, you use the "[Add Expression dialog box]." To create a named expression variable by using the NetScaler command line, you use the add expression command followed by the appropriate parameter.

The following URLs and expressions can be configured with variables instead of static values:

- **Start URL** (-starturl)
- **Deny URL** (-denyurl)
- **Form Action URL** for *Form Field Consistency Check* (-fieldconsistency)
- **Action URL** for *XML SQL Injection Check* (-xmlSQLInjection)
- **Action URL** for X*ML Cross-Site Scripting Check* (-xmlXSS)
- **Form Action URL** for *HTML SQL Injection Check* (-sqlInjection)
- **Form Action URL** for *Field Format Check* (-fieldFormat)
- **Form Origin URL** and **Form Action URL** for *Cross-Site Request Forgery (CSRF) Check* (-csrfTag)
- **Form Action URL** for *HTML Cross-Site Scripting Check* (-crossSiteScripting)
- **Safe Object** (-safeObject)
- **Action URL** for *XML Denial-of-Service (XDoS) check* (-XMLDoS)
- **URL** for *Web Services Interoperability check* (-XMLWSIURL)
- **URL** for *XML Validation check* (-XMLValidationURL)
- **URL** for *XML Attachment check* (-XMLAttachmentURL)

For more information, see "Policies and Expressions."

To use a variable in the configuration, you enclose the variable name between two at (@) symbols and then use it exactly as you would the static value that it replaces. For example, if you are configuring the Deny URL check by using the configuration utility and want to add the named expression variable myDenyURL to the configuration, you would type @myDenyURL@ into the Add Deny URL dialog box, Deny URL text area. To do the same task by using the NetScaler command line, you would type add appfw profile <name> -denyURLAction @myDenyURL@.

PCRE Character Encoding Format

The NetScaler operating system supports direct entry of characters in the printable ASCII character set only—characters with hexadecimal codes between HEX 20 (ASCII 32) and HEX 7E (ASCII 127). To include a character with a code outside that range in your application firewall configuration, you must enter its UTF-8 hexadecimal code as a PCRE regular

© 1999-2017 Citrix Systems, Inc. All rights reserved.

expression.

A number of character types require encoding using a PCRE regular expression if you include them in your application firewall configuration as a URL, form field name, or Safe Object expression. They include:

- **Upper-ASCII characters.** Characters with encodings from HEX 7F (ASCII 128) to HEX FF (ASCII 255). Depending on the character map used, these encodings can refer to control codes, ASCII characters with accents or other modifications, non-Latin alphabet characters, and symbols not included in the basic ASCII set. These characters can appear in URLs, form field names, and safe object expressions.
- **Double-Byte characters.** Characters with encodings that use two 8-byte words. Double-byte characters are used primarily for representing Chinese, Japanese, and Korean text in electronic format. These characters can appear in URLs, form field names, and safe object expressions.
- **ASCII control characters.** Non-printable characters used to send commands to a printer. All ASCII characters with hexadecimal codes less than HEX 20 (ASCII 32) fall into this category. These characters should never appear in a URL or form field name, however, and would rarely if ever appear in a safe object expression.

The NetScaler appliance does not support the entire UTF-8 character set, but only the characters found in the following eight charsets:

- **English US (ISO-8859-1).** Although the label reads, “English US,” the application firewall supports all characters in the ISO-8859-1 character set, also called the Latin-1 character set. This character set fully represents most modern western European languages and represents all but a few uncommon characters in the rest.
- **Chinese Traditional (Big5).** The application firewall supports all characters in the BIG5 character set, which includes all of the Traditional Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in Hong Kong, Macau, Taiwan, and by many people of Chinese ethnic heritage who live outside of mainland China.
- **Chinese Simplified (GB2312).** The application firewall supports all characters in the GB2312 character set, which includes all of the Simplified Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in mainland China.
- **Japanese (SJIS).** The application firewall supports all characters in the Shift-JIS (SJIS) character set, which includes most characters (ideographs) commonly used in modern Japanese.
- **Japanese (EUC-JP).** The application firewall supports all characters in the EUC-JP character set, which includes all characters (ideographs) commonly used in modern Japanese.
- **Korean (EUC-KR).** The application firewall supports all characters in the EUC-KR character set, which includes all characters (ideographs) commonly used in modern Korean.
- **Turkish (ISO-8859-9).** The application firewall supports all characters in the ISO-8859-9 character set, which includes all letters used in modern Turkish.
- **Unicode (UTF-8).** The application firewall supports certain additional characters in the UTF-8 character set, including those used in modern Russian.

When configuring the application firewall, you enter all non-ASCII characters as PCRE-format regular expressions using the hexadecimal code assigned to that character in the UTF-8 specification. Symbols and characters within the normal ASCII character set, which are assigned single, two-digit codes in that character set, are assigned the same codes in the UTF-8

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

character set. For example, the exclamation point (!), which is assigned hex code 21 in the ASCII character set, is also hex 21 in the UTF-8 character set. Symbols and characters from another supported character set have a paired set of hexadecimal codes assigned to them in the UTF-8 character set. For example, the letter a with an acute accent (á) is assigned UTF-8 code C3 A1.

The syntax you use to represent these UTF-8 codes in the application firewall configuration is "\xNN" for ASCII characters; "\xNN\xNN" for non-ASCII characters used in English, Russian, and Turkish; and "\xNN\xNN\xNN" for characters used in Chinese, Japanese, and Korean. For example, if you want to represent a ! in an application firewall regular expression as a UTF-8 character, you would type \x21. If you want to include an á, you would type \xC3\xA1.

Note: Normally you do not need to represent ASCII characters in UTF-8 format, but when those characters might confuse a web browser or an underlying operating system, you can use the character's UTF-8 representation to avoid this confusion. For example, if a URL contains a space, you might want to encode the space as \x20 to avoid confusing certain browsers and web server software.
Below are examples of URLs, form field names, and safe object expressions that contain non-ASCII characters that must be entered as PCRE-format regular expressions to be included in the application firewall configuration. Each example shows the actual URL, field name, or expression string first, followed by a PCRE-format regular expression for it.

- A URL containing extended ASCII characters.

  Actual URL: http://www.josénuñez.com

  Encoded URL: ^http://www[.]jos\xC3\xA9nu\xC3\xB1ez[.]com$

- Another URL containing extended ASCII characters.

  Actual URL: http://www.example.de/trömso.html

  Encoded URL: ^http://www[.]example[.]de/tr\xC3\xB6mso[.]html$

- A form field name containing extended ASCII characters.

  Actual Name: nome_do_usuário

  Encoded Name: ^nome_do_usu\xC3\xA1rio$

- A safe object expression containing extended ASCII characters.

  Unencoded Expression [A-Z]{3,6}¥[1-9][0-9]{6,6}

  Encoded Expression: [A-Z]{3,6}\xC2\xA5[1-9][0-9]{6,6}

You can find a number of tables that include the entire Unicode character set and matching UTF-8 encodings on the Internet. A useful web site that contains this information is located at the following URL:

http://www.utf8-chartable.de/unicode-utf8-table.pl
For the characters in the table on this web site to display correctly, you must have an appropriate Unicode font installed on your computer. If you do not, the visual display of the character may be in error. Even if you do not have an appropriate font installed to display a character, however, the description and the UTF-8 and UTF-16 codes on this set of web pages will be correct.

Inverted PCRE Expressions

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

In addition to matching content that contains a pattern, you can match content that does not contain a pattern by using an inverted PCRE expression. To invert an expression, you simply include an exclamation point (!) followed by white space as the first character in the expression.

Note: If an expression consists only of an exclamation point with nothing following, the exclamation point is treated as a literal character, not syntax indicating an inverted expression.
The following application firewall commands support inverted PCRE expressions:

- Start URL (URL)
- Deny URL (URL)
- Form Field Consistency (form action URL)
- Cookie Consistency (form action URL)
- Cross Site Request Forgery (CSRF) (form action URL)
- HTML Cross-site Scripting (form action URL)
- Field Format (form action URL)
- Field Type (type)
- Confidential Field (URL)

Note: If the security check contains an isRegex flag or check box, it must be set to YES or checked to enable regular expressions in the field. Otherwise the contents of that field are treated as literal and no regular expressions (inverted or not) are parsed.

Disallowed Names for Application Firewall Profiles

The following names are assigned to built-in actions and profiles on the NetScaler appliance, and cannot be used as names for a user-created application firewall profile.

- AGRESSIVE
- ALLOW
- BASIC
- CLIENTAUTH
- COMPRESS
- CSSMINIFY
- DEFLATE
- DENY
- DNS-NOP
- DROP
- GZIP
- HTMLMINIFY
- IMGOPTIMIZE
- JSMINIFY
- MODERATE
- NOCLIENTAUTH
- NOCOMPRESS
- NONE
- NOOP
- NOREWRITE
- RESET
- SETASLEARNNSLOG_ACT

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- SETNSLOGPARAMS_ACT
- SETSYSLOGPARAMS_ACT
- SETTMSESSPARAMS_ACT
- SETVPNPARAMS_ACT
- SET_PREAUTHPARAMS_ACT
- default_DNS64_action
- dns_default_act_Cachebypass
- dns_default_act_Drop
- nshttp_default_profile
- nshttp_default_strict_validation
- nstcp_default_Mobile_profile
- nstcp_default_XA_XD_profile
- nstcp_default_profile
- nstcp_default_tcp_interactive_stream
- nstcp_default_tcp_lan
- nstcp_default_tcp_lan_thin_stream
- nstcp_default_tcp_lfp
- nstcp_default_tcp_lfp_thin_stream
- nstcp_default_tcp_lnp
- nstcp_default_tcp_lnp_thin_stream
- nstcp_internal_apps

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy Labels

Nov 19, 2013

A policy label consists of a set of policies, other policy labels, and virtual server-specific policy banks. The application firewall evaluates each policy bound to the policy label in order of priority. If the policy matches, it filters the connection as specified in the associated profile. Then it does whatever the Goto parameter specifies, which can be to terminate policy evaluation, go to the next policy, or go to the policy with the specified priority. If the Invoke parameter is set, it terminates processing of the current policy label and begins to process the specified policy label or virtual server.

To create an application firewall policy label by using the command line

At the command prompt, type the following commands:

- add appfw policylabel <labelName> http_req
- save ns config

Example

The following example creates a policy label named policylbl1.

```
add appfw policylabel policylbl1 http_req
save ns config
```

To bind a policy to a policy label by using the command line

At the command prompt, type the following commands:

- bind appfw policylabel <labelName> <policyName> <priority> [<gotoPriorityExpression>] [-invoke (<labelType> <labelName>)]
- save ns config

Example

The following example binds the policy policy1 to the policy label policylbl1 with a priority of 1.

```
bind appfw policylabel policylbl1 policy1 1
save ns config
```

To configure an application firewall policy label by using the configuration utility

1. Navigate to Security > Application Firewall > Policy Labels.
2. In the details pane, do one of the following:
   - To add a new policy label, click Add.
   - To configure an existing policy label, select the policy label and the click Open.

   The Create Application Firewall Policy Label or the Configure Application Firewall Policy Label dialog box opens. The dialog boxes are nearly identical.
3. If you are creating a new policy label, in the Create Application Firewall Policy Label dialog box, type a name for your new policy label.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Select Insert Policy to insert a new row and display a drop-down list with all existing application firewall policies.
5. Select the policy you want to bind to the policy label, or select New Policy to create a new policy and follow the instructions in [To create and configure a policy by using the configuration utility]. The policy that you selected or created is inserted into the list of globally bound application firewall policies.
6. Make any additional adjustments.
   - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
   - To modify the policy expression, double click that field to open the Configure Application Firewall Policy dialog box, where you can edit the policy expression.
   - To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
   - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression
7. Repeat steps 5 through 7 to bind any additional application firewall policies you want to the policy label.
8. Click Create or OK, and then click Close. A message appears in the status bar, stating that you have successfully created or modified the policy label.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policies

Mar 28, 2012

The application firewall uses two types of policies: firewall policies and auditing policies. Firewall policies control which traffic is sent to the application firewall. Auditing policies control the log server to which application firewall logs are sent.

Firewall policies can be complex because the policy rule can consist of multiple expressions in the NetScaler expressions language, which is a full-fledged object oriented programming language capable of defining with extreme precision exactly which connections to filter. Because firewall policies operate within the context of the application firewall, they must meet certain criteria that are connected to how the application firewall functions and what traffic is appropriately filtered by it. As long as you keep these criteria in mind, however, firewall policies are similar to policies for other NetScaler features. The instructions here do not attempt to cover all aspects of writing firewall policies, but only provide an introduction to policies and cover those criteria that are unique to the application firewall.

Auditing policies are simple because the policy rule is always ns_true. You need only specify the log server that you want to send logs to, the logging levels that you want to use, and a few other criteria that are explained in detail.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Firewall Policies

Feb 18, 2016

A firewall policy is a rule associated with a profile. The rule is an expression or group of expressions that defines the types of request/response pairs that the application firewall is to filter by applying the profile. Firewall policy expressions are written in the NetScaler expressions language, an object-oriented programming language with special features to support specific NetScaler functions. The profile is the set of actions that the application firewall is to use to filter request/response pairs that match the rule.

Firewall policies enable you to assign different filtering rules to different types of web content. Not all web content is alike. A simple web site that uses no complex scripting and accesses and handles no private data might require only the level of protection provided by a profile created with basic defaults. Web content that contains JavaScript-enhanced web forms or accesses an SQL database probably requires more tailored protection. You can create a different profile to filter that content, and create a separate firewall policy that can determine which requests are attempting to access that content. You then associate the policy expression with a profile you created and globally bind the policy to put it into effect.

The application firewall processes only HTTP connections, and therefore uses a subset of the overall NetScaler expressions language. The information here is limited to topics and examples that are likely to be useful when configuring the application firewall. Following are links to additional information and procedures for firewall policies:

- For procedures that explain how to create and configure a policy, see "Creating and Configuring Application Firewall Policies."
- For a procedure that explains in detail how to create a policy rule (expression), see "To create or configure an Application Firewall rule (expression)."
- For a procedure that explains how to use the Add Expression dialog box to create a policy rule, see "To add a firewall rule (expression) by using the Add Expression dialog box."
- For a procedure that explains how to view the current bindings for a policy, see "Viewing a Firewall Policy's Bindings."
- For procedures that explain how to bind an application firewall policy, see "Binding Application Firewall Policies."
- For detailed information about the NetScaler expressions language, see "Policies and Expressions."

Note

Application firewall evaluates the policies based on the configured priority and goto expressions. At the end of the policy evaluation, the last policy that evaluates to true is used and the security configuration of the corresponding profile is invoked for processing the request.

For example, Consider a scenario where there are 2 policies.

- Policy_1 is a generic policy with Expression=ns_true and has a corresponding profile_1 which is a basic profile. The priority is set to 100.
- Policy_2 is more specific with Expression=HTTP.REQ.URL.CONTAINS("XYZ") and has a corresponding profile_2 which is an advance profile. The GoTo Expression is set to NEXT and the priority is set to 95 which is a higher priority compared to Policy_1.

In this scenario, if the target string "XYZ" is detected in the URL of the processed request, Policy_2 match is triggered as it has a higher priority even though Policy_1 is also a match. However, as per the GoTo expression configuration of Policy_2, the policy evaluation continues and the next policy Policy_1 is also processed. At the end of the policy evaluation, Policy_1 evaluates as true and the basic security checks configured in Profile_1 are invoked.

If the Policy_2 is modified and the GoTo Expression is changed from **NEXT** to **END**, the processed request that has the target string "XYZ", triggers the Policy_2 match due to priority consideration and as per the GoTo expression configuration, the policy evaluation

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ends at this point. Policy_2 evaluates as true and the advanced security checks configured in Profile_2 are invoked.

Policy evaluation is completed in one pass. Once the policy evaluation is completed for the request and the corresponding profile actions are invoked, the request does not go through another round of policy evaluation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Creating and Configuring Application Firewall Policies

Aug 03, 2015

A firewall policy consists of two elements: a *rule*, and an associated *profile*. The rule selects the HTTP traffic that matches the criteria that you set, and sends that traffic to the application firewall for filtering. The profile contains the filtering criteria that the application firewall uses.

The policy rule consists of one or more expressions in the NetScaler expressions language. The NetScaler expressions syntax is a powerful, object-oriented programming language that enables you to precisely designate the traffic that you want to process with a specific profile. For users who are not completely familiar with the NetScaler expressions language syntax, or who prefer to configure their NetScaler appliance by using a web-based interface, the configuration utility provides two tools: the Prefix menu and the Add Expression dialog box. Both help you to write expressions that select exactly the traffic that you want to process. Experienced users who are thoroughly familiar with the syntax may prefer to use the NetScaler command line to configure their NetScaler appliances.

Note: In addition to the default expressions syntax, for backward compatibility the NetScaler operating system supports the NetScaler classic expressions syntax on NetScaler Classic and nCore appliances and virtual appliances. Classic expressions are not supported on NetScaler Cluster appliances and virtual appliances. Current NetScaler users who want to migrate existing configurations to the NetScaler Cluster must migrate any policies that contain classic expressions to the default expressions syntax.
For detailed information about the NetScaler expressions languages, see "Policies and Expressions."

You can create a firewall policy by using the configuration utility or the NetScaler command line.

To create and configure a policy by using the command line interface

At the command prompt, type the following commands:

- add appfw policy <name> <rule> <profileName>
- save ns config

Example

The following example adds a policy named pl-blog, with a rule that intercepts all traffic to or from the host blog.example.com, and associates that policy with the profile pr-blog. This is an appropriate policy to protect a blog hosted on a specific hostname.

```
add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com")" pr-blog
```

To create and configure a policy by using the configuration utility

1. Navigate to Security > Application Firewall > Policies.
2. In the details pane, do one of the following:
   - To create a new firewall policy, click Add. The Create Application Firewall Policy is displayed.
   - To edit an existing firewall policy, select the policy, and then click Edit.

   The Create Application Firewall Policy or Configure Application Firewall Policy is displayed.
3. If you are creating a new firewall policy, in the Create Application Firewall Policy dialog box, Policy Name text box, type a name for your new policy.
   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 128 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If you are configuring an existing firewall policy, this field is read-only. You cannot modify it.

4. Select the profile that you want to associate with this policy from the Profile drop-down list. You can create a new profile to associate with your policy by clicking New, and you can modify an existing profile by clicking Modify.
5. In the Expression text area, create a rule for your policy.
   - You can type a rule directly into the text area.
   - You can click Prefix to select the first term for your rule, and follow the prompts. See "To Create an Application Firewall Rule (Expression)" for a complete description of this process.
   - You can click Add to open the Add Expression dialog box, and use it to construct the rule. See "The Add Expression Dialog Box" for a complete description of this process.
6. Click Create or OK, and then click Close.

To create or configure an Application Firewall rule (expression)

The policy rule, also called the *expression*, defines the web traffic that the application firewall filters by using the profile associated with the policy. Like other NetScaler policy rules (or *expressions*), application firewall rules use NetScaler expressions syntax. This syntax is powerful, flexible, and extensible. It is too complex to describe completely in this set of instructions. You can use the following procedure to create a simple firewall policy rule, or you can read it as an overview of the policy creation process.

1. If you have not already done so, navigate to the appropriate location in the Application Firewall wizard or the NetScaler configuration utility to create your policy rule:
   - If you are configuring a policy in the Application Firewall wizard, in the navigation pane, click Application Firewall, then in the details pane click Application Firewall Wizard, and then navigate to the Specify Rule screen.
   - If you are configuring a policy manually, in the navigation pane, expand Application Firewall, then Policies, and then Firewall. In the details pane, to create a new policy, click Add. To modify an existing policy, select the policy, and then click Open.
2. On the Specify Rule screen, the Create Application Firewall Profile dialog box, or the Configure Application Firewall Profile dialog box, click Prefix, and then choose the prefix for your expression from the drop-down list. Your choices are:
   - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

   After you choose a prefix, the application firewall displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom.

3. Choose your next term.

   If you chose HTTP as your prefix, your only choice is REQ, which specifies the Request/Response pair. (The application firewall operates on the request and response as a unit instead of on each separately.) If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

   When you have decided which term you want, double-click it to insert it into the Expression window.

4. Type a period after the term you just chose. You are then prompted to choose your next term, as described in the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

previous step. When a term requires that you type a value, fill in the appropriate value. For example, if you choose HTTP.REQ.HEADER(""), type the header name between the quotation marks.

5. Continue choosing terms from the prompts and filling in any values that are needed, until your expression is finished. Following are some examples of expressions for specific purposes.

   - **Specific web host.** To match traffic from a particular web host:

     HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
     For shopping.example.com, substitute the name of the web host that you want to match.

   - **Specific web folder or directory.** To match traffic from a particular folder or directory on a Web host:

     HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
     For www.example.com, substitute the name of the web host. For folder, substitute the folder or path to the content that you want to match. For example, if your shopping cart is in a folder called /solutions/orders, you substitute that string for folder.

   - **Specific type of content: GIF images.** To match GIF format images:

     HTTP.REQ.URL.ENDSWITH(".gif")
     To match other format images, substitute another string in place of .gif.

   - **Specific type of content: scripts.** To match all CGI scripts located in the CGI-BIN directory:

     HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
     To match all JavaScripts with .js extensions:

     HTTP.REQ.URL.ENDSWITH(".js")
     For more information about creating policy expressions, see "Policies and Expressions."

   Note: If you use the command line to configure a policy, remember to escape any double quotation marks within NetScaler expressions. For example, the following expression is correct if entered in the configuration utility:
   HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
   If entered at the command line, however, you must type this instead:

   HTTP.REQ.HEADER(\"Host\").EQ(\"shopping.example.com\")

To add a firewall rule (expression) by using the Add Expression dialog box

The Add Expression dialog box (also referred to as the Expression Editor) helps users who are not familiar with the NetScaler expressions language to construct a policy that matches the traffic that they want to filter.

1. If you have not already done so, navigate to the appropriate location in the Application Firewall wizard or the NetScaler configuration utility:
   - If you are configuring a policy in the Application Firewall wizard, in the navigation pane, click Application Firewall, then in the details pane click Application Firewall Wizard, and then navigate to the Specify Rule screen.
   - If you are configuring a policy manually, in the navigation pane, expand Application Firewall, then Policies, and then Firewall. In the details pane, to create a new policy, click Add. To modify an existing policy, select the policy, and then click Open.
2. On the Specify Rule screen, in the Create Application Firewall Profile dialog box, or in the Configure Application Firewall Profile dialog box, click Add.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Add Expression dialog box, in the Construct Expression area, in the first list box, choose one of the following prefixes:
   - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol. The default choice.
   - **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.
4. In the second list box, choose your next term. The available terms differ depending on the choice you made in the previous step, because the dialog box automatically adjusts the list to contain only those terms that are valid for the context. For example, if you selected HTTP in the previous list box, the only choice is REQ, for requests. Because the application firewall treats requests and associated responses as a single unit and filters both, you do not need to specific responses separately. After you choose your second term, a third list box appears to the right of the second. The Help window displays a description of the second term, and the Preview Expression window displays your expression.
5. In the third list box, choose the next term. A new list box appears to the right, and the Help window changes to display a description of the new term. The Preview Expression window updates to display the expression as you have specified it to that point.
6. Continue choosing terms, and when prompted filling in arguments, until your expression is complete. If you make a mistake or want to change your expression after you have already selected a term, you can simply choose another term. The expression is modified, and any arguments or additional terms that you added after the term that you modified are cleared.
7. When you have finished constructing your expression, click OK to close the Add Expression dialog box. Your expression is inserted into the Expression text area.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding Application Firewall Policies

Mar 11, 2018

After you have configured your application firewall policies, you bind them to Global or a bind point to put them into effect. After binding, any request or response that matches an application firewall policy is transformed by the profile associated with that policy.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler OS, policy priorities work in reverse order - the higher the number, the lower the priority.

Because the application firewall feature implements only the first policy that a request matches, not any additional policies that it might also match, policy priority is important for achieving the results that you intend. If you give your first policy a low priority (such as 1000), you configure the application firewall to perform it only if other policies with a higher priority do not match a request. If you give your first policy a high priority (such as 1), you configure the application firewall to perform it first, and skip any other policies that might also match. You can leave yourself plenty of room to add other policies in any order, without having to reassign priorities, by setting priorities with intervals of 50 or 100 between each policy when you bind your policies.

For more information about binding policies on the NetScaler appliance, see "Policies and Expressions."

To bind an application firewall policy by using the command line interface

At the command prompt, type the following commands:

- bind appfw global <policyName> <priority>
- save ns config
- bind appfw profile <profile_name> -crossSiteScripting data

Example

The following example binds the policy named pl-blog and assigns it a priority of 10.

```
bind appfw global pl-blog 10
save ns config
```

To bind an application firewall policy by using the configuration utility

1. Do one of the following:
   - Navigate to Security > Application Firewall, and in the details pane, click Application Firewall policy manager.
   - Navigate to Security > Application Firewall > Policies > Firewall Policies, and in the details pane, click Policy Manager.
2. In the Application Firewall Policy Manager dialog, choose the bind point to which you want to bind the policy from the drop-down list. The choices are:
   - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance, and are applied before any other policies.
   - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
   - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting

© 1999-2017 Citrix Systems, Inc. All rights reserved.

CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.

- **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance.
- **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.
- **None.** Do not bind the policy to any bind point.

3. Click Continue. A list of existing application firewall policies appears.
4. Select the policy you want to bind by clicking it.
5. Make any additional adjustments to the binding.
   - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
   - To modify the policy expression, double click that field to open the Configure Application Firewall Policy dialog box, where you can edit the policy expression.
   - To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
   - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression
6. Repeat steps 3 through 6 to add any additional application firewall policies you want to globally bind.
7. Click OK. A message appears in the status bar, stating that the policy has been successfully bound.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Viewing a Firewall Policy's Bindings

Aug 03, 2015

You can quickly check to determine what bindings are in place for any firewall policy by viewing the bindings in the configuration utility.

To view bindings for an application firewall policy

1. Navigate to Security > Application Firewall > Policies > Firewall Policies
2. In the details pane, select the policy that you want to check, and then click Show Bindings. The Binding Details for Policy: Policy message box is displayed, with a list of bindings for the selected policy.
3. Click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Supplemental Information about Application Firewall Policies

Aug 03, 2015

Following is supplemental information about particular aspects of application firewall policies that system administrators who manage the application firewall might need to know.

Correct but Unexpected Behavior

Web application security and modern web sites are complex. In a number of scenarios, a NetScaler policy might cause the application firewall to behave differently in certain situations than a user who is familiar with policies would normally expect. Following are a number of cases where the application firewall may behave in an unexpected fashion.

- **Request with a missing HTTP Host header and an absolute URL.** When a user sends a request, in the majority of cases the request URL is relative. That is, it takes as its starting point the Referer URL, the URL where the user's browser is located when it sends the request. If a request is sent without a Host header, and with a relative URL, the request is normally blocked both because it violates the HTTP specification and because a request that fails to specify the host could under some circumstances constitute an attack. If a request is sent with an absolute URL, however, even if the Host header is missing, the request bypasses the application firewall and is forwarded to the web server. Although such a request violates the HTTP specification, it poses no possible threat because an absolute URL contains the host.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Auditing Policies

Jun 12, 2014

Auditing policies determine the messages that are generated and logged during an Application Firewall session. These messages are logged in SYSLOG format to the local NSLOG server or to an external logging server. Different types of messages are logged on the basis of the level of logging selected.

To create an auditing policy, you must first create either an NSLOG server or a SYSLOG server. After specifying the server, you create the policy and specify the type of log and the server to which logs are sent.

To create an auditing server by using the command line interface

You can create two different types of auditing server: an NSLOG server or a SYSLOG server. The command names are different, but the parameters for the commands are the same.

To create an auditing server, at the NetScaler command prompt, type the following commands:

- add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat (**MMDDYYYY** | **DDMMYYYY**)] [-logFacility <logFacility>] [-tcp (**NONE** | **ALL**)] [-acl (**ENABLED** | **DISABLED**)] [-timeZone (**GMT_TIME** | **LOCAL_TIME**)] [-userDefinedAuditlog (**YES** | **NO**)] [-appflowExport (**ENABLED** | **DISABLED**)]
- save ns config

Example

The following example creates a syslog server named syslog1 at IP 10.124.67.91, with loglevels of emergency, critical, and warning, log facility set to LOCAL1, that logs all TCP connections:

```
add audit syslogAction syslog1 10.124.67.91 -logLevel emergency critical warning -logFacility
LOCAL1 -tcp ALL
save ns config
```

To modify or remove an auditing server by using the command line interface

- To modify an auditing server, type the set audit <type> command, the name of the auditing server, and the parameters to be changed, with their new values.
- To remove an auditing server, type the rm audit <type> command and the name of the auditing server.

Example

The following example modifies the syslog server named syslog1 to add errors and alerts to the log level:

```
set audit syslogAction syslog1 10.124.67.91 -logLevel emergency critical warning alert error
-logFacility LOCAL1 -tcp ALL
save ns config
```

To create or configure an auditing server by using the configuration utility

1. Navigate to Security > Application Firewall > Policies > Auditing.
2. In the details pane, click the Server tab.
3. Do one of the following:
   - To add a new auditing server, click Add.
   - To modify an existing auditing server, select the server, and then click Edit.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. In the Create Auditing Server or Configure Auditing Server dialog box, set the following parameters:
   - Name
   - Auditing Type
   - IP Address
   - Port
   - Log Levels
   - Log Facility
   - TCP Logging
   - ACL Logging
   - User-Configurable Log Messages
   - AppFlow Logging
   - Date Format
   - Time Zone
5. Click Create or OK.

To create an auditing policy by using the command line interface

You can create an NSLOG policy or a SYSLOG policy. The type of policy must match the type of server. The command names for the two types of policy are different, but the parameters for the commands are the same.

At the command prompt, type the following commands:

- add audit syslogPolicy <name> [-rule <expression>] [-action <string>]
- save ns config

Example

The following example creates a policy named syslogP1 that logs application firewall traffic to a syslog server named syslog1.

```
add audit syslogPolicy syslogP1 -rule "ns_true" -action syslog1
save ns config
```

To configure an auditing policy by using the command line interface

At the command prompt, type the following commands:

- set audit syslogPolicy <name> [-rule <expression>] [-action <string>]
- save ns config

Example

The following example modifies the policy named syslogP1 to log application firewall traffic to a syslog server named syslog2.

```
set audit syslogPolicy syslogP1 -rule "ns_true" -action syslog2
save ns config
```

To configure an auditing policy by using the configuration utility

1. Navigate to Security > Application Firewall > Policies > Auditing.
2. In the details pane, do one of the following:
   - To add a new policy, click Add.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- To modify an existing policy, select the policy, and then click Edit.

3. In the Create Auditing Policy or Configure Auditing Policy dialog box, set the following parameters:
   - Name
   - Auditing Type
   - Server
4. Click Create or OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Imports

Mar 28, 2012

Several application firewall features make use of external files that you upload to the application firewall when configuring it. Using the configuration utility, you manage those files in the Imports pane, which has four tabs corresponding to the four types of files you can import: HTML error objects, XML error objects, XML schemas, and Web Services Description Language (WSDL) files. Using the NetScaler command line, you can import these types of files, but you cannot export them.

HTML Error Object

When a user's connection to an HTML or Web 2.0 page is blocked, or a user asks for a non-existent HTML or Web 2.0 page, the application firewall sends an HTML-based error response to the user's browser. When configuring which error response the application firewall should use, you have two choices:

- You can configure a redirect URL, which can be hosted on any Web server to which users also have access. For example, if you have a custom error page on your Web server, 404.html, you can configure the application firewall to redirect users to that page when a connection is blocked.
- You can configure an HTML error object, which is an HTML-based Web page that is hosted on the application firewall itself. If you choose this option, you must upload the HTML error object to the application firewall. You do that in the Imports pane, on the HTML Error Object tab.

The error object must be a standard HTML file that contains no non-HTML syntax except for application firewall error object customization variables. It cannot contain any CGI scripts, server-parsed code, or PHP code. The customization variables enable you to embed troubleshooting information in the error object that the user receives when a request is blocked. While most requests that the application firewall blocks are illegitimate, even a properly configured application firewall can occasionally block legitimate requests, especially when you first deploy it or after you make significant changes to your protected Web sites. By embedding information in the error page, you provide the user with the information that he or she needs to give to the technical support person so that any issues can be fixed.

The application firewall error page customization variables are:

- ${NS_TRANSACTION_ID}. The transaction ID that the application firewall assigned to this transaction.
- ${NS_APPFW_SESSION_ID}. The application firewall session ID.
- ${NS_APPFW_VIOLATION_CATEGORY}. The specific application firewall security check or rule that was violated.
- ${NS_APPFW_VIOLATION_LOG}. The detailed error message associated with the violation.
- ${COOKIE("<CookieName>")}. The contents of the specified cookie. For <CookieName>, substitute the name of the specific cookie that you want to display on the error page. If you have multiple cookies whose contents you want to display for troubleshooting, you can use multiple instances of this customization variable, each with the appropriate cookie name.
  Note: If you have blocking enabled for the Cookie Consistency Check, any blocked cookies are not displayed on the error page because the application firewall blocks them.

To use these variables, you embed them in the HTML or XML of the error page object as if they were an ordinary text string. When the error object is displayed to the user, for each customization variable the application firewall substitutes the information to which the variable refers. An example HTML error page that uses custom variables is shown below.

<!doctype html public "-//w3c//dtd html 4.0//en"> <html> <head> <title>Page Not Accessible</title> </head> <body> <h1>Page Not Accessible</h1> <p>The page that you accessed is not available. Y
To use this error page, copy it into a text or HTML editor. Substitute the appropriate local information for the following variables, which are enclosed in square brackets to distinguish them from the NetScaler variables. (Leave those unchanged.):

- **[homePage]**. The URL for your web site's home page.
- **[helpDeskEmailAddress]**. The email address that you want users to use to report blocking incidents.
- **[helpDeskPhoneNumber]**. The phone number that you want users to call to report blocking incidents.
- **[cookieName]**. The name of the cookie whose contents you want to display on the error page.

XML Error Object

When a user's connection to an XML page is blocked, or a user asks for a nonexistent XML application, the application firewall sends an XML-based error response to the user's browser. You configure the error response by uploading an XML-based error page to the application firewall in the Imports Pane, on the XML Error Object tab. All XML error responses are hosted on the application firewall. You cannot configure a redirect URL for XML applications.

Note: You can use the same customization variables in an XML error object as in an HTML error object.

XML Schema

When the application firewall performs a validation check on a user's request for an XML or Web 2.0 application, it can validate the request against the XML schema or design type document (DTD) for that application and reject any request that does not follow the schema or DTD. Both an XML schema and a DTD are standard XML configuration files that describe the structure of a specific type of XML document.

WSDL

When the application firewall performs a validation check on a user's request for an XML SOAP-based web service, it can validate the request against the web services type definition (WSDL) file for that web service. A WSDL file is a standard XML SOAP configuration file that defines the elements of a specific XML SOAP web service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Importing and Exporting Files

Oct 02, 2014

You can import HTML or XML error objects, XML schemas, DTDs, and WSDLs to the application firewall by using the configuration utility or the command line. You can edit any of these files in a web-based text area after importing them, to make small changes directly on the NetScaler ADC instead of having to make them on your computer and then reimport them. Finally, you can export any of these files to your computer, or delete any of these files, by using the configuration utility.

Note: You cannot delete or export an imported file by using the command line.

To import a file by using the command line interface

At the command prompt, type the following commands:

- import appfw htmlerrorpage <src> <name>
- save ns config

Example

The following example imports an HTML error object from a file named error.html and assigns it the name HTMLError.

```
import htmlerrorpage error.html HTMLError
save ns config
```

To import a file by using the configuration utility

Before you attempt to import an XML schema, DTD, or WSDL file, or an HTML or XML error object from a network location, verify that the NetScaler ADC can connect to the Internet or LAN computer where the file is located. Otherwise, you cannot import the file or object.

1. Navigate to Security > Application Firewall > Imports.
2. Navigate to Application Firewall > Imports.
3. In the Application Firewall Imports pane, select the tab for the type of file you want to import, and then click Add.
   The tabs are HTML Error Page, XML Error Page, XML Schema or WSDL. The upload process is identical on all four tabs from the user point of view.

4. Fill in the dialog fields.
   - **Name**—A name for the imported object.
   - **Import From**—Choose the location of the HTML file, XML file, XML schema or WSDL that you want to import in the drop-down list:
     - **URL:** A web URL on a website accessible to the ADC.
     - **File:** A file on a local or networked hard disk or other storage device.
     - **Text:** Type or paste the text of the custom response directly into a text field in the configuration utility.

     The third text box changes to the appropriate value. The three possible values are provided below.

   - **URL**—Type the URL into the text box.
   - **File**—Type the path and filename to the HTML file directly, or click Browse and browse to the HTML file.
   - **Text**—The third field is removed, leaving a blank space.
5. Click Continue. The File Contents dialog is displayed. If you chose URL or File, the File Contents text box contains the HTML file that you specified. If you chose Text, the File Contents text box is empty.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

6. If you chose Text, type or copy and paste the custom response HTML that you want to import.
7. Click Done.
8. To delete an object, select the object, and then click Delete.

To export a file by using the configuration utility

Before you attempt to export an XML schema, DTD, or WSDL file, or an HTML or XML error object, verify that the application firewall appliance can access the computer where the file is to be saved. Otherwise, you cannot export the file.

1. Navigate to Security > Application Firewall > Imports.
2. In the Application Firewall Imports pane, select the tab for the type of file you want to export.
   The export process is identical on all four tabs from the user point of view.
3. Select the file that you want to export.
4. Expand the Action drop-down list, and select Export.
5. In the dialog box, choose Save File and click OK.
6. In the Browse dialog box, navigate to the local file system and directory where you want to save the exported file, and click Save.

To edit an HTML or XML Error Object in the configuration utility

You edit the text of HTML and XML error objects in the configuration utility without exporting and then reimporting them.

1. Navigate to Security > Application Firewall > Imports, and then select the tab for the type of file that you want to modify.
2. Navigate to Application Firewall > Imports, and then select the tab for the type of file that you want to modify.
3. Select the file that you want to modify, and then click Edit.
   The text of the HTML or XML error object is displayed in a browser text area. You can modify the text by using the standard browser-based editing tools and methods for your browser.

   Note: The edit window is designed to allow you to make minor changes to your HTML or XML error object. To make extensive changes, you may prefer to export the error object to your local computer and use standard HTML or XML web page editing tools.
4. Click OK, and then click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Global Configuration

Sep 12, 2013

The application firewall global configuration affects all profiles and policies. The Global Configuration items are:

- **Engine Settings.** A collection of global settings—session cookie name, session time-out, maximum session lifetime, logging header name, undefined profile, default profile, and import size limit—that pertain to all connections that the application firewall processes, rather than to a specific subset of connections.
- **Confidential Fields.** A set of form fields in web forms that contain sensitive information that should not be logged to the application firewall logs. Form fields such as password fields on a logon page or credit card information on a shopping cart checkout form are normally designated as confidential fields.
- **Field Types.** The list of web form field types used by the Field Formats security check. Each of these field types is defined by a PCRE-compliant regular expression that defines the type of data and the minimum/maximum length of data that should be allowed in that type of form field.
- **XML Content Types.** The list of content types recognized as XML and subjected to XML-specific security checks. Each of these content types is defined by a PCRE-compliant regular expression that defines the exact MIME type assigned to that content.
- **JSON Content Types.** The list of content types recognized as JSON and subjected to JSON-specific security checks. Each of these content types is defined by a PCRE-compliant regular expression that defines the exact MIME type assigned to that content.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Engine Settings

Oct 07, 2014

The engine settings affect all requests and responses that the application firewall processes. They include the following items:

- **Cookie name**—The name of the cookie that stores the NetScaler session ID.
- **Session timeout**—The maximum inactive period allowed. If a user session shows no activity for this length of time, the session is terminated and the user is required to reestablish it by visiting a designated start page.
- **Cookie post-encrypt prefix**—The string that precedes the encrypted portion of any encrypted cookies.
- **Maximum session lifetime**—The maximum amount of time, in seconds, that a session is allowed to remain live. After this period is reached, the session is terminated and the user is required to reestablish it by visiting a designated start page. This setting cannot be less then the session timeout. To disable this setting, so that there is no maximum session lifetime, set the value to zero (0).
- **Logging header name**—The name of the HTTP header that holds the Client IP, for logging.
- **Undefined profile**—The profile applied when the corresponding policy action evaluates as undefined.
- **Default profile**—The profile applied to connections that do not match a policy.
- **Import size limit**—The maximum cumulative total byte count of all files imported to the ADC, including signatures, WSDLs, schemas, HTML and XML error pages. During an import, if the size of the imported object would cause the cumulative total sizes of all imported files to exceed the configured limit, the import operation fails and the ADC displays the following error message: *ERROR: Import failed - exceeding the configured total size limit on the imported objects*.
- **Learn message rate limit**—The maximum number of requests and responses per second that the learning engine is to process. Any additional requests or responses over this limit are not sent to the learning engine.
- **Entity decoding**—Decode HTML entities when running application firewall checks.
- **Log malformed request**—Enable logging of malformed HTTP requests.
- **Use configurable secret key**—Use a configurable secret key for application firewall operations.
- **Reset learned data**—Remove all learned data from the application firewall. Restarts the learning process by collecting fresh data.
-
-

Two settings, *Reset Learned Data* and *Signatures Auto-Update*, are found in different places depending on whether you use the command line or the configuration utility to configure your application firewall. When using the command line, you configure Reset Learned Data by using the reset appfw learningdata command, which takes no parameters and has no other functions. You configure Signatures Auto-Update in the set appfw settings command: the -signatureAutoUpdate parameter enables or disables auto-updating of the signatures, and -signatureUrl configures the URL which hosts the updated signatures file.

When using the configuration utility, you configure Reset Learned Data in Security > Application Firewall > Engine Settings; the Reset Learned Data button is at the bottom of the dialog box. You configure Signatures Auto-Update for each set of signatures in Security > Application Firewall > Signatures, by selecting the signatures file, clicking the right mouse button and selecting Auto Update Settings.

Normally, the default values for the application firewall settings are correct. If the default settings cause a conflict with other servers or cause premature disconnection of your users, however, you might need to modify them.

To configure engine settings by using the command line interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

At the command prompt, type the following commands:

- set appfw settings [-sessionCookieName <name>] [-sessionTimeout <positiveInteger>] [-sessionLifetime <positiveInteger>] [-clientIPLoggingHeader <headerName>] [-undefaction <profileName>] [-defaultProfile <profileName>] [-importSizeLimit <positiveInteger>] [-logMalformedReq (ON | OFF)] [-signatureAutoUpdate (ON | OFF)] [-signatureUrl <expression>] [-cookiePostEncryptPrefix <string>] [-entityDecoding (ON | OFF)] [-useConfigurableSecretKey (ON | OFF)] [-learnRateLimit <positiveInteger>]
- save ns config

Example

```
set appfw settings -sessionCookieName citrix-appfw-id -sessionTimeout 3600
-sessionLifetime 14400 -clientIPLoggingHeader NS-AppFW-Client-IP -undefaction APPFW_RESET
-defaultProfile APPFW_RESET -importSizeLimit 4096
save ns config
```

To configure engine settings by using the configuration utility

1. Navigate to Security > Application Firewall
2. In the details pane, click Change Engine Settings.
3. In the Application Firewall Engine Settings dialog box, set the following parameters:
   - Cookie Name
   - Session Timeout
   - Cookie Post Encrypt Prefix
   - Maximum Session Lifetime
   - Logging Header Name
   - Undefined Profile
   - Default Profile
   - Import Size Limit
   - Learn Messages Rate Limit
   -
   - Entity Decoding
   - Log Malformed Request
   - Use Secret Key
   - Learn Message Rate Limit
   - Signatures Auto Update
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Confidential Fields

Jun 12, 2014

You can designate web-form fields as confidential to protect the information users type into them. Normally, any information a user types into a web form on one of your protected web servers is logged in the NetScaler logs. The information typed into a web-form field designated as confidential, however, is not logged. That information is saved only where the web site is configured to save such data, normally in a secure database.

Common types of information that you may want to protect with a confidential field designation include:

- Passwords
- Credit card numbers, validation codes, and expiration dates
- Social security numbers
- Tax ID numbers
- Home addresses
- Private telephone numbers

In addition to being good practice, proper use of confidential field designations may be necessary for PCI-DSS compliance on ecommerce servers, HIPAA compliance on servers that manage medical information in the United States, and compliance with other data protection standards.

Important: In the following two cases, the Confidential Field designation does not function as expected:

- If a Web form has either a confidential field or an action URL longer than 256 characters, the field or action URL is truncated in the NetScaler logs.
- With certain SSL transactions, the logs are truncated if either the confidential field or the action URL is longer than 127 characters.

In either of these cases, the application firewall masks a fifteen-character string with the letter "x," instead of the normal eight character string. To ensure that any confidential information is removed, the user must use form field name and action URL expressions that match the first 256, or (in cases where SSL is used) the first 127 characters.

To configure your application firewall to treat a web-form field on a protected web site as confidential, you add that field to the Confidential Fields list. You can enter the field name as a string, or you can enter a PCRE-compatible regular expression specifying one or more fields. You can enable the confidential-field designation when you add the field, or you can modify the designation later.

To add a confidential field by using the command line interface

At the command prompt, type the following commands:

- add appfw confidField <fieldName> <url> [-isRegex (REGEX | NOTREGEX)] [-comment "<string>"] [-state (ENABLED | DISABLED)]
- save ns config

Example

The following example adds all web form fields whose names begin with Password to the confidential fields list.

```
add appfw confidField Password "https?://www[.]example[.]com/[^<>]*[^a-z]password[0-9a-z._-]*[.](asp|cgi|htm|html|htp|js|php)" -isRegex REGEX -state ENABLED
save ns config
```

To modify a confidential field by using the command line interface

At the command prompt, type the following commands:

- set appfw confidField <fieldName> <url> [-isRegex (REGEX | NOTREGEX)] [-comment "<string>"] [-state (ENABLED | DISABLED)]
- save ns config

Example

The following example modifies the confidential field designation to add a comment.

```
set appfw confidField Password "https?://www[.]example[.]com/[^<>]*[^a-z]password[0-9a-z._-]*[.](asp|cgi|htm|html|htp|js|php)" -comment "Protect password fields." -isRegex REGEX -state ENABLED
save ns config
```

To remove a confidential field by using the command line interface

At the command prompt, type the following commands:

- rm appfw confidField <fieldName> <url>
- save ns config

To configure a confidential field by using the configuration utility

1. Navigate to Security > Application Firewall.
2. In the details pane, under Settings, click Manage Confidential Fields.
3. In the Manage Confidential Fields dialog box, do one of the following:
   - To add a new form field to the list, click Add.
   - To change an existing confidential field designation, select the field, and then click Edit.

   The Application Firewall Confidential Fields dialog box appears.

   Note: If you select an existing confidential field designation and then click Add, the Create Confidential Form Field dialog box displays the information for that confidential field. You can modify that information to create your new confidential field.
4. In the dialog box, fill out the elements. They are:
   - **Enabled check box.** Select or clear to enable/disable this confidential field designation.
   - **Is form field name a regular expression check box.** Select or clear to enable PCRE-format regular expressions in the form field name.
   - **Field Name.** Enter a literal string or PCRE-format regular expression that either represents a specific field name or that matches multiple fields with names that follow a pattern.
   - **Action URL.** Enter a literal URL or a regular expression that defines one or more URLs of the web page(s) on which the web form(s) that contains the confidential field are located.
   - **Comments.** Enter a comment. Optional.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. Click Create or OK.
6. To remove a confidential field designation from the confidential fields list, select the confidential field listing you want to remove, then click Remove to remove it, and then click OK to confirm your choice.
7. When you have finished adding, modifying, and removing confidential field designations, click Close.

Examples

Following are some regular expressions that define form field names that you might find useful:

- ^passwd_ (Applies confidential-field status to all field names that begin with the “passwd_” string.)
- ^(([0-9a-zA-Z._-]*||\\x[0-9A-Fa-f][0-9A-Fa-f])+-)?passwd_ (Applies confidential-field status to all field names that begin with the string passwd_, or that contain the string -passwd_ after another string that might contain non-ASCII special characters.)

Following are some regular expressions that define specific URL types that you might find useful. Substitute your own web host(s) and domain(s) for those in the examples.

- If the web form appears on multiple web pages on the web host www.example.com, but all of those web pages are named logon.pl?, you could use the following regular expression: https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)*logon[.]pl\?
- If the web form appears on multiple web pages on the web host www.example-español.com, which contains the n-tilde (ñ) special character, you could use the following regular expression, which represents the n-tilde special character as an encoded UTF-8 string containing C3 B1, the hexadecimal code assigned to that character in the UTF-8 charset: https?://www[.]example-espa\xC3\xB1ol[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)* logon[.]pl\?
- If the web form containing query.pl appears on multiple web pages on different hosts within the example.com domain, you could use the following regular expression: https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])*example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?
- If the web form containing query.pl appears on multiple web pages on different hosts in different domains, you could use the following regular expression: https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])*[0-9A-Za-z][0-9A-Za-z_-.]+[.][a-z]{2,6}/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?
- If the web form appears on multiple web pages on the web host www.example.com, but all of those web pages are named logon.pl?, you could use the following regular expression: https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Field Types

Jun 12, 2014

A field type is a PCRE-format regular expression that defines a particular data format and minimum/maximum data lengths for a form field in a web form. Field types are used in the Field Formats check.

The application firewall comes with several default field types, which are:

- integer. A string of any length consisting of numbers only, without a decimal point, and with an optional preceding minus sign (-).
- alpha. A string of any length consisting of letters only.
- alphanum. A string of any length consisting of letters and/or numbers.
- nohtml. A string of any length consisting of characters, including punctuation and spaces, that does not contain HTML symbols or queries.
- any. Anything at all.
  Important: Assigning the any field type as the default field type, or to a field, allows active scripts, SQL commands, and other possibly dangerous content to be sent to your protected web sites and applications in that form field. You should use the any type sparingly, if you use it at all.

You can also add your own field types to the Field Types list. For example, you might want to add a field type for a social security number, postal code, or phone number in your country. You might also want to add a field type for a customer identification number or store credit card number.

To add a field type to the Field Types list, you enter the field name as a literal string or PCRE-format regular expression.

To add a field type by using the command line interface

At the command prompt, type the following commands:

- add appfw fieldType <name> <regex> <priority> [-comment "<string>"]
- save ns config

Example

The following example adds a field type named SSN that matches US Social Security numbers to the Field Types list, and sets its priority to 1.

```
add appfw fieldType SSN "^[1-9][0-9]{2,2}-[0-9}{2,2}-[0-9]{4,4}$" 1
save ns config
```

To modify a field type by using the command line interface

At the command prompt, type the following commands:

- set appfw fieldType <name> <regex> <priority> [-comment "<string>"]
- save ns config

Example

The following example modifies the field type to add a comment.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
set appfw fieldType SSN "^[1-9][0-9]{2,2}-[0-9}{2,2}-[0-9]{4,4}$" 1 -comment "US Social Security Number"
save ns config
```

To remove a field type by using the command line interface

At the command prompt, type the following commands:

- rm appfw fieldType <name>
- save ns config

To configure a field type by using the configuration utility

1. Navigate to Security > Application Firewall.
2. In the details pane, under Settings, click Manage Field Types.
3. In the Manage Field Types dialog box, do one of the following:
   - To add a new field type to the list, click Add.
   - To change an existing field type, select the field type, and then click Edit.

   The Configure Field Type dialog box appears.

   Note: If you select an existing field type designation and then click Add, the dialog box displays the information for that field type. You can modify that information to create your new field type.
4. In the dialog box, fill out the elements. They are:
   - Name
   - Regular Expression
   - Priority
   - Comment
5. Click Create or OK.
6. To remove a field type from the Field Types list, select the field type listing you want to remove, then click Remove to remove it, and then click OK to confirm your choice.
7. When you have finished adding, modifying, and removing field types, click Close.

Examples

Following are some regular expressions for field types that you might find useful:

- ^[1-9][0-9]{2,2}-[0-9}{2,2}-[0-9]{4,4}$ U.S. Social Security numbers
- ^[A-C][0-9]{7,7}$ California driver's license numbers.
- ^+[0-9]{1,3} [0-9() -]{1,40}$ International phone numbers with country codes.
- ^[0-9]{5,5}-[0-9]{4,4}$ U.S. ZIP code numbers.
- ^[0-9A-Za-z][0-9A-Za-z.+_-]{0,25}@([0-9A-Za-z][0-9A-Za-z_-]*[.]){1,4}[A-Za-z]{2,6}$ Email addresses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

XML Content Types

Jun 12, 2014

By default, the application firewall treats files that follow certain naming conventions as XML. You can configure the application firewall to examine web content for additional strings or patterns that indicate that those files are XML files. This can ensure that the application firewall recognizes all XML content on your site, even if certain XML content does not follow normal XML naming conventions, ensuring that XML content is subjected to XML security checks.

To configure the XML content types, you add the appropriate patterns to the XML Content Types list. You can enter a content type as a string, or you can enter a PCRE-compatible regular expression specifying one or more strings. You can also modify the existing XML content types patterns.

To add an XML content type pattern by using the command line interface

At the command prompt, type the following commands:

- add appfw XMLContentType <XMLContenttypevalue> [-isRegex (REGEX | NOTREGEX)]
- save ns config

Example

The following example adds the pattern .*/xml to the XML Content Types list and designates it as a regular expression.

```
add appfw XMLContentType ".*/xml" -isRegex REGEX
```

To remove an XML content type pattern by using the command line interface

At the command prompt, type the following commands:

- rm appfw XMLContentType <XMLContenttypevalue>
- save ns config

To configure the XML content type list by using the configuration utility

1. Navigate to Security > Application Firewall.
2. In the details pane, under Settings, click Manage XML Content Types.
3. In the Manage XML Content Types dialog box, do one of the following:
   - To add a new XML content type, click Add.
   - To modify an existing XML content type, select that type and then click Edit.

   The Configure Application Firewall XML Content Type dialog appears.

   Note: If you select an existing XML content type pattern and then click Add, the dialog box displays the information for that XML content type pattern. You can modify that information to create your new XML content type pattern.
4. In the dialog box, fill out the elements. They are:
   - **IsRegex.** Select or clear to enable PCRE-format regular expressions in the form field name.
   - **XML Content Type** Enter a literal string or PCRE-format regular expression that matches the XML content type pattern that you want to add.
5. Click Create.
6. To remove an XML content type pattern from the list, select it, then click Remove to remove it, and then click OK to confirm your choice.
7. When you have finished adding and removing XML content type patterns, click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

JSON Content Types

Jun 12, 2014

By default, the application firewall treats files with the content type "application/json" as JSON files.The default setting enables the application firewall to recognize JSON content in requests and responses, and to handle that content appropriately.

You can configure the application firewall to examine web content for additional strings or patterns that indicate that those files are JSON files. This can ensure that the application firewall recognizes all JSON content on your site, even if certain JSON content does not follow normal JSON naming conventions, ensuring that JSON content is subjected to JSON security checks.

To configure the JSON content types, you add the appropriate patterns to the JSON Content Types list. You can enter a content type as a string, or you can enter a PCRE-compatible regular expression specifying one or more strings. You can also modify the existing JSON content types patterns.

To add a JSON content type pattern by using the command line interface

At the command prompt, type the following commands:

- add appfw JSONContentType <JSONContenttypevalue> [-isRegex (REGEX | NOTREGEX)]
- save ns config

Example

The following example adds the pattern .*/json to the JSON Content Types list and designates it as a regular expression.

```
add appfw JSONContentType ".*/json" -isRegex REGEX
```

To configure the JSON content type list by using the configuration utility

1. Navigate to Security > Application Firewall.
2. In the details pane, under Settings, click Manage JSON Content Types.
3. In the Manage JSON Content Types dialog box, do one of the following:
   - To add a new JSON content type, click Add.
   - To modify an existing JSON content type, select that type and then click Edit.

   The Configure Application Firewall JSON Content Type dialog appears.

   Note: If you select an existing JSON content type pattern and then click Add, the dialog box displays the information for that JSON content type pattern. You can modify that information to create your new JSON content type pattern.
4. In the dialog box, fill out the elements. They are:
   - **IsRegex.** Select or clear to enable PCRE-format regular expressions in the form field name.
   - **JSON Content Type** Enter a literal string or PCRE-format regular expression that matches the JSON content type pattern that you want to add.
5. Click Create or OK.
6. To remove a JSON content type pattern from the list, select it, then click Remove to remove it, and then click OK to confirm your choice.
7. When you have finished adding and removing XML content type patterns, click Close.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Statistics and Reports

Sep 03, 2013

The information maintained in the logs and statistics, and displayed in the reports, provides important guidance for configuring and maintaining the application firewall.

The Application Firewall Statistics

When you enable the statistics action for application firewall signatures or security checks, the application firewall maintains information about connections that match that signature or security check. You can view the accumulated statistics information on the Monitoring tab of the main logon page of your application firewall appliance by selecting one of the following choices in the Select Group list box:

- **Application Firewall.** A summary of all statistics information gathered by your application firewall appliance for all profiles.
- **Application Firewall (per profile).** The same information, but displayed per-profile rather than summarized.

You can use this information to monitor how your application firewall is operating and determine whether there is any abnormal activity or abnormal amounts of hits on a signature or security check. If you see such a pattern of abnormal activity, you can check the logs for that signature or security check, to diagnose the issue, and then take corrective action.

The Application Firewall Reports

The application firewall reports provide information about your application firewall configuration and how it is handling traffic for your protected web sites.

The PCI DSS Report

The Payment Card Industry (PCI) Data Security Standard (DSS), version 1.2, consists of twelve security criteria that most credit card companies require businesses who accept online payments via credit and debit cards to meet. These criteria are designed to prevent identity theft, hacking, and other types of fraud. If an internet service provider or online merchant does not meet the PCI DSS criteria, that ISP or merchant risks losing authorization to accept credit card payments through its web site.

ISPs and online merchants prove that they are in compliance with PCI DSS by having an audit conducted by a PCI DSS Qualified Security Assessor (QSA) Company. The PCI DSS report is designed to assist them both before and during the audit. Before the audit, it shows which application firewall settings are relevant to PCI DSS, how they should be configured, and (most important) whether your current application firewall configuration meets the standard. During the audit, the report can be used to demonstrate compliance with relevant PCI DSS criteria.

The PCI DSS report consists of a list of those criteria that are relevant to your application firewall configuration. Under each criterion, it lists your current configuration options, indicates whether your current configuration complies with the PCI DSS criterion, and explains how to configure the application firewall so that your protected web site(s) will be in compliance with that criterion.

The PCI DSS report is located under System > Reports. To generate the report as an Adobe PDF file, click Generate PCI DSS Report. Depending on your browser settings, the report is displayed in the pop-up window or you are prompted to save it to your hard disk.

Note: To view this and other reports, you must have the Adobe Reader program installed on your computer.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The PCI DSS report consists of the following sections:

- **Description.** A description of the PCI DSS Compliance Summary report.
- **Firewall License and Feature Status.** Tells you whether the application firewall is licensed and enabled on your NetScaler appliance.
- **Executive Summary.** A table that lists the PCI DSS criteria and tells you which of those criteria are relevant to the application firewall.
- **Detailed PCI DSS Criteria Information.** For each PCI DSS criterion that is relevant to your application firewall configuration, the PCI DSS report provides a section that contains information about whether your configuration is currently in compliance and, if it is not, how to bring it into compliance.
- **Configuration.** Data for individual profiles, which you access either by clicking Application Firewall Configuration at the top of the report, or directly from the Reports pane. The Application Firewall Configuration report is the same as the PCI DSS report, with the PCI DSS-specific summary omitted, and is described below.

The Application Firewall Configuration Report

The Application Firewall Configuration report is located under System > Reports. To display it, click Generate Application Firewall Configuration Report. Depending on your browser settings, the report is displayed in the pop-up window or you are prompted to save it to your hard disk.

The Application Firewall Configuration report starts with a Summary page, which consists of the following sections:

- **Application Firewall Policies.** A table that lists your current application firewall policies, showing the policy name, the content of the policy, the action (or profile) it is associated with, and global binding information.
- **Application Firewall Profiles.** A table that lists your current application firewall profiles and indicates which policy each profile is associated with. If a profile is not associated with a policy, the table displays INACTIVE in that location.

To download all report pages for all policies, at the top of the Profiles Summary page click Download All Profiles. You display the report page for each individual profile by selecting that profile in the table at the bottom of the screen. The Profile page for an individual profile shows whether each check action is enabled or disabled for each check, and the other configuration settings for the check.

To download a PDF file containing the PCI DSS report page for the current profile, click Download Current Profile at the top of the page. To return to the Profiles Summary page, click Application Firewall Profiles. To go back to the main page, click Home. You can refresh the PCI DSS report at any time by clicking Refresh in the upper right corner of the browser. You should refresh the report if you make changes to your configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Firewall Logs

May 11, 2016

The application firewall generated log messages can be quite useful for keeping track of the configurational changes, application firewall policy invocations, and security check violations.

When the log action is enabled for security checks or signatures, the resulting log messages provide information about the requests and responses that the application firewall has observed while protecting your web sites and applications. The most important information is the action taken by the application firewall when a signature or a security check violation was observed. For some security checks, the log message can provide additional useful information, such as the location and the detected pattern that triggered the violation. You can deploy security checks in non-block mode and monitor the logs to determine whether the transactions that are triggering security violations are valid transactions (false positives). If they are, you can either remove, or reconfigure the signature or security checks, deploy relaxations, or take other appropriate measures to mitigate the false positives before you enable blocking for that signature or security check. An excessive increase in the number of violation messages in logs can indicate a surge in malicious requests. This can alert you that your application might be under attack to exploit a specific vulnerability that is detected and thwarted by application firewall protections.

NetScaler (Native) Format Logs

The application firewall uses the NetScaler format logs (also called native format logs) by default. These logs have the same format as those generated by other NetScaler features. Each log contains the following fields:

- Timestamp. Date and time when the connection occurred.
- Severity. Severity level of the log.
- Module. NetScaler module that generated the log entry.
- Event Type. Type of event, such as signature violation or security check violation.
- Event ID. ID assigned to the event.
- Client IP. IP address of the user whose connection was logged.
- Transaction ID. ID assigned to the transaction that caused the log.
- Session ID. ID assigned to the user session that caused the log.
- Message. The log message. Contains information identifying the signature or security check that triggered the log entry.

You can search for any of these fields, or any combination of information from different fields. Your selection is limited only by the capabilities of the tools you use to view the logs. You can observe the application firewall log messages in the configuration utility by accessing the NetScaler syslog viewer, or you can manually connect to the NetScaler appliance and access logs from the command line interface, or you can drop into shell and tail the logs directly from the /var/log/ folder.

Example of a Native Format Log message

```
Jun 22 19:14:37 <local0.info> 10.217.31.98 06/22/2015:19:14:37 GMT ns 0-PPE-1 :
default APPFW APPFW_XSS 60 0 :  10.217.253.62 616-PPE1 y/3upt2K8ySWWld3Kavbxyni7Rw0000
pr_ffc http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
12345&drinking_pref=on&text_area=%3Cscript%3E%0D%0A&loginButton=ClickToLogin&as_sfid=
AAAAAAWEXcNQLISokNmqaYF6dvfqlChNzSMsdyO9JXOJomm2v
BwAMOqZIChv21EcgBc3rexIUcfm0vckKIsgoOeC_BArx1Ic4NLxxkWMtrJe4H7SOfkiv9NL7AG4juPIanTvVo
%3D&as_fid=feeec8758b41740eedeeb6b35b85dfd3d5def30c Cross-site script check failed for
field text_area="Bad tag: script" <blocked>
```

Common Event Format (CEF) Logs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The application firewall also supports CEF logs. CEF is an open log management standard that improves the interoperability of security-related information from different security and network devices and applications. CEF enables customers to use a common event log format so that data can easily be collected and aggregated for analysis by an enterprise management system. The log message is broken into different fields so that you can easily parse the message and write scripts to identify important information.

Analyzing the CEF Log Message

In addition to date, timestamp, client IP, log format, appliance, company, build version, module, and security check information, application firewall CEF Log messages include the following details:

- src – source IP address
- spt – source port number
- request – request URL
- act – action (e.g. blocked, transformed)
- msg – message (Message regarding the observed security check violation)
- cn1 – event ID
- cn2 – HTTP Transaction ID
- cs1 – profile name
- cs2 – PPE ID (e.g. PPE1)
- cs3 - Session ID
- cs4 – Severity (e.g. INFO, ALERT)
- cs5 – event year
- method – Method (e.g. GET/POST)

For example, consider the following CEF format log message, which was generated when a Start URL violation was triggered:

```
Jun 12 23:37:17 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0
|APPFW|APPFW_STARTURL|6|src=10.217.253.62 spt=47606 method=GET
request=http://aaron.stratum8.net/FFC/login.html msg=Disallow Illegal URL. cn1=1340
cn2=653 cs1=pr_ffc cs2=PPE1 cs3=EsdGd3VD0OaaURLcZnj05Y6DOmE0002 cs4=ALERT cs5=2015
act=blocked
```

The above message can be broken down in the following components:

Message	**Description**
June 12	Date
23:37:17	Current time
<local0.info>	
10.217.31.98	IP Address of the VIP that received the request
CEF:0	Log format

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Citrix	Company name
NetScaler	Appliance
NS11.0	Version
APPFW	Module
APPFW_STARTURL	Security Check violation
6	Severity
src=10.217.253.62	Request is received from Client IP 10.217.253.62
spt=47606	Source port number is 47606
method=GET	The request is a "GET" request
request=http://aaron.stratum8.net/FFC/login.html	The Requested URL is "http://aaron.stratum8.net/FFC/login.html"
msg=Disallow Illegal URL	The Violation Log message generated by application firewall is "Disallow Illegal URL"
cn1=1340	Event ID is 1340
cn2=653	HTTP Transaction ID is 653
cs1=pr_ffc	The request is processed by the profile named "pr_ffc"
cs2=PPE1	The request is processed by PPE1
cs3= EsdGd3VD0OaaURLcZnj05Y6DOmE0002	Application firewall session ID is "EsdGd3VD0OaaURLcZnj05Y6DOmE0002"
cs4=ALERT	ALERT is the string representation of the severity level (6)
cs5=2015	Current year is 2015
act=blocked	The action taken by application firewall is to "Block" the request

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example of a request check violation in CEF log format: request is not blocked

```
Jun 13 00:21:28 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|
APPFW_FIELDCONSISTENCY|6|src=10.217.253.62 spt=761 method=GET request=
http://aaron.stratum8.net/FFC/login.php?login_name\=abc&passwd\=
123456789234&drinking_pref\=on&text_area\=&loginButton\=ClickToLogin&as_sfid\
=AAAAAAWlahZuYoIFbjBhYMP05mJLTwEflY0a7AKGMg3jIBaKmwtK4t7M7lNxOgj7Gmd3SZc8KUj6CR6a
7W5kIWDRHN8PtK1Zc-txHkHNx1WknuG9DzTuM7t1THhluevXu9I4kp8%3D&as_fid\=feeec8758b4174
0eedeeb6b35b85dfd3d5def30c msg=Field consistency check failed for field passwd cn1=1401
cn2=707 cs1=pr_ffc cs2=PPE1 cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=ALERT cs5=2015 act=
not blocked
```

Example of a response check violation in CEF format: response is transformed

```
Jun 13 00:25:31 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|
APPFW_SAFECOMMERCE|6|src=10.217.253.62 spt=34041 method=GET request=
http://aaron.stratum8.net/FFC/CreditCardMind.html msg=Maximum number of potential credit
card numbers seen cn1=1470 cn2=708 cs1=pr_ffc cs2=PPE1
cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=ALERT cs5=2015 act=transformed
```

Example of a request side signature violation in CEF format: request is blocked

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|
APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method=GET request=
http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=Signature violation rule ID 807:
web-cgi /wwwboard/passwd.txt access  cn1=140 cn2=841 cs1=pr_ffc cs2=PPE0
cs3=OyTgjbXBqcpBFeENKDIde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=blocked
```

Logging Geolocation in the Application Firewall Violation Messages

Geolocation, which identifies the geographic location from which requests originate, can help you configure the application firewall for the optimal level of security. To bypass security implementations such as rate limiting, which rely on the IP addresses of the clients, malware or rogue computers can keep changing the source IP address in requests. Identifying the specific region from where requests are coming can help determine whether the requests are from a valid user or a device attempting to launch cyberattacks. For example, if an excessively large number of requests are received from a specific area, it is easy to determine whether they are being sent by users or a rogue machine. Geolocation analysis of the received traffic can be very useful in deflecting attacks such as denial of service (DoS) attacks.

The application firewall offers you the convenience of using the built-in NetScaler database for identifying the locations corresponding to the IP addresses from which malicious requests are originating. You can then enforce a higher level of security for requests from those locations. Citrix default syntax (PI) expressions give you the flexibility to configure location based policies that can be used in conjunction with the built-in location database to customize firewall protection, bolstering your defense against coordinated attacks launched from rogue clients in a specific region.

You can use the NetScaler built-in database, or you can use any other database. If the database does not have any location information for the particular client IP address, the CEF log shows geolocation as an Unknown geolocation.

Note: Geolocation logging uses the Common Event Format (CEF). By default, CEF logging and GeoLocationLogging are OFF. You must explicitly enable both parameters.

Example of a CEF log message showing geolocation information

```
June 8 00:21:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|
APPFW_STARTURL|6|src=10.217.253.62 geolocation=NorthAmerica.US.Arizona.Tucson.*.*
spt=18655 method=GET request=http://aaron.stratum8.net/FFC/login.html
msg=Disallow Illegal URL. cn1=77 cn2=1547 cs1=test_pr_adv cs2=PPE1
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cs3=KDynjg1pbFtfhC/nt0rBU1o/Tyg0001 cs4=ALERT cs5=2015 act=not blocked
Example of a log message showing geolocation= Unknown
June 9 23:50:53 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|
APPFW|APPFW_STARTURL|6|src=10.217.30.251 geolocation=Unknown spt=5086
method=GET request=http://aaron.stratum8.net/FFC/login.html msg=Disallow Illegal URL.
cn1=74 cn2=1576 cs1=test_pr_adv cs2=PPE2 cs3=PyR0eOEM4gf6GJiTyauiHByL88E0002
cs4=ALERT cs5=2015 act=not blocked

Using the Command Line to Configure the Log Action and other Log Parameters

To configure the log action for a security checks of a profile by using the command line

At the command prompt, type one of the following commands:

- set appfw profile <name> SecurityCheckAction ([log] | [none])
- unset appfw profile <name> SecurityCheckAction

Examples

set appfw profile pr_ffc StartURLAction log

unset appfw profile pr_ffc StartURLAction

To configure CEF logging by using the command line

The CEF logging is disabled by default. At the command prompt, type one of the following commands to change or display the current setting:

- set appfw settings CEFLogging on
- unset appfw settings CEFLogging
- sh appfw settings | grep CEFLogging

To configure the logging of the credit card numbers by using the command line

At the command prompt, type one of the following commands:

- set appfw profile <name> -doSecureCreditCardLogging ([ON] | [OFF])
- unset appfw profile <name> -doSecureCreditCardLogging

To configure Geolocation logging by using the command line

1. Use the set command to enable GeoLocationLogging. You can enable the CEF logging at the same time. Use the unset command to disable geolocation logging. The show command shows the current settings of all the application firewall parameters, unless you include the grep command to show the setting for a specific parameter.
   - set appfw settings GeoLocationLogging ON [CEFLogging ON]
   - unset appfw settings GeoLocationLogging
   - sh appfw settings | grep GeoLocationLogging
2. Specify the database
   add locationfile /var/netscaler/inbuilt_db/Citrix_Netscaler_InBuilt_GeoIP_DB.csv

   or

   add locationfile <path to database file>

Customizing Application Firewall Logs

Default format (PI) expressions give you the flexibility to customize the information included in the logs. You have the

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

option to include the specific data that you want to capture in the application firewall generated log messages. For example, if you are using AAA-TM authentication along with the application firewall security checks, and would like to know the accessed URL that triggered the security check violation, the name of the user who requested the URL, the source IP address, and the source port from which the user sent the request, you can use the following commands to specify customized log messages that include all the data:

- **set audit syslogparams userDefinedAuditlog yes**
- **add audit messageaction** custom1 **-stringBuilderExpr** "HTTP.REQ.URL + \" \" + HTTP.REQ.USER.NAME + \" \" + CLIENT.IP.SRC + \":\" + CLIENT.TCP.SRCPORT"
  -**bypassSafetyCheck YES**
- **add appfw policy** appfw_pol true <your_profile> -**logAction** custom1

Configuring Syslog policy to Segregate Application Firewall Logs

The application firewall offers you an option to isolate and redirect the application firewall security log messages to a different log file. This might be desirable if the application firewall is generating a large number of logs, making it difficult to view other NetScaler log messages. You can also use this option when you are interested only in viewing the application firewall log messages and do not want to see the other log messages.

To redirect the application firewall logs to a different log file, configure a syslog action to send the application firewall logs to a different log facility. You can use this action when configuring the syslog policy, and bind it globally for use by application firewall.

Example

1. Switch to the shell and use an editor such as vi to edit the /etc/syslog.conf file. Add a new entry to use local2.* to send logs to a separate file as shown in the following example:
   local2.* /var/log/ns.log.appfw

2. Restart the syslog process. You can use the grep command to identify the syslog process ID (PID), as shown in the following example:
   root@ns# **ps -A | grep syslog**

   1063 ?? Ss 0:03.00 /usr/sbin/syslogd -b 127.0.0.1 -n -v -v -8 -C

   root@ns# **kill -HUP** 1063

3. From the command line interface, configure the syslog action and policy. Bind it as a global application firewall policy.
   add audit syslogAction sysact1 127.0.0.1 -logLevel ALL -logFacility LOCAL2

   add audit syslogPolicy syspol1 ns_true sysact1

   bind appfw global syspol1 1

4. All application firewall security check violations will now be redirected to the /var/log/ns.log.appfw file. You can tail this file to view the application firewall violations that are getting triggered during the processing of the ongoing traffic.
   root@ns# tail -f ns.log.appfw

Warning: If you have configured the syslog policy to redirect the logs to a different log facility, the application firewall log messages no longer appear in the /var/log/ns.log file.

Viewing the Application Firewall Logs

You can view the logs by using the syslog viewer, or by logging onto the NetScaler appliance, opening a UNIX shell, and using

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

the UNIX text editor of your choice.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the application firewall security check violations:

- Shell
- tail -f /var/log/ns.log

You can use the vi editor, or any Unix text editor or text search tool, to view and filter the logs for specific entries. For example, you can use grep command to access the log messages pertaining to the Credit Card violations:

- tail -f /var/log/ns.log | grep SAFECOMMERCE

To access the log messages by using the configuration utility

The Citrix configuration utility includes a very useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- To view log messages for a specific security check of a profile, navigate to **Application Firewall** > **Profiles**, select the target profile, and click Security Checks. Highlight the row for the target security check and click Logs. When you access the logs directly from the selected security check of the profile, it filters out the log messages and displays only the logs pertaining to the violations for the selected security check. Syslog viewer can display application firewall logs in the Native format as well as the CEF format. However, in order for the syslog viewer to filter out the target profile specific log messages, the logs must be in the CEF log format when accessed from the profile.
- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click Syslog messages link to display the Syslog Viewer, which displays all log messages, including all application firewall security check violation logs for all profiles. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the Audit Messages section, click Syslog messages link to display the Syslog Viewer, which displays all log messages, including all security check violation logs for all profiles.

The HTML based Syslog Viewer provides the following filter options for selecting only the log messages that are of interest to you:

- **File**—The current /var/log/ns.log file is selected by default, and the corresponding messages appear in the Syslog Viewer. A list of other log files in the /var/log directory are available in a compressed .gz format. To download and un-compress an archived log file, just select the log file from the dropdown option. The log messages pertaining to the selected file are then displayed in the syslog viewer. To refresh the display, click the Refresh icon (a circle of two arrows).
- **Module list box**—You can select the NetScaler module whose logs you want to view. You can set it to APPFW for application firewall logs.
- **Event Type list box**—This box contains a set of check boxes for selecting the type of event you are interested in. For example, to view the log messages pertaining to the signature violations, you can select the **APPFW_SIGNATURE_MATCH** check box. Similarly, you can select a check box to enable the specific security check that is of interest to you. You can select multiple options.
- **Severity**—You can select a specific severity level to show just the logs for that severity level. Leave all the check boxes blank if you want to see all logs.
  To access the application firewall security check violation log messages for a specific security check, filter by selecting **APPFW** in the dropdown options for Module. The Event Type displays a rich set of options to further refine your selection. For example, if you select the **APPFW_FIELDFORMAT** check box and click the Apply button, only log messages pertaining to the Field Formats security check violations appear in the Syslog Viewer. Similarly, if you select the **APPFW_SQL** and **APPFW_STARTURL** check boxes and click the **Apply** button, only log messages pertaining to these

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

two security check violations will appear in the syslog viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **EventType**, **EventID**, **ClientIP**, **TransactionID**, and so on appear below the log message. You can select any of these options to highlight the corresponding information in the logs.

Click to Deploy: This functionality is available only in the configuration utility. You can use the Syslog Viewer to not only view the logs but also to deploy relaxation rules based on the log messages for the application firewall security check violations. The log messages must be in CEF log format for this operation. If the relaxation rule can be deployed for a log message, a check box appears at the right edge of the Syslog Viewer box in the row. Select the check box, and then select an option from the Action list to deploy the relaxation rule. **Edit & Deploy**, **Deploy**, and **Deploy All** are available as Action options. For example, you can select an individual log message to edit and deploy. You can also select the check boxes for multiple log messages from one or more security checks and use the Deploy or Deploy All option. Click to Deploy functionality is currently supported for the following security checks:

- StartURL
- URL Buffer overflow
- SQL Injection
- XSS
- Field consistency
- Cookie consistency

To use Click to Deploy functionality in the configuration utility

1. In the Syslog Viewer, select APPFW in the Module options.
2. Select the security check for which to filter corresponding log messages.
3. Enable the check box to select the rule.
4. Use the Action drop-down list of options to deploy the relaxation rule.
5. Verify that the rule appears in the corresponding relaxation rule section.

Note: SQL Injection and XSS rules that are deployed by using Click to Deploy option do not include the fine grain relaxation recommendations.

Highlights

- **CEF Log Format support**—The CEF log format option provides a convenient option to monitor, parse, and analyze the application firewall log messages to identify attacks, fine tune configured settings to decrease false positives, and gather statistics.
- **Click to Deploy**—The Syslog viewer provides an option to filter, evaluate, and deploy relaxation rules for single or multiple security check violations from one convenient location.
- **Option to customize log message**—You can use advanced PI expressions to customize log messages and include the data you want to see in the logs.
- **Segregate application firewall specific logs**—You have an option to filter and redirect application-firewall specific logs to a separate log file.
- **Remote Logging**—You can redirect the log messages to a remote syslog server.
- **Geolocation Logging**—You can configure the application firewall to include the geolocation of the area from where the request is received. A built-in geolocation database is available, but you have the option to use an external geolocation database. The NetScaler appliance supports both IPv4 and IPv6 static geolocation databases.
- **Information rich log message**—Following are some examples of the type of information that can be included in the logs, depending on the configuration:
  - An application firewall policy was triggered.
  - A security check violation was triggered.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- A request was considered to be malformed.
- A request or the response was blocked or not blocked.
- Request data (such as SQL or XSS special characters) or response data (such as Credit card numbers or safe object strings) was transformed.
- The number of credit cards in the response exceeded the configured limit.
- The credit card number and type.
- The log strings configured in the signature rules, and the signature ID.
- Geolocation information about the source of the request.
- Masked (X'd out) user input for protected confidential fields.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Appendices

Sep 30, 2013

The following supplemental material provides additional detail about complex or peripheral application firewall tasks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

PCRE Character Encoding Format

Mar 28, 2012

The NetScaler operating system supports direct entry of characters in the printable ASCII character set only—characters with hexadecimal codes between HEX 20 (ASCII 32) and HEX 7E (ASCII 127). To include a character with a code outside that range in your application firewall configuration, you must enter its UTF-8 hexadecimal code as a PCRE regular expression.

A number of character types require encoding using a PCRE regular expression if you include them in your application firewall configuration as a URL, form field name, or Safe Object expression. They include:

- **Upper-ASCII characters.** Characters with encodings from HEX 7F (ASCII 128) to HEX FF (ASCII 255). Depending on the character map used, these encodings can refer to control codes, ASCII characters with accents or other modifications, non-Latin alphabet characters, and symbols not included in the basic ASCII set. These characters can appear in URLs, form field names, and safe object expressions.
- **Double-Byte characters.** Characters with encodings that use two 8-byte words. Double-byte characters are used primarily for representing Chinese, Japanese, and Korean text in electronic format. These characters can appear in URLs, form field names, and safe object expressions.
- **ASCII control characters.** Non-printable characters used to send commands to a printer. All ASCII characters with hexadecimal codes less than HEX 20 (ASCII 32) fall into this category. These characters should never appear in a URL or form field name, however, and would rarely if ever appear in a safe object expression.

The NetScaler appliance does not support the entire UTF-8 character set, but only the characters found in the following eight charsets:

- **English US (ISO-8859-1).** Although the label reads, “English US,” the application firewall supports all characters in the ISO-8859-1 character set, also called the Latin-1 character set. This character set fully represents most modern western European languages and represents all but a few uncommon characters in the rest.
- **Chinese Traditional (Big5).** The application firewall supports all characters in the BIG5 character set, which includes all of the Traditional Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in Hong Kong, Macau, Taiwan, and by many people of Chinese ethnic heritage who live outside of mainland China.
- **Chinese Simplified (GB2312).** The application firewall supports all characters in the GB2312 character set, which includes all of the Simplified Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in mainland China.
- **Japanese (SJIS).** The application firewall supports all characters in the Shift-JIS (SJIS) character set, which includes most characters (ideographs) commonly used in modern Japanese.
- **Japanese (EUC-JP).** The application firewall supports all characters in the EUC-JP character set, which includes all characters (ideographs) commonly used in modern Japanese.
- **Korean (EUC-KR).** The application firewall supports all characters in the EUC-KR character set, which includes all characters (ideographs) commonly used in modern Korean.
- **Turkish (ISO-8859-9).** The application firewall supports all characters in the ISO-8859-9 character set, which includes all letters used in modern Turkish.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Unicode (UTF-8).** The application firewall supports certain additional characters in the UTF-8 character set, including those used in modern Russian.

When configuring the application firewall, you enter all non-ASCII characters as PCRE-format regular expressions using the hexadecimal code assigned to that character in the UTF-8 specification. Symbols and characters within the normal ASCII character set, which are assigned single, two-digit codes in that character set, are assigned the same codes in the UTF-8 character set. For example, the exclamation point (!), which is assigned hex code 21 in the ASCII character set, is also hex 21 in the UTF-8 character set. Symbols and characters from another supported character set have a paired set of hexadecimal codes assigned to them in the UTF-8 character set. For example, the letter a with an acute accent (á) is assigned UTF-8 code C3 A1.

The syntax you use to represent these UTF-8 codes in the application firewall configuration is "\xNN" for ASCII characters; "\xNN\xNN" for non-ASCII characters used in English, Russian, and Turkish; and "\xNN\xNN\xNN" for characters used in Chinese, Japanese, and Korean. For example, if you want to represent a ! in an application firewall regular expression as a UTF-8 character, you would type \x21. If you want to include an á, you would type \xC3\xA1.

Note: Normally you do not need to represent ASCII characters in UTF-8 format, but when those characters might confuse a web browser or an underlying operating system, you can use the character's UTF-8 representation to avoid this confusion. For example, if a URL contains a space, you might want to encode the space as \x20 to avoid confusing certain browsers and web server software.
Below are examples of URLs, form field names, and safe object expressions that contain non-ASCII characters that must be entered as PCRE-format regular expressions to be included in the application firewall configuration. Each example shows the actual URL, field name, or expression string first, followed by a PCRE-format regular expression for it.

- A URL containing extended ASCII characters.

  Actual URL: http://www.josénuñez.com

  Encoded URL: ^http://www[.]jos\xC3\xA9nu\xC3\xB1ez[.]com$

- Another URL containing extended ASCII characters.

  Actual URL: http://www.example.de/trömso.html

  Encoded URL: ^http://www[.]example[.]de/tr\xC3\xB6mso[.]html$

- A form field name containing extended ASCII characters.

  Actual Name: nome_do_usuário

  Encoded Name: ^nome_do_usu\xC3\xA1rio$

- A safe object expression containing extended ASCII characters.

  Unencoded Expression [A-Z]{3,6}¥[1-9][0-9]{6,6}

  Encoded Expression: [A-Z]{3,6}\xC2\xA5[1-9][0-9]{6,6}

You can find a number of tables that include the entire Unicode character set and matching UTF-8 encodings on the Internet. A useful web site that contains this information is located at the following URL:

http://www.utf8-chartable.de/unicode-utf8-table.pl

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

For the characters in the table on this web site to display correctly, you must have an appropriate Unicode font installed on your computer. If you do not, the visual display of the character may be in error. Even if you do not have an appropriate font installed to display a character, however, the description and the UTF-8 and UTF-16 codes on this set of web pages will be correct.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Whitehat WASC Signature Types for WAF Use

May 02, 2013

The Citrix NetScaler Application Firewall accepts and generates blocking rules for all vulnerability types that the Whitehat scanners generate. However, certain vulnerabilities are most applicable to a web application firewall. Following are lists of those vulnerabilities, categorized by whether they are addressed by WASC 1.0, WASC 2.0, or best practices signature types.

WASC 1.0 Signature Types

- HTTP Request Smuggling
- HTTP Response Splitting
- HTTP Response Smuggling
- Null Byte Injection
- Remote File Inclusion
- URL Redirector Abuse

WASC 2.0 Signature Types

- Abuse of Functionality
- Brute Force
- Content Spoofing
- Denial of Service
- Directory Indexing
- Information Leakage
- Insufficient Anti-automation
- Insufficient Authentication
- Insufficient Authorization
- Insufficient Session Expiration
- LDAP Injection
- Session Fixation

Best Practices

- Autocomplete Attribute
- Insufficient Cookie Access Control
- Insufficient Password Strength
- Invalid HTTP Method Usage
- Non-HttpOnly Session Cookie
- Persistent Session Cookie
- Personally Identifiable Information
- Secured Cachable HTTP Messages
- Unsecured Session Cookie

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Streaming Support for Request Processing

Jan 18, 2016

Note: This feature is available in NetScaler release 10.5.e.

The Citrix application firewall now uses request side streaming, which results in a significant performance boost. Instead of buffering the entire request before processing it, the application firewall now looks at the incoming data, field by field, to inspect the input of each field for any configured security check violation (SQL, XSS, Field Consistency, Field Formats, etc.). As soon as the processing of the data for a field is completed, it is forwarded to the backend while the evaluation continues for the remaining fields. This significantly improves the processing time specially when handling large posts where the forms have large number of fields.

Although the streaming process is transparent to the users, minor configuration adjustments are required due to the following changes:

RegEx Pattern Match: RegEx pattern match is now restricted to 4K for contiguous character string match.

Field Name Match: Application firewall learning engine can only distinguish the first 128 bytes of the name for learning. If a form has multiple fields with names that have identical string match for the first 128 bytes, the learning engine may not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields.

Removing white spaces, percent decoding, unicode decoding, and charset conversion which is done during canonicalization is carried out prior to security check inspection. The 128 byte limit is applicable to the canonicalized representation of the field name in UTF-8 character format. The ASCII characters are 1 byte but the UTF-8 representation of the characters in some international languages may range from 1-4 bytes. If each character in the name takes 4 bytes when converted to UTF-8 format, only first 32 characters in the name may be distinguished by the learned rule for such a language.

Field Consistency Check: When the field Consistency check is enabled, all the forms in the session are now stored based on the “as_fid” tag inserted by the application firewall without consideration for the “action_url.”

- **Mandatory Form tagging for Form Field consistency**: When the field consistency check is enabled, the form tag must be enabled also. The Field Consistency protection might not work if form tagging is turned off.
- **Sessionless Form Field Consistency**: The application firewall no longer carries out the “GET” to “POST” conversion of forms when sessionless field consistency parameter is enabled. The form tag is required for sessionless field consistency also.
- **Tampering of as_fid**: If a form is submitted after tampering as_fid, it now triggers field consistency violation even if no other field was tampered. In non-streaming requests, this was allowed because the forms could be validated using the “action_url” stored in the session.

Signatures: The signatures now have the following specifications:

- **Location**: It is now a mandatory requirement that location must be specified for each pattern. All patterns in the rule **MUST** have a <Location> tag.
- **Fast Match**: All signature rules must have a fast match pattern. If there is no fast match pattern, an attempt will be made to select one if possible. Fast match must be a literal string but some PCRE’s can be used for fast match if they contain a usable literal string.
- **Deprecated Locations**: Following locations are no longer supported in signature rules.
  - HTTP_ANY
  - HTTP_RAW_COOKIE
  - HTTP_RAW_HEADER
  - HTTP_RAW_RESP_HEADER

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- HTTP_RAW_SET_COOKIE

XSS/SQL Transform: Raw data is used for transformation because the SQL special characters (single quote('), backslash (\), and semicolon (;)), and XSS tags ((<) and(>)) are same in all languages and do not need canonicalization of data. All representations of these characters, such as HTML entity encoding, percent encoding, or ASCII are evaluated for transform operation.

The application firewall no longer inspects both the attribute name and value for the XSS transform operation. Now only XSS attribute names are transformed when streaming is engaged.

Processing XSS Tags: As part of the streaming changes in 10.5.e build onwards, the application firewall processing of the Cross-site Scripting tags has changed. In earlier releases, presence of either open bracket (<), or close bracket (>), or both open and close brackets (<>) was flagged as Cross-site Scripting Violation. The behavior has changed in 10.5.e build onwards. Presence of only the open bracket character (<), or only the close bracket character (>) is no longer considered as an attack. It is when an open bracket character (<) is followed by a close bracket character (>), the Cross-site scripting attack gets flagged. Both characters must be present in the right order (< followed by >) to trigger Cross-site scripting violation.

Note

Change in SQL violation log Message: As part of the streaming changes in 10.5.e build onwards, we now process the input data in blocks. RegEx pattern matching is now restricted to 4K for contiguous character string matching. With this change, the SQL violation log messages might include different information compared to the earlier builds. The keyword and special character in the input could be separated by a large number of bytes. We now keep track of the SQL keywords and special strings when processing the data, instead of buffering the entire input value. In addition to the field name, the log message now includes the SQL keyword, or the SQL special character, or both the SQL keyword and the SQL special character, as determined by the configured setting. The rest of the input is no longer included in the log message, as shown in the following example:

Example:

In 10.5, when the application firewall detects the SQL violation, the entire input string might be included in the log message, as shown below:

SQL Keyword check failed for field **text=\"select a name from testbed1\;\\(\;\\)\".*\<blocked\>**

In 11.0, we log only the field name, keyword and special character (if applicable) in the log message, as shown below:

SQL Keyword check failed for field **text="select(;)" <blocked>**

This change is applicable to requests that contain **application/x-www-form-urlencoded**, or **multipart/form-data**, or **text/x-gwt-rpc** content-types. Log messages generated during processing of **JSON** or **XML** payloads are not affected by this change.

RAW POST Body: The security check inspections are always done on RAW POST body.

Form ID: The application firewall inserted “as_fid” tag, which is a computed hash of the form, will no longer be unique for the user session. It will now have an identical value for a specific form irrespective of the user or the session.

Charset: If a request does not have a charset, the default charset specified in the application profile is used when processing the request.

Counters:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Counters with prefix “se_” and “appfwreq_” are added to track the streaming engine and the application firewall streaming engine request counters respectively.

nsconsmg -d statswt0 -g se_err_

nsconsmg -d statswt0 -g se_tot_

nsconsmg -d statswt0 -g se_cur_

nsconsmg -d statswt0 -g appfwreq_err_

nsconsmg -d statswt0 -g appfwreq_tot_

nsconsmg -d statswt0 -g appfwreq_cur_

_err counters: indicate the rare event which should have succeeded but failed due to either memory allocation problem or some other resource crunch.

_tot counters: ever increasing counters.

_cur counters: counters indicating current values that keep changing based on usage from current transactions.

Tips:

- The application firewall security checks should work exactly the same as before.
- There is no set ordering for the processing of the security checks.
- The response side processing is not affected and remains unchanged.
- Streaming is not engaged if CVPN is used.

Important

Calculating the Cookie length: In release 10.5.e (in a few interim enhancement builds prior to 59.13xx.e build) as well as in the 11.0 release (in builds prior to 65.x), application firewall processing of the Cookie header was changed. In those releases, every cookie is evaluated individually, and if the length of any one cookie received in the Cookie header exceeds the configured BufferOverflowMaxCookieLength, the Buffer Overflow violation is triggered. As a result of this change, requests that were blocked in 10.5 and earlier release builds might be allowed, because the length of the entire cookie header is not calculated for determining the cookie length. In some situations, the total cookie size forwarded to the server might be larger than the accepted value, and the server might respond with "400 Bad Request".

Note that this change has been reverted. The behavior in the 10.5.e ->59.13xx.e and subsequent 10.5.e enhancement builds as well as in the 11.0 release 65.x and subsequent builds is now similar to that of the non-enhancement builds of release 10.5. The entire raw Cookie header is now considered when calculating the length of the cookie. Surrounding spaces and the semicolon (;) characters separating the name-value pairs are also included in determining the cookie length.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Trace HTML Requests with Security Logs

Jan 16, 2015

Note: This feature is available in NetScaler release 10.5.e.

Troubleshooting a problem, which requires analysis of data received in the client request can be quite challenging specially when there is heavy traffic flowing through the box. Diagnosing issues which may affect the functionality or security of the application require a quick response.

The NetScaler now offers the option to isolate traffic for a specific application firewall profile and collect nstrace for the HTML requests that trigger a log or block action or malformed requests that might be causing reset or aborts. The nstrace collected in –appfw mode will include details of the entire request including the application firewall generated log messages. You can use "Follow TCP stream" in the trace to view the details of the individual transaction including headers, payload, as well as the corresponding log message, together in the same screen.

This gives you a comprehensive overview regarding your traffic. Having a detailed view of the request, payload, and associated log records can be very useful to analyze security check violation. You can easily identify the pattern that is triggering the violation. If the pattern should be allowed, you can take a decision to modify the configuration and/or add a relaxation rule.

This enhancement helps in troubleshooting the NetScaler ADC and offers the following benefits:

1. **Isolate traffic for specific profile**: This enhancement can be quite useful when you need to isolate traffic for only one profile or specific transactions of a profile for troubleshooting. You no longer have to skim through the entire data collected in the trace or need special filters to isolate requests that are of interest to you which can be tedious especially with heavy traffic. You now have the option to view only the data that you are interested in.
2. **Collect data for specific requests**: The trace can be collected for a specified duration. You can collect trace for only a couple of requests to isolate, analyze, and debug specific transactions if needed.
3. **Identify resets or aborts**: Unexpected closing of connections are not easily visible. The trace collected in –appfw mode captures a reset or an abort, triggered by the application firewall. This allows a quicker isolation of issue when you do not see a security check violation message. Malformed requests or other non-RFC compliant requests terminated by application firewall will now be easier to identify.
4. **View decrypted SSL traffic**: HTTPS traffic is captured in plain text to allow for easier troubleshooting.
5. **Provides comprehensive view**: Allows you to look at the entire request at the packet level, check the payload, look at the logs to check what security check violation is being triggered and identify the match pattern in the payload. If the payload consists of any unexpected data, junk strings, or non-printable characters (null character, \r or \n etc), they are easy to discover in the trace.
6. **Modify configuration**: The debugging can provide useful information to decide if the observed behavior is the correct behavior or the configuration should be modified.
7. **Expedite response time**: Faster debugging on target traffic can improve the response time to provide explanations and/or root cause analysis by Citrix engineering and support team.

Please see any task topic in eDocs for documenting tasks. http://support.citrix.com/proddocs/topic/ns-security-10-5-map/appfw-config-manual-cli-tsk.html

To configure debug tracing for a profile by using the command line interface

Step 1. Enable tracing for the profile. You can use the show command to verify the configured setting.

- set appfw profile <profile> -trace ON

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Step 2. Start collecting trace. You can continue to use all the options which are applicable for the nstrace command.

- start nstrace -mode APPFW

Stop collecting the trace

- stop nstrace

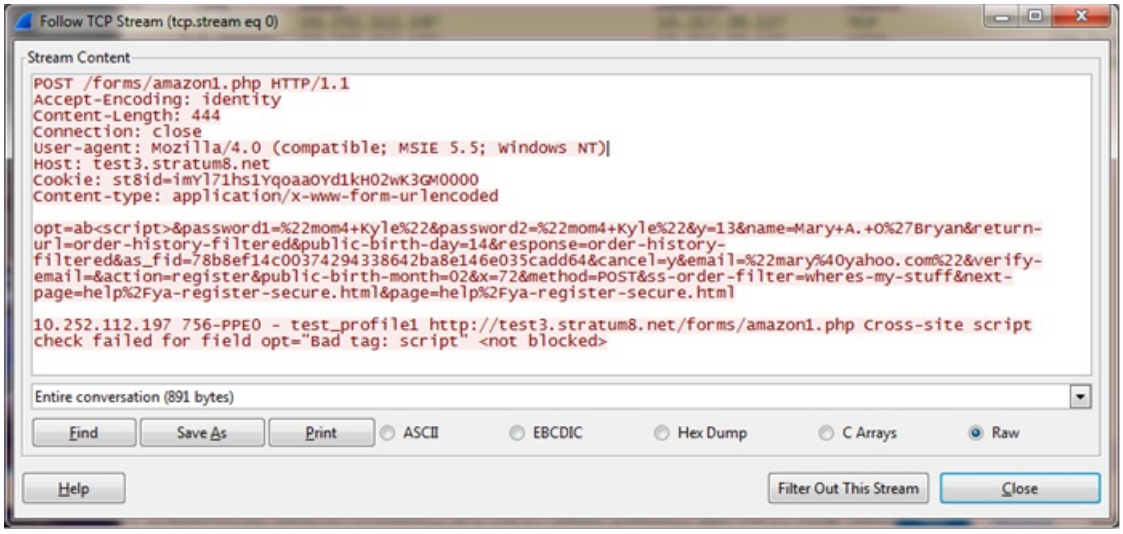
Location of the trace: The nstrace is stored in a time-stamped folder which is created in the /var/nstrace directory and can be viewed using wireshark. You can tail the /var/log/ns.log to see the log messages providing details regarding the location of the new trace.

Tips:

- When –appfw mode option is used, the nstrace will only collect the data for the profile(s) for which “trace” was enabled.
- Enabling trace on the profile will not automatically start collecting the traces till you explicitly execute the “start nstrace” command to collect the trace.
- Although, enabling trace on a profile may not have any adverse effect on the performance of the application firewall but you may want to enable this feature only for the duration for which you want to collect the data. It is recommended that you turn the –trace flag off after you have collected the trace. This will prevent the risk of inadvertently getting data from profiles for which you had enabled this flag in the past.
- The Block or Log action must be enabled for the security check for the transaction record to be included in the nstrace.
- Resets and aborts will be logged independently of security checks actions when trace is “On” for the profile(s).
- This feature is only applicable for troubleshooting the requests received from the client. The traces in –appfw mode do not include the responses received from the server.
- You can continue to use all the options which are applicable for the nstrace command. For example, "start nstrace -tcpdump enabled -size 0 -mode appFW"

- If a request triggers multiple violations, the nstrace for that record will include all the corresponding log messages.
- CEF log message format is supported for this functionality.
- Signature violations triggering block and/or log action for request side checks will also be included in the trace.
- Only HTML (non-XML) requests are collected in the trace.

Example of a Log record in the trace:



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Firewall Support for Cluster Configurations

Jan 28, 2011

Note: Application firewall support for Striped and partially striped configurations was introduced in NetScaler release 11.0.

A cluster is a group of NetScaler appliances that are configured and managed as a single system. Each appliance in the cluster is called a node. Depending on the number of nodes the configurations are active on, cluster configurations are referred to as striped, partially striped, or spotted configurations. The application firewall is fully supported in all configurations.

The two main advantages of striped and partially striped virtual server support in cluster configurations are the following:

1. Session failover support—Striped and partially striped virtual server configurations support session failover. The advanced application firewall security features, such as Start URL Closure and the Form Field Consistency check, maintain and use sessions during transaction processing. In ordinary high availability configurations, or in spotted cluster configurations, when the node that is processing the application firewall traffic fails, all the session information is lost and the user has to reestablish the session. In striped virtual server configurations, user sessions are replicated across multiple nodes. If a node goes down, a node running the replica becomes the owner. Session information is maintained without any visible impact to the user.
2. Scalability—Any node in the cluster can process the traffic. Multiple nodes of the cluster can process the incoming requests served by the striped virtual server. This improves the application firewall's ability to handle multiple simultaneous requests, thereby improving the overall performance.

Security checks and signature protections can be deployed without the need for any additional cluster-specific application firewall configuration. You just do the usual application firewall configuration on the configuration coordinator (CCO) node for propagation to all the nodes.

Cluster details are available at http://support.citrix.com/proddocs/topic/ns-system-11-map/ns-cluster-home-con.html.

Note: The session information is replicated across multiple nodes, but not across all the nodes in the striped configuration. Therefore, failover support accommodates a limited number of simultaneous failures. If multiple nodes fail simultaneously, the application firewall might lose the session information if a failure occurs before the session is replicated on another node.

Highlights

- Application firewall offers scalability, high throughput, and session failover support in cluster deployments.
- All application firewall security checks and signature protections are supported in all cluster configurations.
- Character-Maps are not yet supported for a cluster. The learning engine recommends Field-Types in learned rules for the Field Format security check.
- Stats and learned rules are aggregated from all the nodes in a cluster.
- Distributed Hash Table (DHT) provides the caching of the session and offers the ability to replicate session information across multiple nodes. When a request comes to the virtual server, the NetScaler appliance creates application firewall sessions in the DHT, and can also retrieve the session information from the DHT.
- Clustering is licensed with the Enterprise and Platinum licenses. This feature is not available with the Standard license.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Reputation

Mar 21, 2016

Citrix offers reputation based security. Using reputation assessment to determine the risk of processing requests, you can take actions such as blocking or dropping certain requests to improve the performance of your application.

The NetScaler IP reputation feature uses IP reputation checks to prevent Zero day attacks and provide protection against malicious sources associated with web attacks, phishing activity, or web scanning.

For additional details, see IP Reputation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

IP Reputation

Aug 16, 2017

IP reputation is an extremely effective tool in identifying the IP address that is sending unwanted requests. You can use the IP reputation list to preemptively reject requests that are coming from the IP with the bad reputation. For example, you can use this feature to optimize application firewall performance by filtering out the requests that you do not want to process. You can reset or drop the connection, or you can configure a responder policy to take a specific responder action.

Following are some examples of attacks that you can prevent by using IP Reputation:

- **Virus Infected personal computers** (home PCs) are the single biggest source of Spam on the internet. IP Reputation can identify the IP address that is sending unwanted requests. IP reputation can be especially useful for blocking large scale DDoS, DoS, or anomalous syn flood attacks from known infected sources.
- **Centrally managed and automated botnet** attacks have gained popularity for stealing passwords, because it doesn't take long when hundreds of computers work together to crack your password. It is easy to launch botnet attacks to figure out passwords that use commonly used dictionary words.
- **Compromised web-server** attacks are not as common as they used to be, because awareness and server security have increased, so hackers and spammers look for easier targets. There are still web servers and online forms that hackers can compromise and use to send spam (often the more vicious types, such as viruses and porn) but usually this type of activity is easier to detect and and quickly shut down, or block with a reputation list such as SpamRats.
- **Windows Exploits** (such as Active IPs offering/distributing malware, shell code, rootkits, worms or viruses).
- **Known spammers and hackers**.
- **Mass e-mail marketing campaigns**.
- **Phishing Proxies** (IP addresses hosting phishing sites, and other fraud such as ad click fraud or gaming fraud).
- **Anonymous proxies** (IPs providing proxy and anonymization services including The Onion Router aka TOR).

A NetScaler appliance uses **Webroot** as the service provider for the dynamically generated malicious IP database and the metadata for those IP addresses. Metadata might include geolocation details, threat category, threat count, and so on. The Webroot threat Intelligence engine receives real-time data from millions of sensors. It automatically and continuously captures, scans, analyses and scores the data, using advanced machine learning and behavioral analysis. Intelligence about a threat is continually updated.

As soon as a threat is detected anywhere in the network, the IP address is flagged as malicious and all appliances connected to the network are immediately protected. The dynamic changes in the IP addresses are processed with high speed and accuracy by leveraging advanced machine learning.

As stated in the datasheet from Webroot, the Webroot's sensor network identifies many key IP threat types, including spam sources, Windows exploits, botnets, scanners, and others. (See the flow diagram on the datasheet)

The NetScaler appliance uses an iprep client process to get the database from Webroot. The iprep client uses the HTTP GET method to get the absolute IP list from Webroot for the first time. Subsequently, it checks delta changes once every 5 minutes.

> ## Important
>
> Make sure the NetScaler appliance (NSIP) has Internet access and DNS is configured before you use the IP Reputation feature.
>
> To access the webroot database, the NetScaler appliance should be able to connect to **api.bcti.brightcloud.com** on **port 443**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Each node in the High Availability (HA) or Cluster deployment gets the database directly from the webroot and should be able to access this FQDN (Fully Qualified Domain Name).

PI expressions: The IP Reputation feature can be configured by using PI expressions (NetScaler default syntax expressions) in the policies bound to supported modules such as application firewall and responder. Following are two examples showing expressions that can be used to detect whether the client IP address is malicious

1. **CLIENT.IP.SRC.IPREP_IS_MALICIOUS**: This expression evaluates to TRUE if the client is included in the malicious IP list.
2. **CLIENT.IP.SRC.IPREP_THREAT_CATEGORY (CATEGORY)**: This expression evaluates to TRUE if the client IP is malicious IP and is in the specified threat category.

Following are the possible values for the threat category.

SPAM_SOURCES: Includes tunneling spam messages through a proxy, anomalous SMTP activities, and forum spam activities.

WINDOWS_EXPLOITS: includes active IP addresses offering or distributing malware, shell code, rootkits, or viruses

WEB_ATTACKS: includes cross site scripting, iFrame injection, SQL injection, cross domain injection, or domain password brute force attack.

BOTNETS: includes Botnet C&C channels, and infected machines controlled by Bot master.

SCANNERS: includes all reconnaissance such as probes, host scan, domain scan, and password brute force attack.

DOS: Deniel of Service - includes DOS, DDOS, anomalous sync flood, and anomalous traffic detection.

REPUTATION: denies access from IP addresses currently known to be infected with malware. This category also includes IPs with average low Webroot Reputation Index score. Enabling this category will prevent access from sources identified to contact malware distribution points.

PHISHING: includes IP addresses hosting phishing sites and other kinds of fraud activities such as ad click fraud or gaming fraud.

PROXY: includes IP addresses providing proxy services.

NETWORK: includes stealing of data, such as credit card data or passwords, utilization of the computer as a portion of a botnet attack, for spamming or creating Denial of service attacks, uploading or downloading of files, software installation, such as additional malware, keystroke logging.

CLOUD_PROVIDERS: includes data breaches. Weak identity, credential and access management. Insecure interfaces and APIs. System and application vulnerability

MOBILE_THREATS: includes application-based threats, web-based threats, network-based threats and physical threats.

Note

The IP reputation feature can check both source and destination IP addresses. It can also detect malicious IPs in the header. If the PI Expression in a policy can identify the IP address, the IP reputation check can determine whether it is malicious.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

IPRep log message. The /var/log/iprep.log file contains useful messages that capture information about communication with the Webroot database. The information can be about the credentials used during Webroot communication, failure to connect with Webroot, what is included in an update (such as number of IP addresses in the database), and so on.

Creating a blacklist or whitelist of IPs using policy data set. You can maintain a whitelist to allow access to specific IP addresses that are blacklisted in the Webroot database. You can also create a customized blacklist of IP addresses to supplement the Webroot reputation check. These lists can be created by using policy **data set**. A data set is a specialized form of pattern set that is ideally suited for IPv4 address matching. To use data sets, first create the data set and bind IPv4 addresses to it. Then, when you configure a policy for comparing a string in a packet, use an appropriate operator and pass the name of the pattern set or data set as an argument.

To use the dataset to create a customized whitelist of addresses to treat as exceptions during IP reputation evaluation, configure the policy so that the PI expression evaluates to False even if an address in the whitelist is listed as malicious by Webroot (or any service provider).

Enabling or disabling IP reputation. IP reputation is a part of the general reputation feature, which is license based. When you enable or disable the reputation feature, it enables or disables IP Reputation.

General procedure. Deploying IP reputation involves the following tasks

- Verify that the license installed on the NetScaler appliance has IP reputation support. Platinum and standalone application firewall licenses support the IP reputation feature.
- Enable the IP reputation and application firewall features.
- Add an application firewall profile.
- Add an application firewall policy using the PI expressions to identify the malicious IP addresses in the IP Reputation database.
- Bind the application firewall policy to an appropriate bind point.
- Verify that any request received from a malicious address gets logged in the ns.log file to show that the request was processed as specified in the profile.

Using the Command Line to Configure the IP Reputation Feature

To enable IP reputation using CLI, you can use the following commands:

> **enable feature** [**rep** | **reputation**]

> **disable feature** [**rep** | **reputation**]

The following examples show how you can add an application firewall policy using the PI expression to identify malicious addresses. You can use the built-in profiles, or add a new profile, or configure an existing profile to invoke the desired action when a request matches a policy match.

Examples 3 and 4 show how to create a policy dataset to generate a Black (to be blocked) or White (to be allowed) list of IP addresses.

Example 1

The following command creates a policy that identifies malicious IP addresses and block the request if a match is triggered:

> add appfw policy pol1 CLIENT.IP.SRC.IPREP_IS_MALICIOUS APPFW_BLOCK

Example 2

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The following command creates a policy that uses the reputation service to check the client IP address in a specific header (X-Forwarded-For) and reset the connection if a match is triggered:

> add appfw policy pol1 "HTTP.REQ.HEADER(\"X-Forwarded-For\").TYPECAST_IP_ADDRESS_AT.IPREP_IS_MALICIOUS" APPFW_RESET

Example 3

The following example shows how to add a list to add exceptions that allow specified IP addresses:

> add policy dataset Allow_list ipv4

> bind policy dataset Allow_list 10.217.25.17 -index 1

> bind policy dataset Allow_list 10.217.25.18 -index 2

Example 4

The following example shows how to add the customized list to flag specified IP addresses as malicious:

> add policy dataset Block_List ipv4

> bind policy dataset Block_List 10.217.31.48 -index 1

> bind policy dataset Block_List 10.217.25.19 -index 2

Example 5

The following example shows a policy expression to block the client IP if it matches an IP address configured in the customized Block_list (example 4) or if it matches an IP address listed in the Webroot database unless relaxed by inclusion in the Allow_list (example 3).

> add appfw policy "Ip_Rep_Policy" "((CLIENT.IP.SRC.IPREP_IS_MALICIOUS || CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY(\"Block_List\")) && ! (CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY(\"Allow_List\")))" APPFW_BLOCK

Using Proxy server

If the NetScaler appliance does not have direct access to the Internet and is connected to proxy, use the following command to configure the iprep client to send requests to the proxy.

> set reputation settings –proxyServer <proxy server ip> -proxyPort <proxy server port>

Example

> set reputation settings proxyServer 10.102.30.112 proxyPort 3128

> set reputation settings –proxyServer testproxy.citrite.net –proxyPort 3128

> unset reputation settings –proxyserver –proxyport

> sh reputation settings

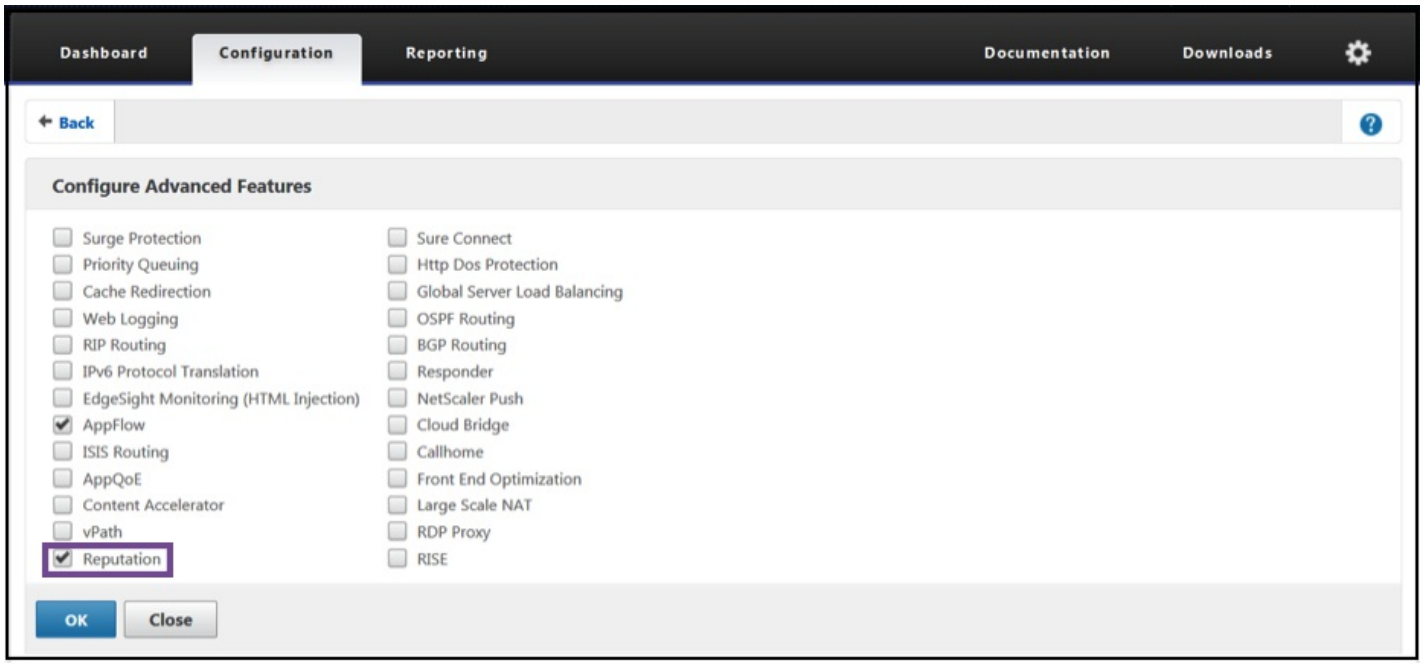
© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> The Proxy Server IP can be an IP address or a fully qualified domain name (FQDN).

Using the GUI to Configure the IP Reputation Feature

To enable reputation (which enables IP reputation) feature in GUI

Navigate to the **System** -> **Settings**. In the **Modes and Features** section, click the link to access the **Configure Advanced Features** pane and enable the **Reputation** check box. Click **OK**.



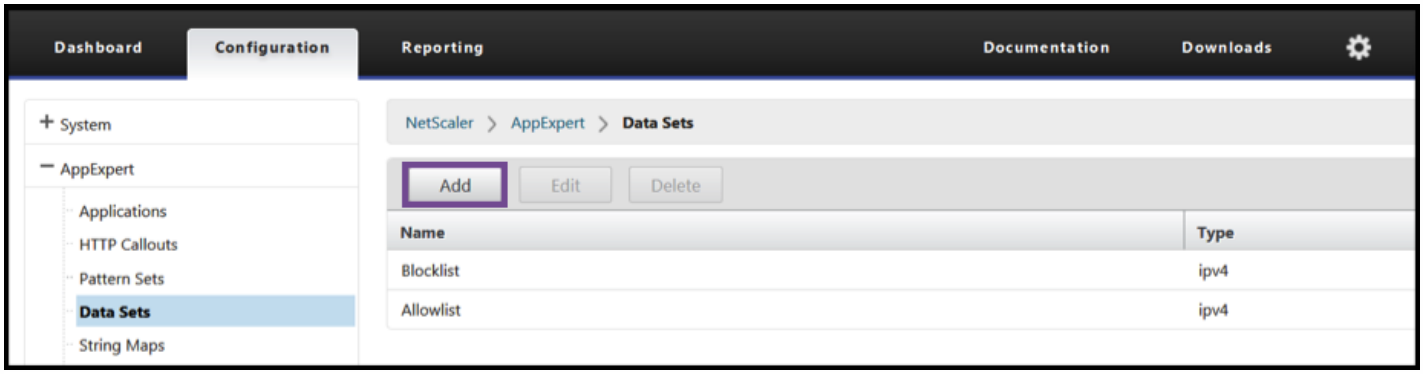
To configure a proxy server by using the configuration utility

1. On the configuration tab, navigate to **Security** > **Reputation**. Under **Settings**, click **Change Reputation Settings** to configure a proxy server. You can also enable or disable the reputation feature. **Proxy Server** can be an IP address or a fully qualified domain name (FQDN). **Proxy port** accepts values between [1 – 65535].

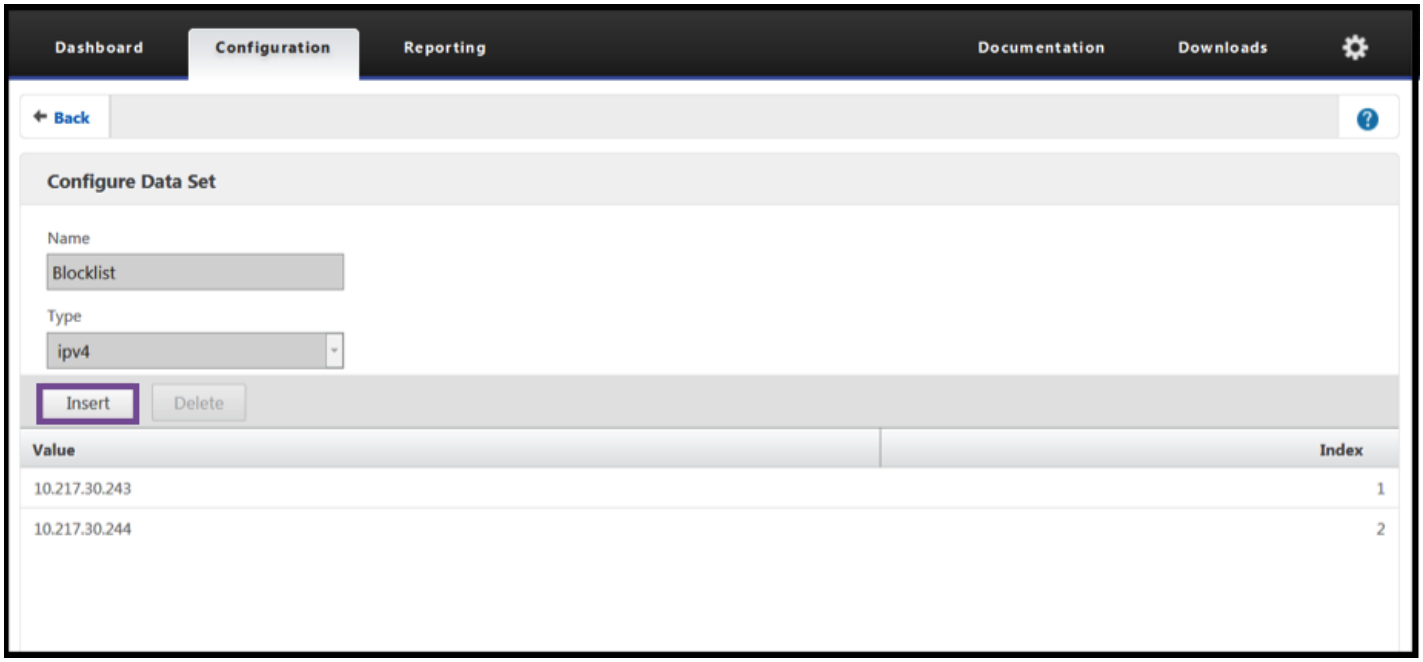
© 1999-2017 Citrix Systems, Inc. All rights reserved.

To use a dataset to create a whitelist (list of safe IPs) and blacklist (list of unsafe IPs) of Client IP addresses

- On the **Configuration** tab, navigate to **AppExpert** > **Data Sets**.
- Click **Add**.



- In the **Create Data Set** (or **Configure Data set**) pane, provide a meaningful name for the list of the IP addresses. The name should reflect the purpose of the list. For example, you could use a name such as *whitelist* or *Allow_list* when creating a list of IP addresses you would like to exempt from the action taken when the IP Reputation check determines that the source IP matches an IP address designated as a malicious IP in the webroot database. Similarly, use names such as *Block_list* or *Malicious_ip_list* if you are adding a list of IP addresses you want to flag as malicious to supplement the webroot database.
- Select ***Type** as **IPv4**.
- Click **Insert** to add a new entry.



- In the Configure Policy dataset binding pane, add an IPv4 format IP address in the *Value input box.
- Provide an index.
- Add a comment that explains the purpose of the list. This step is optional, but is recommended because a descriptive

© 1999-2017 Citrix Systems, Inc. All rights reserved.

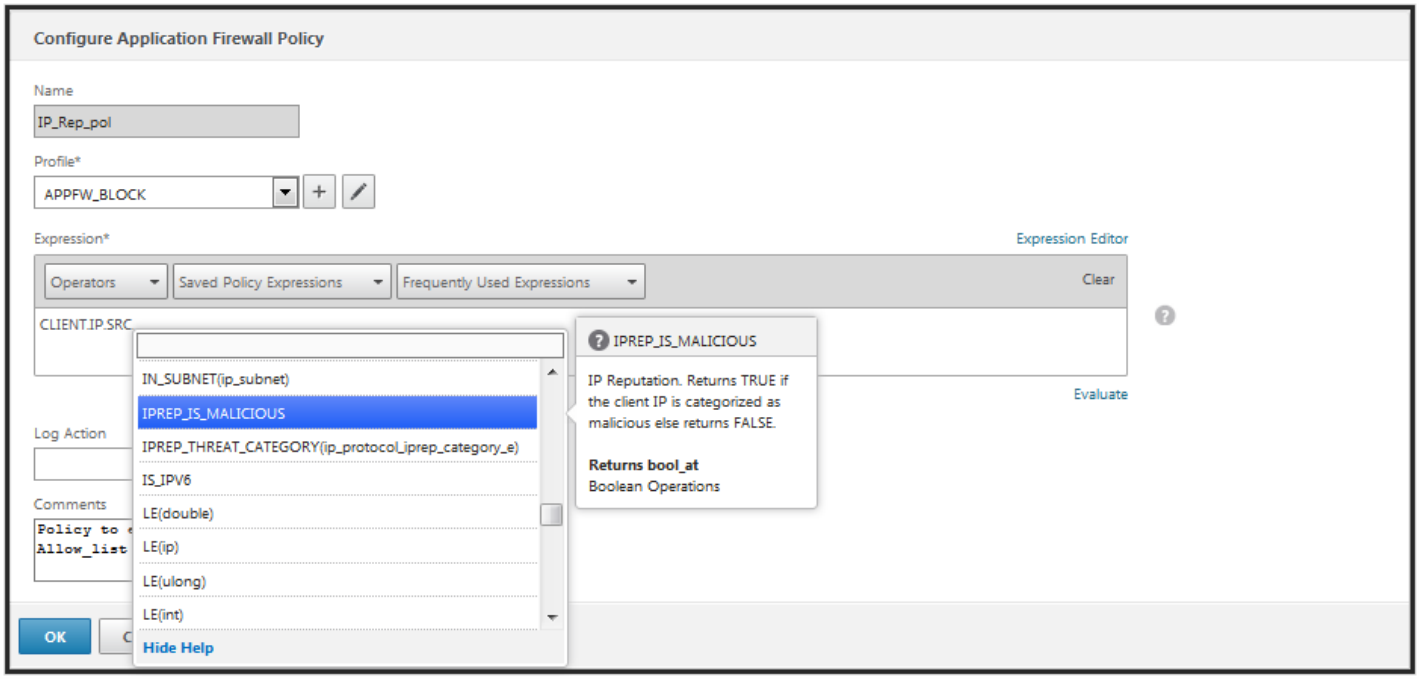
comment is quite helpful in managing the list.

Similarly, you can create a Block_list and add the IP addresses that are to be considered malicious.

Also see, http://docs.citrix.com/en-us/netscaler/11/appexpert/pattern-sets-data-seta.html for additional details regarding using data sets and configuring default syntax policy expressions.

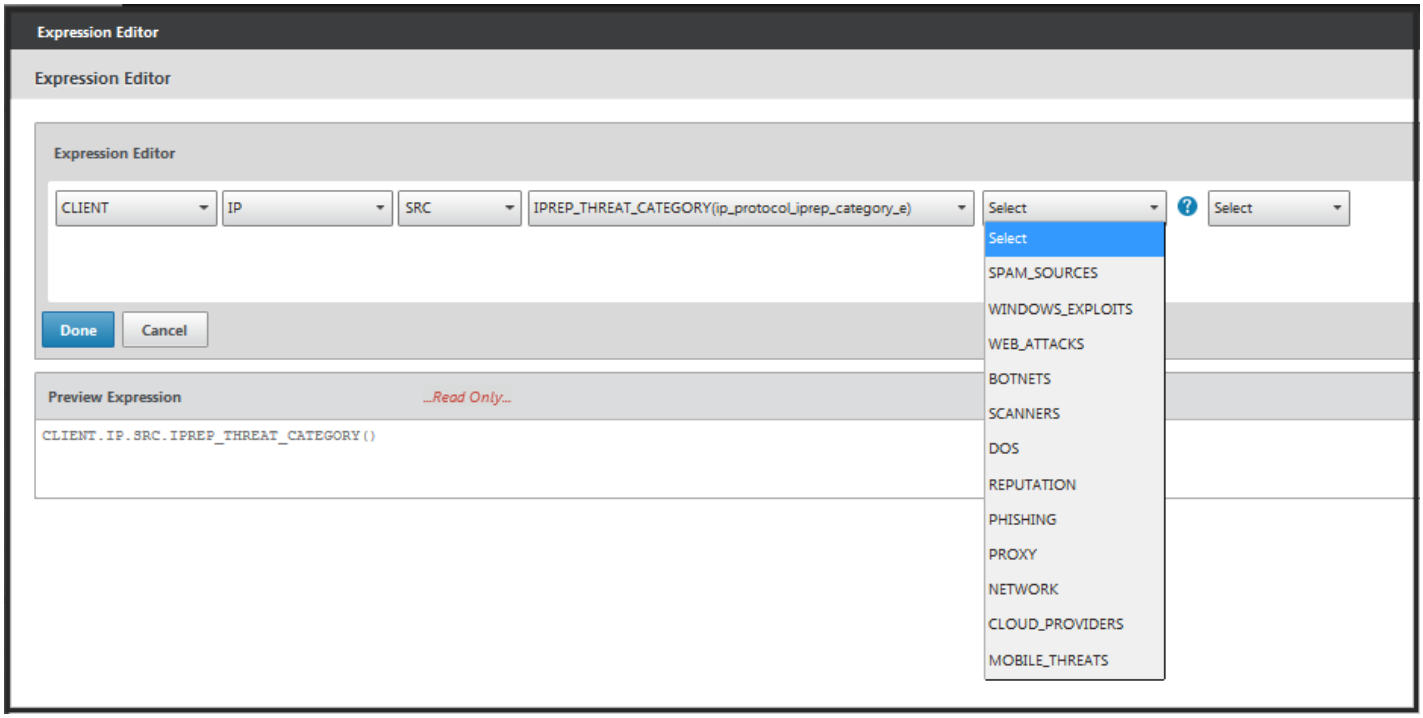
To configure an application firewall policy by using the configuration utility

1. On the **Configuration** tab, navigate to **Security** > **Application Firewall** > **Policies** > **Firewall**. Click **Add** to add a new policy using the PI expressions to use IP reputation.



You can also use the Expression editor to build your own policy expression. The drop-down menu shows preconfigured options that are quite useful for configuring an expression using the threat categories.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



Highlights:

- Quickly and accurately stop bad traffic at the network's edge from known malicious IP addresses posing different types of threats. You can block the request without parsing the body.
- Dynamically configure IP reputation functionality for multiple applications.
- Secure your network against data breach without a performance penalty, and consolidate protections onto a single services fabric using fast and easy deployments.
- You can do IP Reputation checks on source as well as destination IPs.
- You can also inspect the headers to detect malicious IPs.
- IP reputation check is supported in both forward proxy and reverse proxy deployments.
- The iprep process connects with Webroot and updates the database every 5 minutes.
- Each node in the High Availability (HA) or Cluster deployment gets the database from Webroot.
- The IP reputation data is shared across all partitions in admin-partition deployments.
- You can use an AppExpert data set to create lists of IP addresses to add exceptions for IPs blacklisted in the Webroot database. You can also create your own customized blacklist to designate specific IPs as malicious.
- The iprep.db file is created in the /var/nslog/iprep folder. Once created, it is not deleted even if the feature is disabled.
- When the reputation feature is enabled, the NetScaler Webroot database is downloaded. After that, it is updated every 5 minutes.
- The Webroot database major version is currently version: 1.
- The minor version gets updated every day. The update version is incremented after every 5 minutes and is reset back to 1 when the minor version is incremented.
- PI expressions enable you to use IP reputation with other features, such as responder and rewrite.
- The IP addresses in the database are in decimal notation.

Debugging Tips:

- If you cannot see the reputation feature in GUI, verify that you have the right license
- Monitor the messages in /var/log/iprep.log for debugging.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Webroot connectivity**: If you see the "ns iprep: Not able to connect/resolve WebRoot" message, make sure that the appliance has internet access and DNS is configured.
- **Proxy server**: If you see the "ns iprep: iprep_curl_download:88 curl_easy_perform failed. Error code: 5 Err msg:couldnt resolve proxy name" message, make sure that the proxy server configuration is accurate.
- **IP Reputation feature not working**: The iprep process takes about five minutes to start after you enable the reputation feature. The IP reputation feature might not work for that duration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Content Filtering

Sep 03, 2013

Content filtering can do some of the same tasks as the Citrix NetScaler Application Firewall, and is a less CPU-intensive tool. It is limited, however, to examining the header portion of the HTTP request or response and to performing a few simple actions on connections that match. If you have a complex Web site that makes extensive use of scripts and accesses back-end databases, the Application Firewall may be the better tool for protecting that Web site. For more information about the Citrix NetScaler Application Firewall, see the *Citrix Application Firewall Guide* at http://support.citrix.com/article/CTX132360.

Content filtering is based on regular expressions that you can apply to either HTTP requests or HTTP responses. To block requests from a particular site, for example, you could use an expression that compares each request's URL to the URL specified in the expression. The expression is part of a policy, which also specifies an action to be performed on requests or responses that match the expression. For example, an action might drop a request or reset the connection.

Following are some examples of things you can do with content filtering policies:

- Prevent users from accessing certain parts of your Web sites unless they are connecting from authorized locations.
- Prevent inappropriate HTTP headers from being sent to your Web server, possibly breaching security.
- Redirect specified requests to a different server or service.

To configure content filtering, once you have made sure that the feature is enabled, you configure filtering actions for your servers to perform on selected connections (unless the predefined actions are adequate for your purposes). Then you can configure policies to apply the actions to selected connections. Your policies can use predefined expressions, or you can create your own. To activate the policies you configured, you bind them either globally or to specific virtual servers.

To configure content filtering, do the following:

1. [Enabling Content Filtering]
2. [Configuring a Content Filtering Action]
3. [Configuring a Content Filtering Policy]
4. [Binding a Content Filtering Policy]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Content Filtering

Oct 31, 2013

By default, content filtering is enabled on NetScaler appliances running the NetScaler operating system 8.0 or above. If you are upgrading an existing appliance from an operating system version earlier than 8.0, you must update the licenses before you can use content filtering, and you may need to enable the content filtering feature itself manually.

To enable content filtering by using the command line interface

At the command prompt, type the following commands to enable content filtering and verify the configuration:

- enable ns feature ContentFiltering
- show ns feature

Example

```
> enable ns feature ContentFiltering
 Done
> show ns feature

     Feature                Acronym        Status
     -------                -------        ------
 1)  Web Logging            WL             ON
 2)  Surge Protection       SP             OFF
 .
 .
 .
 .
 11) Http DoS Protection    HDOSP          OFF
 12) Content Filtering      CF             ON
 .
 .
 23) HTML Injection         HTMLInjection  ON
 24) NetScaler Push         push           OFF
 Done
```

To enable content by filtering by using the configuration utility

1. In the navigation pane, expand System, and then select Settings.
2. In the details pane, click Configure basic features.
3. In the Configure Basic Features pane, select the Content Filter check box, and then click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Content Filtering Action

Sep 03, 2013

After you enable the content filtering feature, you create one or more actions to tell your NetScaler appliance how to handle the connections it receives.

Content filtering supports the following actions for HTTP requests:

Add

Adds the specified HTTP header before sending the request to the Web server.

Reset

Terminates the connection, sending the appropriate termination notice to the user's browser.

Forward

Redirects the request to the designated service.

Drop

Silently deletes the request, without sending a response to the user's browser.

Corrupt

Modifies the designated HTTP header in a manner that prevents it from performing the function it was intended to perform, then sends the request to the server.

Content filtering supports the following actions for HTTP responses:

Add

Adds the specified HTTP header before sending the response to the user's browser.

ErrorCode

Returns the designated HTTP error code to the user's browser.

Corrupt

Modifies the designated HTTP header in a manner that prevents it from performing the function it was intended to perform, then sends the response to the user's browser.

To configure a content filtering action by using the command line interface

At the command prompt, type the following commands to configure a Content Filtering action and verify the configuration:

- add filter action <name> <qualifier> [<serviceName>] [<value>] [<respCode>] [<page>]
- show filter action <name>

Example

```
> add filter action act_drop Drop
 Done
> show filter action act_drop
1)     Name: act_drop   Filter Type: drop
 Done
```

To configure a content filtering action by using the configuration utility

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, do one of the following:
   - To create a new action, click Add.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- To modify an existing action, select the action, and then click Open.

3. In the Add Filter Action or Configure Filter Action dialog box, specify values for the parameters:
   - Action Name*—name
   - Qualifier*—qualifier (Determines which of the following parameters you can configure)
   - Service Name—servicename
   - HeaderName:Value—value
   - Response Code—respcode
   - Response Page—page
4. Fill in any other required information. For example, if you are configuring an action to send an HTTP error code, you must choose the appropriate error code from a drop-down list. If necessary, you can then modify the text of the error message, which is displayed beneath the drop-down list.
5. Click Create or OK, and then click Close. The Actions list displays the action you configured, and a message in the status bar indicates that your action has been created.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Content Filtering Policy

Sep 03, 2013

To implement content filtering, you must configure at least one policy to tell your NetScaler appliance how to distinguish the connections you want to filter. You must first have configured at least one filtering action, because when you configure a policy, you associate it with an action.

Content filtering policies examine a combination of one or more of the following elements to select requests or responses for filtering:

URL

The URL in the HTTP request.

URL query

Only the query portion of the URL, which is the portion after the query (?) symbol.

URL token

Only the tokens in the URL, if any, which are the parts that begin with an ampersand (&) and consist of the token name, followed by an equals sign (=), followed by the token value.

HTTP method

The HTTP method used in the request, which is usually GET or POST, but can be any of the eight defined HTTP methods.

HTTP version

The HTTP version in the request, which is usually HTTP 1.1.

Standard HTTP header

Any of the standard HTTP headers defined in the HTTP 1.1 specification.

Standard HTTP header value

The value portion of the HTTP header, which is the portion after the colon and space (:).

Custom HTTP header

A non-standard HTTP header issued by your Web site or that appears in a user request.

Custom header value

The value portion of the custom HTTP header, which (as with the standard HTTP header) is the portion after the colon and space (:).

Client Source IP

The IP from which the client request was sent.

Content filtering policies use the simpler of two NetScaler expressions languages, called classic expressions. For a complete description of classic expressions, how they work, and how to configure them manually, see "Policies and Expressions."

Note: Users who are not experienced in configuring policies at the NetScaler command line will usually find using the configuration utility considerably easier.

To configure a content filtering policy by using the command line interface

At the command prompt, type the following commands to configure a content filtering policy and verify the configuration:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- add filter policy <name> -rule <expression> (-reqAction <action> | -resAction <string>
- show filter policy <name>

Example

```
> add filter policy cf-pol -rule "REQ.HTTP.URL CONTAINS http://abc.com" -reqaction DROP
 Done
> show filter policy cf-pol
1)     Name: cf-pol    Rule: REQ.HTTP.URL CONTAINS http://abc.com
       Request action: DROP
       Response action:
       Hits: 0
 Done
```

To configure a content filtering policy by using the configuration utility

1. Navigate to Security > Protection Features > Filter.
2. Navigate to Protection Features > Filter.
3. In the details pane, to create a new policy, click Add.
4. If you are creating a new policy, in the Create Filter Policy dialog box, in the Filter Name text box, type a name for your new policy.
5. Select either Request Action or Response Action to activate the drop-down list to the right of that item.
6. Click the down arrow to the right of the drop-down list and select the action to be performed on the request or response. The default choices are RESET and DROP. Any other actions you have created will also appear in this list.
   Note: You can also click New to create a new Content Filtering action, or Modify to modify an existing Content Filtering action. You can only modify actions you created; the default actions are read-only.
7. If you want to use a predefined expression (or named expression) to define your policy, choose one from the Named Expressions list.
   1. Click the down arrow to the right of the first Named Expressions drop-down list, and choose the category of named expressions that contains the named expression you want to use.
   2. Click the down arrow to the right of the second Named Expressions drop-down list, and choose the named expression you want. As you choose a named expression, the regular expression definition of that named expression appears in the Preview Expression pane beneath the Named Expression list boxes.
   3. Click Add Expression to add that named expression to the Expression list.

   Note: You should perform either this step or step 7, but not both.
8. If you want to create a new expression to define your policy, use the Expression Editor.
   1. Click the Add button. The Add Expression dialog box appears.
   2. In the Add Expression dialog box, choose the type of connection you want to filter. The Flow Type is set to REQ by default, which tells the NetScaler appliance to look at incoming connections, or requests. If you want to filter outgoing connections (responses), you click the right arrow beside the drop-down list and choose RES.
   3. If the Protocol is not already set to HTTP, click the down arrow to the right of the Protocol drop-down list and choose HTTP.
      Note: In the NetScaler classic expressions language, “HTTP” includes HTTPS requests, as well.
   4. Click the down arrow to the right of the Qualifier drop-down list, and then choose a qualifier for your expression. Your choices are:
      METHOD
      The HTTP method used in the request.
      URL

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The contents of the URL header.

URLTOKENS

The URL tokens in the HTTP header.

VERSION

The HTTP version of the connection.

HEADER

The header portion of the HTTP request.

URLLEN

The length of the contents of the URL header.

URLQUERY

The query portion of the contents of the URL header.

URLQUERYLEN

The length of the query portion of the URL header.

The contents of the remaining list boxes change to the choices appropriate to the Qualifier you pick. For example, if you choose HEADER, a text field labeled Header Name* appears below the Flow Type list box.

5. Click the down arrow to the right of the Operator drop-down list, and choose an operator for your expression. Your choices will vary depending on the Protocol you chose in the preceding step. The following list includes all of the operators:

   ==

   Matches the following text string exactly.

   !=

   Does not exactly match the following text string.

   >

   Is greater than the following integer.

   CONTAINS

   Contains the following text string.

   CONTENTS

   The contents of the designated header, URL, or URL query.

   EXISTS

   The specified header or query exists.

   NOTCONTAINS

   Does not contain the following text string.

   NOTEXISTS

   The specified header or query does not exist.

6. If the Value text box is visible, type the appropriate string or number. If you are testing a string in any way, type the string into the Value text box. If you are testing an integer in any way, type the integer into the Value text box.
7. If you chose HEADER as the Protocol, type the header you want in the Header Name* text box.
8. Click OK to add your expression to the Expressions list.
9. Repeat steps B through H to create any additional expressions you want for your profile.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

10. Click Close to close the Expressions Editor.

9. If you created a new expression, in the Expression frame select an option from the Match Any Expression drop-down list. Your choices are:
   - Match Any Expression. If a request matches any expression in the Expressions list, the request matches this policy.
   - Match All Expressions If a request matches all expressions in the Expressions list, the request matches this policy. If it does not matchall of them, it does not match this policy.
   - Tabular Expression Switches the Expressions list to a tabular format with three columns. In the first column you can place a BEGIN [(] operator. The second column contains the expressions you have selected or created. In the third column, you can place any of the other operators in the following list, to create complex policy groups in which each group can be configured for match any expression or match all expressions.
   - The AND [&&] operator tells the appliance to require that are quest match both the current expression and the following expression.
   - The OR [| |] operator tells the appliance to require that arequest match either the current expression or the following expression, or both. Only if the request does not match either expression does it not match the policy.
   - The END [)] operator tells the appliance that this is the last expression in this expression group or policy.
     Note: The Tabular format allows you to create a complex policy that contains both “Match Any Expression” and “Match All Expressions” on a per-expression basis. You are not limited to just one or the other.
   - Advanced Free-Form Switches off the Expressions Editor entirely and modifies the Expressions list into a text area. In the text area, you can type the PCRE-format regular expression of your choice to define this policy. This is both the most powerful and the most difficult method of creating a policy, and is recommended only for those thoroughly familiar with the NetScaler appliance and PCRE-formatregular expressions.
     Caution: If you switch to Advanced Free Form expression editing mode, you cannot switch back to any of the other modes. Do not choose this expression editing mode unless you are sure that is what you want.

10. Repeat steps 6 through 8 to add any additional expressions you want to the Expressions list. You can mix named expressions and expressions created in the Expressions Editor. To the NetScaler appliance, they are all the same.

11. Click Create to create your new policy. Your new policy appears in the Policies pane list.

12. Click Close. To create additional Content Filtering policies, repeat the previous procedure. To remove a Content Filtering policy, select the policy in the Policies tab and click Remove.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding a Content Filtering Policy

Oct 31, 2013

You must bind each content filtering policy to put it into effect. You can bind policies globally or to a particular virtual server. Globally bound policies are evaluated each time traffic directed to any virtual server matches the policy. Policies bound to a specific vserver are evaluated only when that vserver receives traffic that matches the policy.

To bind a policy to a virtual server by using the command line interface

At the command prompt, type the following commands to bind a policy to a virtual server and verify the configuration:

- bind lb vserver <name>@ -policyName <string> -priority <positive_integer>
- show lb vserver <name>

Example

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver vs-loadbal
1)     vs-loadbal (10.102.29.20:80) - HTTP     Type: ADDRESS
       State: OUT OF SERVICE
       Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
       Time since last state change: 2 days, 00:58:03.260
       Effective State: DOWN
       Client Idle Timeout: 180 sec
       Down state flush: ENABLED
       Disable Primary Vserver On Down : DISABLED
       Port Rewrite : DISABLED
       No. of Bound Services :  0 (Total)       0 (Active)
       Configured Method: LEASTCONNECTION
       Mode: IP
       Persistence: NONE
       Vserver IP and Port insertion: OFF
       Push: DISABLED  Push VServer:
       Push Multi Clients: NO
       Push Label Rule: none

 Done
```

To globally bind a policy by using the command line interface

At the command prompt, type the following commands to globally bind a policy and verify the configuration:

- bind filter global (<policyName> [-priority <positive_integer>]) [-state (ENABLED | DISABLED)]
- show filter global

Example

```
bind filter global cf-pol -priority 1
Done show filter global
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

1) Policy Name: cf-pol Priority: 1
Done

To bind a policy to a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server to which you want to bind the content filtering policy from the list, and click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, select the Policies tab, and then select the check box in the Active column of the filter policy that you want to bind to the virtual server.
4. Click OK. The policies you have bound display a check mark and the word Yes in the Policies Bound column of the Policies tab.

To globally bind a policy by using the configuration utility

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, in the Policies tab, select the policy that you want to bind, and then click Global Bindings.
3. In the Bind/Unbind Filter Policies dialog box, in the Policy Name drop-down list, select a policy, and then click Add. The policy is added to the Configured list.
   Note: To select multiple policies from the list, press and hold the Ctrl key, then click each policy you want.
4. Click OK, and then click Close. The policies you have bound display a check mark and the word Yes in the Globally Bound column of the Policies tab.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Content Filtering for a Commonly Used Deployment Scenario

Oct 31, 2013

This example provides instructions for using the configuration utility to implement a content filtering policy in which, if a requested URL contains root.exe or cmd.exe, the content filtering policy filter-CF-nimda is evaluated and the connection is reset.

To configure this content filtering policy, you must do the following:

- Enable content filtering
- Configure content filtering policy
- Bind content filtering policy globally or to a virtual server
- Verify the configuration

Note: Since this example uses a default content filtering action, you do not need to create a separate content filtering action.

To enable content filtering

1. In the navigation pane, expand System, and click Settings.
2. In the details pane, under Modes & Features, click Change Basic Features.
3. In the Configure Basic Features dialog box, select the Content Filtering check box, and then click OK.
4. In the Enable/Disable feature(s) dialog box, click Yes. A message appears in the status bar, stating that the selected feature is enabled.

To configure the content filtering policy filter-CF-nimda

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, click Add. The Create Filter Policy dialog box appears.
3. In the Create Filter Policy dialog box, in the Filter Name text box, type the name filter-CF-nimda.
4. Select the Request Action option, and in the drop-down list, select RESET.
5. In the Expression frame, select Match Any Expression from the drop-down list, and then click Add.
6. In the Add Expression dialog box, Expression Type drop-down list, select General.
7. In the Flow Type drop-down list, select REQ.
8. In the Protocol drop-down list, select HTTP.
9. In the Qualifier drop-down list, select URL.
10. In the Operator drop-down list, select CONTAINS.
11. In the Value text box, type cmd.exe, and then click OK. The expression is added in the Expression text box.
12. To create another expression, repeat Steps 7 through 11, but in the Value text box, type root.exe. Then click OK, and finally click Close.
13. Click Create on the Create Filter Policy dialog box. The filter policy filter-CF-nimda appears in the Filter list.
14. Click Close.

To globally bind the content filtering policy

1. Navigate to Security > Protection Features > Filter. The Filter page appears in the right pane.
2. In the details pane, Policies tab, select the policy that you want to bind and click Global Bindings. The Bind/Unbind Filter Policies dialog box appears.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. In the Bind/Unbind Filter Policies dialog box, in the Policy Name drop-down list, select the policy filter-CF-nimda, and click Add. The policy is added to the Configured list.
4. Click OK, and then click Close. The policy you have bound displays a check mark and Yes in the Globally Bound column of the Policies tab.

To bind the content filtering policy to a virtual server

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane virtual servers list, select vserver-CF-1 to which you want to bind the content filtering policy and click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, select the Policies tab.
4. In the Active column, select the check box for the policy filter-CF-nimda, and then click OK. Your content filtering policy is now active, and should be filtering requests. If it is functioning correctly, the Hits counter is incremented every time there is a request for a URL containing either root.exe or cmd.exe. This allows you to confirm that your content filtering policy is working. The content filtering policy is bound to the virtual server.

To verify the content filtering configuration by using the command line interface

At the command prompt, type the following command to verify the content filtering configuration:
show filter policy filter-CF-nimda

Example

```
sh filter policy filter-CF-nimda
    Name: filter-CF-nimda   Rule: REQ.HTTP.URL CONTAINS cmd.exe || REQ.HTTP.URL CONTAINS root.exe
    Request action: RESET
    Response action:
    Hits: 0
Done
```

Note: The Hits counter displays an integer that denotes the number of times the filter-CF-nimda policy is evaluated. In the preceding steps, the Hits counter is set to zero because no requests for a URL containing either cmd.exe or root.exe have been made yet. If you want to see the counter increment in real time, you can simply request a URL that contains either of these strings.

To verify the content filtering configuration by using the configuration utility

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, select the filter policy filter-CF-nimda. The bottom of the pane should display the following:

   Request Action:

   RESET

   Rule:

   REQ.HTTP.URL CONTAINS cmd.exe || REQ.HTTP.URL CONTAINS root.exe

   Hits:

   0

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Troubleshooting

Jul 22, 2013

If the content filtering feature does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for Troubleshooting

Updated: 2013-07-22

You can use the following tools and resources to troubleshoot most Content Filtering issues on a NetScaler appliance:

- The Wireshark application customized for the NetScaler trace files
- Trace files recorded when accessing the resource
- The configuration files
- The ns.log file
- The iehttpheaders, or a Fiddler trace or a similar utility

Troubleshooting Content Filtering Issues

Updated: 2013-08-02

To troubleshoot a content filtering issue, proceed as follows:

- Verify that the feature is enabled.
- Verify that the content filtering policy is configured correctly. Pay special attention to the expression that evaluates the incoming requests.
  Note: Most content filtering issues are caused by incorrect configuration, and the error is most often in the policy configuration.
- Check the policy's Hits counter to verify that it is incrementing. If it is not, the policy is not getting evaluated.
- If the policy is getting evaluated and the required filtering is still not performed, you need to look into the policy expressions and action.
- If the policy's expression seems valid, test it by assigning a simple NSTRUE value to see if the evaluation of the expression is creating any issue.
- Reevaluate whether the filtering should be based on the request or the response.
- Verify that the action is configured correctly. For example, if a custom action is used to corrupt a header in the request, verify that the header name in the action is correct. If you are not sure about the header name, start a browser with iehttpheaders or a similar utility, and then verify the headers in the request. When this feature is used, you can use nstrace to find out if appropriate action is performed when the packets leave NetScaler appliance.
- An iehttpheaders or Fiddler trace can help you find header options and names, client-side request headers, and response headers recorded on the client.
- To check the modifications made to the request header, record an nstrace on the NetScaler appliance or a Wireshark trace on the server.
- If none of the above measures resolves the issue, verify that the connection has not become untrackable, which can happen in certain circumstances. If a connection becomes untrackable, the appliance does not perform any application-level processing of the requests. In that event, contact Citrix Technical Support.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Denial-of-Service Protection

Mar 16, 2012

Internet hackers can bring down a site by sending a surge of GET requests or other HTTP-level requests. HTTP Denial-of-Service (HTTP Dos) Protection provides an effective way to prevent such attacks from being relayed to your protected Web servers. The HTTP DoS feature also ensures that a NetScaler appliance located between the internet cloud and your Web servers is not brought down by an HTTP DoS attack.

Most attackers on the Internet use applications that discard responses to reduce computation costs, and minimize their size to avoid detection. The attackers focus on speed, devising ways to send attack packets, establish connections or send HTTP requests as rapidly as possible.

Real HTTP clients such as Internet Explorer, Firefox, or NetScape browsers can understand HTML Refresh meta tags, Java scripts, and cookies. In standard HTTP the clients have most of these features enabled. However, the dummy clients used in DoS attacks cannot parse the response from the server. If malicious clients attempt to parse and send requests intelligently, it becomes difficult for them to launch the attack aggressively.

When the NetScaler appliance detects an attack, it responds to a percentage of incoming requests with a Java or HTML script containing a simple refresh and cookie. (You configure that percentage by setting the Client Detect Rate parameter.) Real Web browsers and other Web-based client programs can parse this response and then resend a POST request with the cookie. DoS clients drop the NetScaler appliance's response instead of parsing it, and their requests are therefore dropped as well.

Even when a legitimate client responds correctly to the NetScaler appliance's refresh response, the cookie in the client's POST request may become invalid in the following conditions:

- If the original request was made before the NetScaler appliance detected the DoS attack, but the resent request was made after the appliance had come under attack.
- When the client's think time exceeds four minutes, after which the cookie becomes invalid.

Both of these scenarios are rare, but not impossible. In addition, the HTTP DoS protection feature has the following limitations:

- Under an attack, all POST requests are dropped, and an error page with a cookie is sent.
- Under an attack, all embedded objects without a cookie are dropped, and an error page with a cookie is sent.

The HTTP DoS protection feature may affect other NetScaler features. Using DoS protection for a particular content switching policy, however, creates additional overhead because the policy engine must find the policy to be matched. There is some overhead for SSL requests due to SSL decryption of the encrypted data. Because most attacks are not on a secure network, though, the attack is less aggressive.

If you have implemented priority queuing, while it is under attack a NetScaler appliance places requests without proper cookies in a low-priority queue. Although this creates overhead, it protects your Web servers from false clients. HTTP DoS protection typically has minimal effect on throughput, since the test JavaScript is sent for a small percentage of requests only. The latency of requests is increased, because the client must re-issue the request after it receives the JavaScript. These requests are also queued

To implement HTTP DoS protection, you enable the feature and define a policy for applying this feature. Then you configure your services with the settings required for HTTP DoS. You also bind a TCP monitor to each service and bind your

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

policy to each service to put it into effect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Layer 3-4 SYN Denial-of-Service Protection

May 18, 2017

Any NetScaler appliance with system software version 8.1 or later automatically provides protection against SYN DoS attacks.

To mount such an attack, a hacker initiates a large number of TCP connections but does not respond to the SYN-ACK messages sent by the victimized server. The source IP addresses in the SYN messages received by the server are typically spoofed. Because new SYN messages arrive before the half-open connections initiated by previous SYN messages time out, the number of such connections increases until the server no longer has enough memory available to accept new connections. In extreme cases, the system memory stack can overflow.

A NetScaler appliance defends against SYN flood attacks by using SYN cookies instead of maintaining half-open connections on the system memory stack. The appliance sends a cookie to each client that requests a TCP connection, but it does not maintain the states of half-open connections. Instead, the appliance allocates system memory for a connection only upon receiving the final ACK packet, or, for HTTP traffic, upon receiving an HTTP request. This prevents SYN attacks and allows normal TCP communications with legitimate clients to continue uninterrupted.

SYN DoS protection on the NetScaler appliance ensures the following:

- The memory of the NetScaler is not wasted on false SYN packets. Instead, memory is used to serve legitimate clients.
- Normal TCP communications with legitimate clients continue uninterrupted, even when the Web site is under SYN flood attack.

In addition, because the NetScaler appliance allocates memory for HTTP connection state only after it receives an HTTP request, it protects Web sites from idle connection attacks.

SYN DoS protection on your NetScaler appliance requires no external configuration. It is enabled by default.

Disabling SYN Cookies

SYN cookies are enabled by default on a NetScaler appliance to prevent SYN attacks. If your deployment requires you to disable SYN cookies, for example, for server-initiated data connections or in cases where a connection is not established because the first packet is dropped or reordered, use one of the following methods to disable SYN cookies.

To disable SYN cookies by using the NetScaler command line

At the command prompt, type:

set nstcpprofile nstcp_default_profile -synCookie DISABLED

Arguments

synCookie

Enable or disable the SYNCOOKIE mechanism for TCP handshake with clients. Disabling SYNCOOKIE prevents SYN attack protection on the NetScaler appliance.

Possible values: ENABLED, DISABLED

Default: ENABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To disable SYN cookies by using the NetScaler GUI

1. Navigate to **System** > **Profiles** > **TCP Profiles**.
2. Select a profile and click **Edit**.
3. Clear the **TCP SYN Cookie** check box.
4. Click **OK**.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling HTTP DoS Protection

Sep 03, 2013

To configure HTTP DoS protection, you must first enable the feature.

To enable HTTP DoS protection by using the command line interface

At the command prompt, type the following commands to enable HTTP DoS protection and verify the configuration:

- enable ns feature HttpDoSProtection
- show ns feature

Example

```
> enable ns feature HttpDoSProtection
 Done
> show ns feature

      Feature                    Acronym          Status
      -------                  -------          ------
 1)   Web Logging                WL               ON
 2)   Surge Protection           SP               OFF
 .
 .
 .
 10)  Global Server Load Balancing  GSLB          ON
 11)  Http DoS Protection        HDOSP            ON
 12)  Content Filtering          CF               ON
 .
 .
 23)  HTML Injection             HTMLInjection    ON
 24)  NetScaler Push             push             OFF
 Done
>
```

To enable HTTP DoS protection by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, click Configure Advanced Features.
3. In the Configure Advanced Features dialog box, select the HTTP DoS Protection check box.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Defining an HTTP DoS Policy

Sep 03, 2013

After you enable HTTP DoS protection, you next create a policy.

Note: Before changing the default setting for Client Detect Rate, see "Tuning the Client Detection/JavaScript Challenge Response Rate."

To configure a HTTP DoS policy by using the command line interface

At the command prompt, type one of the following commands to configure an HTTP DoS policy and verify the configuration:

- add dos policy <name> -qDepth <positive_integer> [-cltDetectRate <positive_integer>]
- set dos policy <name> -qDepth <positive_integer> [-cltDetectRate <positive_integer>]

Example

```
> add dos policy pol-HTTP-DoS  -qDepth 30
 Done
> set dos policy pol-HTTP-DoS  -qDepth 40
 Done
> show dos policy
1)     Policy: pol-HTTP-DoS    QDepth: 40
 Done
>
```

To configure an HTTP DoS policy by using the configuration utility

1. Navigate to Security > Protection Features > HTTP DoS.
2. In the details pane, do one of the following:
   - To create a new policy, click Add.
   - To modify an existing policy, select the policy, and then click Open.
3. In the Create HTTP DoS Policy or Configure HTTP DoS Policy dialog box, specify values for the parameters:
   - Name*—name (You cannot change the name of an existing policy.)
   - QDepth*—qdepth
   - Client Detect Rate—cltDetectRate (Before changing the default setting for cltDetectRate, see "Tuning the Client Detection/JavaScript Challenge Response Rate.")
4. Click OK to create your new policy. The policy that you created appears in the details pane, and the status bar displays a message indicating that the DoS policy is successfully configured.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring an HTTP DoS Service

Sep 03, 2013

After you configure an HTTP DoS policy, you must configure a service for your policy. The service accepts HTTP traffic that is protected by the HTTP DoS policy.

To configure an HTTP DoS service by using the command line interface

At the command prompt, type one of the following commands to configure an HTTP DoS service and verify the configuration:

- add service <name>@ (<IP>@ | <serverName>@) HTTP <port> [-maxClient <positive_integer>] [-maxReq <positive_integer>] -state ENABLED
- set service <name>@ (<IP>@ | <serverName>@) HTTP <port> [-maxClient <positive_integer>] [-maxReq <positive_integer>] -state ENABLED

Example

```
> add service ser-HTTP-Dos1 10.102.29.40 HTTP 87
 Done
> set service ser-HTTP-Dos1 -maxReq 20
 Done
> show service
1)     srv-http-10 (10.102.29.30:80) - HTTP
       State: DOWN
       Last state change was at Wed Jul  8 07:49:52 2009
       Time since last state change: 34 days, 00:48:18.700
       Server Name: 10.102.29.30
       Server ID : 0   Monitor Threshold : 0
       Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
       Use Source IP: NO
       Client Keepalive(CKA): NO
       Access Down Service: NO
       TCP Buffering(TCPB): NO
       HTTP Compression(CMP): NO
       Idle timeout: Client: 180 sec   Server: 360 sec
       Client IP: DISABLED
       Cacheable: NO
       SC: OFF
       SP: OFF
       Down state flush: ENABLED
        .
        .
        .

5)     ser-HTTP-Dos1 (10.102.29.40:87) - HTTP
       State: DOWN
       Last state change was at Tue Aug 11 08:23:40 2009
       Time since last state change: 0 days, 00:14:30.300
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Server Name: 10.102.29.40
    Server ID : 0   Monitor Threshold : 0
    Max Conn: 0     Max Req: 20     Max Bandwidth: 0 kbits
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
    TCP Buffering(TCPB): NO
    HTTP Compression(CMP): YES
    Idle timeout: Client: 180 sec   Server: 360 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: OFF
    Down state flush: ENABLED
 Done
>
```

To configure an HTTP DoS service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, do one of the following:
   - To create a new service, click Add.
   - To modify an existing service, select the service, and then click Open.
3. In the Create Server or Configure Server dialog box, specify values for the following parameters, which correspond to the descriptions in "Parameters for configuring an HTTP DoS service" as follows (asterisk indicates a required parameter):
   - Service Name*—name (You cannot change the name of an existing service.)
   - Server*—IP or serverName (Specify one or the other, not both.)
   - Port*—port
4. If the Enable Service check box is not selected, select it.
5. Select the Advanced tab, and select the Override Global check box to enable those choices.
6. Specify values for the following parameters.
   - Max Clients*—maxClient
   - Max Requests*—maxReq
7. Click Create or OK, and then click Close. The service appears in the list of services.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding an HTTP DoS Monitor and Policy

Sep 03, 2013

To put HTTP DoS protection into effect after you have configured an HTTP DoS service, you must bind the monitor, and then bind the service to the HTTP DoS policy.

To bind the monitor to the service by using the command line interface

At the command prompt, type the following commands to bind the monitor to the service and verify the configuration:

- bind lb monitor <monitorName> <serviceName>
- show lb monitor

Example

```
> bind lb monitor tcp ser-HTTP-DoS
 Done
> show lb monitor
1)  Name.......:  ping-default  Type......:    PING   State....ENABLED
2)  Name.......:   tcp-default  Type......:     TCP   State....ENABLED
3)  Name.......:         ping  Type......:    PING   State....ENABLED
4)  Name.......:          tcp  Type......:     TCP   State....ENABLED
5)  Name.......:         http  Type......:    HTTP   State....ENABLED
 .
 .
 .
17)  Name.......:      ldns-dns  Type......: LDNS-DNS   State....ENABLED
 Done
```

To bind the policy to the service by using the command line interface

At the command prompt, type the following commands to bind the policy to the service and verify the configuration:

bind service <serviceName> -policyName <policyname>

Example

```
> bind service ser-HTTP-DoS -policyName pol-HTTP-DoS
 Done
> show service
1)    srv-http-10 (10.102.29.30:80) - HTTP
      State: DOWN
      Last state change was at Wed Jul  8 07:49:52 2009
      Time since last state change: 34 days, 01:24:58.510
      Server Name: 10.102.29.30
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): NO
      HTTP Compression(CMP): NO
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    Idle timeout: Client: 180 sec   Server: 360 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: ON
    Down state flush: ENABLED
    .
    .
    .
4)   ser-HTTP-Dos (10.102.29.18:88) - HTTP
    State: DOWN
    Last state change was at Tue Aug 11 08:19:45 2009
    Time since last state change: 0 days, 00:55:05.40
    Server Name: 10.102.29.18
    Server ID : 0   Monitor Threshold : 0
    Max Conn: 0     Max Req: 0     Max Bandwidth: 0 kbits
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
    TCP Buffering(TCPB): NO
    HTTP Compression(CMP): YES
    Idle timeout: Client: 180 sec   Server: 360 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: ON
    Down state flush: ENABLED
5)   ser-HTTP-Dos1 (10.102.29.40:87) - HTTP
    State: DOWN
    Last state change was at Tue Aug 11 08:23:40 2009
    Time since last state change: 0 days, 00:51:10.110
    Server Name: 10.102.29.40
    Server ID : 0   Monitor Threshold : 0
    Max Conn: 0     Max Req: 20     Max Bandwidth: 0 kbits
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
    TCP Buffering(TCPB): NO
    HTTP Compression(CMP): YES
    Idle timeout: Client: 180 sec   Server: 360 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: OFF
    Down state flush: ENABLED
 Done
>
```

To bind the monitor and policy to the service by using the configuration utility

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, select the service that you want to bind, and then click Open.
3. In the Configure Service dialog box, select the Monitor tab, click the name of the monitor you want in the Monitors list, and then click Add. The selected monitor is added to the Configured frame.
4. Select the Policies tab, then select the HTTP DoS tab.
5. Select a policy from the Available Policies list, and then click Add. The policy appears in the Configured Policies list.
6. Click OK, and then click Close. A message appears in the status bar, stating that the service has been configured.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tuning the Client Detection/JavaScript Challenge Response Rate

Sep 03, 2013

After you have enabled and configured HTTP DoS protection, if more than the maximum specified number of clients are waiting in the NetScaler surge queue for the HTTP DoS service, the HTTP DoS protection function is triggered. The default rate of challenged JavaScript responses sent to the client is one percent of the server response rate. The default response rate is inadequate in many real attack scenarios, however, and may need to be tuned.

For example, assume that the Web server is capable of a maximum of 500 responses/sec, but is receiving 10,000 Gets/sec. If 1% of the server responses are sent as JavaScript challenges, responses are reduced to almost none: 5 client (500 *0.01) JavaScript responses, for 10000 waiting client requests. Only about 0.05% of the real clients receive JavaScript challenge responses. However, if the client detection/JavaScript challenge response rate is very high (for example, 10%, generating 1000 challenge JavaScript responses per second), it may saturate the upstream links or harm the upstream network devices. Exercise care when modifying the default **Client Detect Rate** value.

If the configured triggering surge queue depth is, for example, 200, and the surge queue size is toggling between 199 and 200, the NetScaler toggles between the “attack” and “no-attack” modes, which is not desirable. The HTTP DoS feature includes a window mechanism is provided. When the surge queue size reaches the designated queue depth value, triggering “attack” mode, the surge queue size must fall for the NetScaler to enter “no-attack” mode. In the scenario just described, if the value of WINDOW_SIZE is set to 20, the surge queue size must fall below 180 before the NetScaler enters “no-attack” mode. During configuration, you must specify a value more than the WINDOW_SIZE for the **QDepth** parameter when adding a DoS policy or setting a DoS policy.

The triggering surge queue depth should be configured on the basis of previous observations of traffic characteristics. For more information about setting up a correct configuration, see "Guidelines for HTTP DoS Protection Deployment."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Guidelines for HTTP DoS Protection Deployment

Mar 16, 2012

Citrix recommends you to deploy the HTTP DoS protection feature in a tested and planned manner and closely monitor its performance after the initial deployment. Use the following information to fine-tune the deployment of HTTP DoS Protection.

- The maximum number of concurrent connections supported by your servers.
- The average and normal values of the concurrent connections supported by your servers.
- The maximum output rate (responses/sec) that your server can generate.
- The average traffic that your server handles.
- The typical bandwidth of your network.
- The maximum bandwidth available upstream.
- The limits affecting bandwidth (such as external links, a particular router, or other critical devices on the path that may suffer from a traffic surge).
- Whether allowing a greater number of clients to connect is more important than protecting upstream network devices.

To determine the characteristics of a HTTP DoS attack, you should consider the following issues.

- What is the rate of incoming fake requests that you have experienced in the past?
- What types of requests have you received (complete posts, incomplete gets)?
- Did previous attacks saturate your downstream links? If not, what was the bandwidth?
- What types of source IP addresses and source ports did the HTTP requests have (e.g., IP addresses from one subnet, constant IP, ports increasing by one).
- What types of attacks do you expect in future? What type have you seen in the past?
- Any or all information that can help you tune DoS attack protection.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Priority Queuing

Sep 03, 2013

The priority queuing feature lets you filter incoming HTTP traffic on the basis of categories that you create and define, and prioritize those HTTP requests accordingly. Priority queuing directs high-priority requests to the server ahead of low-priority requests, so that users who need resources for important business uses receive expedited access to your protected Web servers.

Note: The priority queuing feature is not supported in NetScaler 9.2 nCore.
To implement priority queuing, you create priority queuing policies that specify a priority, weight, threshold, and implicit action. When an incoming request matches a priority queuing policy, the request is processed as the associated action indicates. For example, you can create a priority queuing policy that places all matching requests above a certain threshold in a surge queue, while giving priority treatment to other requests.

You can bind up to three priority queuing policies to a single load balancing virtual server. The priority levels are:

Level 1

A Level 1 policy processes priority requests.

Level 2

A Level 2 policy processes requests that should receive responses as soon as Level 1 requests have been cleared from the queue.

Level 3

A Level 3 policy processes non-priority requests that receive responses only after requests in the first two queues have been cleared.

You can use weighted queuing to adjust the relative priority of each of these queues. Weights can range from 0 to 101. A weight of 101 tells the NetScaler appliance to clear all requests in that queue before forwarding any requests in the lower-priority queues to the Web server. A weight of 0 tells the appliance to send requests in that queue to the Web server only when there are no requests waiting in any of the other queues.

You must assign a unique name to each priority queuing policy. Policy names can be up to 127 characters. Multiple policies bound to the same load balancing virtual server cannot have the same priority level. No two virtual servers that have one or more common underlying physical services can have priority queuing configured or enabled on both virtual servers simultaneously.

To configure priority queuing the NetScaler, you perform the following steps:

- Enable the load balancing feature
- Define a server and service
- Define a load balancing virtual server
- Bind the service to the load balancing virtual server
- Enable the priority queuing feature
- Create the priority queuing policies
- Bind the priority queuing policies to the load balancing virtual server
- Enable priority queuing on load balancing virtual server

For information about enabling load balancing, creating servers, creating virtual servers and services, and binding these servers and services, see the *Citrix NetScaler Traffic Management Guide* at http://support.citrix.com/article/CTX132359. For

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

complete information about policies and expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at http://support.citrix.com/article/CTX132362.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling Priority Queuing

Sep 03, 2013

To use the priority queuing feature the NetScaler appliance, you must first enable it.

To enable priority queuing by using the command line interface

At the command prompt, type the following commands to enable priority queuing and verify the configuration:

- enable ns feature PriorityQueuing
- show ns feature

Example

```
> enable ns feature PriorityQueuing
 Done
> show ns feature

     Feature                Acronym          Status
     -------                -------          ------
 1)  Web Logging            WL               ON
 2)  Surge Protection       SP               OFF
 3)  Load Balancing         LB               ON
 .
 .
 .
 8)  Priority Queuing       PQ               ON
 .
 23) HTML Injection         HTMLInjection    ON
 24) NetScaler Push         push             OFF
 Done
```

To enable priority queuing by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, click Configure advanced features.
3. In the Configure Advanced Features dialog box, select the Priority Queuing check box.
4. Click OK.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a Priority Queuing Policy

Sep 03, 2013

To configure a priority queuing policy, you can use either the configuration utility or the command line.

Note: For more information about using the command line, see "Command Reference."

To configure a priority queuing policy by using the command line interface

At the command prompt, type the following command to configure a priority queuing policy and verify the configuration:

add pq policy <policyName> -rule <expression> -priority <positive_integer> [-weight <positive_integer>] [-qDepth <positive_integer> | -polqDepth <positive_integer>]

Example

```
> add pq policy pol_cgibin -rule "URL == '/cgi-bin/'" -priority 1
 Done
> show pq policy pol_cgibin
1)     Policy: pol_cgibin      Rule: URL == '/cgi-bin/'       Priority: 1     Weight: 10
       Hits: 0
 Done
```

To configure a priority queuing policy by using the configuration utility

1. Navigate to Security > Protection Features > Priority Queuing.
2. In the details pane, do one of the following:
   - To create a new policy, click Add.
   - To modify an existing policy, select the policy, and then click Open.
3. If you are creating a new policy, in the Create PQ Policy dialog box, in the Name text box, type a name for your new policy.
   The name can consist of from one to 127 letters, numbers and the hyphen and underscore symbol.

   If you are modifying an existing policy, skip this step. You cannot change the name of an existing policy.

4. In the Rule text box, either enter the policy expression directly, or click New to create a policy expression. If you click New, perform the following steps:
   1. In the Create Expression dialog box, click Add.
   2. In the Add Expression dialog box, leave Expression Type set to General, and in the Flow Type drop-down list, select a Flow Type. Your choices are REQ (for requests) and RES (for responses).
   3. In the Protocol drop-down list, select a protocol. If you selected REQ in the previous step, your choices are HTTP (Web-based connections), SSL (secure Web connections), TCP and IP. If you selected RES in the previous step, your choices are HTTP, TCP and IP.
   4. In the Qualifier drop-down list, select a qualifier.
      Your choices depend upon your selections in the previous step. Common choices are HTTP VERSION (the version of the HTTP connection), HTTP HEADER (the specified HTTP header), TCP SOURCEPORT/ DESTPORT (the source or destination port of a TCP connection), and IP SOURCEIP/DESTIP (the source or destination IP of the connection).

      If you choose HTTP HEADER, the Header text box appears beneath the original row of text boxes. You fill in the name of the HTTP header you want.

      For a complete description of the available choices, see "Policies and Expressions."

© 1999-2017 Citrix Systems, Inc. All rights reserved.

5. In the Operator drop-down list, select an operator.
   For a complete description of the available choices, see "Policies and Expressions."

6. In the Value text box, type the value you want to test for.
   This may be a text string or a number, depending upon the context. For a complete description of values appropriate to the specific context, see "Policies and Expressions."

7. Click OK. The expression is added in the Expression text box.
8. Click Create. The expression appears in the Rule text box.

5. In the Priority and Weight text boxes, type numeric values, for example, 1 and 30. For more information about Priority and Weight, see "Setting Up Weighted Queuing."
6. Enter a numeric value for either Queue Depth or Policy Queue Depth, for example 234, and click Create.
   - Queue Depth Defines the total number of waiting clients or requests on the virtual server to which the policy is bound.
   - Policy Queue Depth Defines the total number of waiting clients or requests belonging to the policy.

   The policy is created and appears in the Priority Queuing page.
   Note: To create additional priority queuing policies, repeat the procedure in the preceding section, and click Close after you finish.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding a Priority Queuing Policy

Sep 03, 2013

After you create a priority queuing policy, you must bind it to the appropriate virtual server to put it into effect.

To bind a priority queuing policy by using the command line interface

At the command prompt, type the following commands to bind a policy and verify the configuration:

- bind lb vserver <name> -policyName <policyname>
- show lb vserver <name>

Example

```
> bind lb vserver lbvip -policyname pol_cgibin
 Done
> show lb vserver lbvip
      lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
      State: DOWN
      Last state change was at Wed Jul 15 05:54:24 2009 (+782 ms)
      Time since last state change: 26 days, 05:44:37.370
      Effective State: DOWN
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
      No. of Bound Services :  0 (Total)       0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Vserver IP and Port insertion: OFF
      Push: DISABLED  Push VServer:
      Push Multi Clients: NO
      Push Label Rule: none

1)     Policy : ns_cmp_msapp Priority:0

1)     Priority Queuing Policy : pol_cgibin
 Done
>
```

To bind a priority queuing policy by using the configuration utility

1. In the navigation pane, locate and select the virtual server to which you want to bind the priority queuing policy.
   - To select a load balancing virtual server, expand Traffic Management > Load Balancing > Virtual Servers, then select the load balancing virtual server that you want..
   - To select a content switching virtual server, expand Traffic Management > Content Switching > Virtual Servers, then select the content switching virtual server that you want..
2. In the Configure Virtual Server dialog box, select the Policies tab.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

3. Click the double right-arrow (») symbol to display the complete list of policy types, and then select Priority Queuing from the drop-down list.
4. Click Insert Policy.
5. In the Policy Name row, select the policy that you want to bind from the drop-down list.
6. ClickOK to save your changes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Setting Up Weighted Queuing

Sep 03, 2013

When priority queuing is implemented, lower-priority requests are typically kept on hold while higher-priority requests are served. The lower-priority requests may therefore be delayed if there is a constant flow of higher-priority requests.

To prevent delays for low-priority requests across multiple priority levels, you can configure weighted queuing for serving requests. The default weights for the priorities are:

- Gold - Priority 1 - Weight 3
- Silver - Priority 2 - Weight 2
- Bronze - Priority 3 - Weight 1

You assign the minimum weight, zero (0), to requests that the NetScaler appliance should send to the server only if no requests are stored in any of the other queues. You assign the maximum weight, 101, to requests that the appliance should send to the server immediately, ahead of any requests stored in any of the other queues. Weights between these two set the relative priority of a particular queue in relation to the other queues. Queues with a higher weight are processed first; queues with a lower weight after the others have been processed. To assign the weights, see "Configuring a Priority Queuing Policy."

Note: The weight assigned to a higher-priority queue must be larger than the weight assigned to a lower-priority queue. For example, the weight assigned to The Gold (Priority 1) queue must be greater than the weight assigned to the Silver (Priority 2) queue.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SureConnect

Sep 03, 2013

You can use the SureConnect feature of the Citrix NetScaler appliance to service all incoming connections with either the requested content or a custom Web page that displays information about a delay in the request being serviced.

When servers are overloaded with the requests, the servers might either respond slowly or not at all. The SureConnect feature enables the NetScaler appliance to detect and compensate such conditions by ensuring that every client request gets serviced in some way, such as either a custom Web page or actual content is sent to the client.

SureConnect is activated when the response time or maximum server connections to a client request exceeds a limit that you have set. The SureConnect browser window displays one of the following:

- A progress bar with the amount of time remaining until the requested content will be available.
- Alternate Web content of your choice (alternate page).
- Both a progress bar and alternate page.
- Complete custom content of your choice.

You can configure whether the SureConnect progress bar alone is displayed or both the progress bar and the alternate page are displayed.

When the server becomes responsive again, the original request for content is served. If the user chooses, the alternate content window can remain in focus.

Subsequent requests from the same user within the same session are served immediately. This can be configured using the settings described later in this section.

SureConnect can be activated when a response is delayed, and when the number of user connections to a given URL exceeds a specified threshold.

SureConnect works with all standard browsers, including Microsoft Internet Explorer, Netscape Navigator, and Mozilla Firefox.

SureConnect is advantageous in the following situations:

- **Full server queue**

  The server can respond fast, but there are too many users. This results in the server's queue being full and unable to process additional client requests.

  SureConnect Solution: In this situation, the SureConnect window is displayed, showing the time left until the content will be available. The alternate page is displayed under the progress bar, if an alternate page has been configured.

- **Large response delay**

  The server response is slow. Typically, if a Web server does not respond to a client request quickly, the user will leave the site.

  SureConnect Solution: When the predicted delay reaches a configured time threshold, the SureConnect window displays the progress bar and the optional alternate page in the client browser.

- **Client time-out**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

When the client requests content from a very slow Web site, a time-out message displays in the client browser, and the content is not delivered. The user may leave the site.

SureConnect Solution: The appliance stores the request until the server is no longer busy and delivers the requested content to the client.

- **Server experiencing a traffic surge**

  The server typically responds quickly, but the current load of open connections is greater than the server capacity to serve them. Therefore, the server response is delayed.

  SureConnect Solution: A SureConnect window is displayed in the client browser, showing the time left. The alternate page from the server is also displayed if it has been configured.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Installing SureConnect

Mar 21, 2012

SureConnect files must be installed on the alternate content server, which can be the same as the primary server.

On a Windows server, extract the sc_xx.exe file (where xx is the build number), or on a UNIX server, extract the sc_xx.tar file (where xx is the build number).

Note: You must install SureConnect in the default Web root directory.
If the alternate content server is the same as the primary server, place the SureConnect and alternate content files in any directory under the Web root directory. Specify this path when you add a policy to configure SureConnect. By default, SureConnect files are installed in the /Citrix NetScaler appliance directory under the default Web root directory.

If the alternate content server is different than the primary server, the SureConnect and alternate content files must be in a unique directory under the Web root directory. By default, this unique directory is the /Citrix NetScaler system directory. Specify this path when you add a policy to configure SureConnect.

The following files are extracted:
- Alternate content files (progressbar.htm, alternatepage.htm, and barandpage.htm)
- System-Logo.gif
- Customer-Logo.gif
- Sample.gif
- README.txt.

Installing on UNIX

This section describes how to install SureConnect alternate content on a UNIX server. The following are the prerequisites:
- The UNIX server is running the Apache server.
- The shell with the # prompt is in use.
- Apache is installed in the default location.
- The sc_xx.tar file is downloaded from the organization's Web site into the /var/ftp/incoming directory.

To install SureConnect

1. At the command prompt, navigate to the htdocs directory:
   cd /usr/local/apache/htdocs
2. Type the following command:
   tar xvpf/var/ftp/incoming/sc_xx.tar

The output from the .tar file is displayed. A /Citrix NetScaler system directory is created under the specified path and the SureConnect files are installed.

Installing on Windows

Updated: 2013-09-03

This section describes how to install SureConnect alternate content on a Windows server. The following are the prerequisites:
- The server is running the Microsoft Internet Information Server.
- The DOS prompt is being used.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- The SureConnect zip (self-extracting) file is downloaded from the organization Web site using FTP into the C:\inetpub\wwwroot directory.

To install SureConnect on Windows

Do one of the following:

- At the command prompt, navigate to the wwwroot directory:
  cd c:\inetpub\wwwroot
- Type the name of the executable file:
  sc_xx.exe
- Double-click the sc_xx.exe icon from the Microsoft Windows Explorer Web browser, extract from the compressed file into the default path (for example, the c:\inetpub\wwwroot directory).

Output from the zip file is displayed. A /Citrix NetScaler system directory is created under the specified path, and the SureConnect files are installed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring SureConnect

Sep 03, 2013

The following topics describe how to configure SureConnect for scenarios involving alternate server failure.

- "Configuring the Response for Alternate Server Failure"
- "Configuring the SureConnect Policies"
- "Customizing the Alternate Content File"
- "Configuring SureConnect for Citrix NetScaler Features"

Configuring the Response for Alternate Server Failure

Updated: 2013-09-03

If the alternate server fails, and the primary server cannot immediately deliver the requested content to the client, SureConnect does not display alternate content from the failed alternate server in the client Web browser.

The Citrix NetScaler appliance automatically sends a response to the client browser. You can customize the server response to display information suited to your needs.

The default response is:

Your Request is being processed… Estimated Time: _____ Secs

Customizing the Default Response

The NetScaler appliance automatically sends the response to the client if the alternate server fails, or if the appliance is configured to send the default response.

To customize the default response of the appliance, create a vsr.htm file (a sample is provided in this section) as follows:

- The file can contain any valid HTML statements other than embedded objects.
- The file size cannot exceed 800 bytes.
- The file must reside on the NetScaler appliance. If you have a high availability (HA) setup, the file must reside on the primary and secondary nodes. Any changes made to the file on the primary node must also be applied to the file on the secondary node.
- Put vsr.htm file in the /etc directory.

To customize the default response

Change any of the contents between the </HEAD> and </HTML> tags in the vsr.htm file. Following is the sample content from vsr.htm file. The sections that you can edit are in bold text.

```
HTTP/1.1 200 OK
Server: NS_WS3.0
Content-Type: text/html
Cache-control: no-cache
Pragma: no-cache
Set-Cookie: NSC_BPIP=@@SID@@; path=/
<HTML> <HEAD> <META HTTP-EQUIV="Refresh" CONTENT="0">
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

</HEAD> **Your request is being processed...**
**
Estimated Delay: @@DELAY@@ Sec ** </HTML>
Note: Include @@DELAY@@ to display the predicted delayed response time in seconds.

SureConnect with In-Memory response (NS action)

Updated: 2013-09-03

When defining the SureConnect policy by using the add sc policy command, you can configure the NetScaler Appliance to serve alternative content to the client.

To enable SureConnect and configure the in-memory response, perform the following tasks:

- Enable the SureConnect feature on the appliance by using the enable feature SC command
- Define the services by using the add service <servicename> <IP address> <servicetype> <port> command. This identifies the original server for which the SureConnect is configured and the types of services.
- Add a SureConnect policy by using the add sc policy command. You can configure a URL-based policy or a rule-based policy. The incoming requests are validated against the URL or rule you specify in the policy.

Note: You can configure the SureConnect feature on a load balancing virtual server. In that case, perform the following additional actions:

- Enable Load Balancing by using the enable feature LB command.
- Enable SureConnect feature on the virtual server by using the set lb vserver <vservername> -sc ON command.
- Bind services to the virtual server by using the bind lb vserver <name> <serviceName> command.
- Bind policies to the virtual server by using the bind lb vserver <name> -policyname <name> command.

The following example illustrates how to configure SureConnect for the load balancing feature so that SureConnect will display alternative content from the NetScaler appliance.

In this example, two physical servers, with IP addresses, 10.101.3.187 and 10.101.3.188 are load balanced by the NetScaler appliance. The appliance has one configured virtual server, vs-NSact, whose IP address is 10.101.3.201. The file that contains the alternative content is vsr.htm. It is copied from the file system into system memory. Services are loaded until the SureConnect policy triggers, and the appliance supplies the alternate content.

```
enable feature SC LB
add service psvc1 10.101.3.187 http 80
add service psvc2 10.101.3.188 http 80
add lb vserver vs-NSact HTTP 10.101.3.201 80
bind lb vserver vs-NSact psvc1
bind lb vserver vs-NSact psvc2
add sc policy policyNS -url /cgi-bin/*.cgi -delay 400000
-action NS
set sc parameter -vsr /nsconfig/ssl/vsr.htm
bind lb vserver vs-NSact -policyName policyNS
set lb vserver vs-NSact -sc ON
save config
```

Table 1. Parameter values used in this example

Service	

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	psvc1, psvc2
Server	10.101.3.187, 10.101.3.188
Protocol	HTTP
Port	80
Load Balancing Virtual Server	
Name	vs-NSact
IP Address	10.101.3.201
Protocol	HTTP
Port	80
SureConnect Policy	
Name	policyNS
URL	/cgi-bin/*.cgi
Delay(microseconds)	400000
SC Parameter	
VSR File Name	vsr.htm

To configure this example by using the configuration utility

1. In the In the navigation pane, navigate to System > Settings. In the Modes and Features pane, perform the following actions:
   1. Click Configure Basic Features, select Load Balancing, and Click Go.
   2. Click Configure Advanced Features, select SureConnect, and Click Go.
2. In the navigation pane, navigate to Security > Protection Features > SureConnect. In the details pane, click Parameters. In the Configure SureConnect Parameters window, browse and select the VSR filename.
3. Navigate to Traffic Management > Load Balancing > Services. In the details pane, click Add. In the **Create Services** window, enter the paramter values as shown in Table 5-1, and click **OK**.
4. Navigate to Traffic Management > Load Balancing > Virtual servers. In the details pane, click Add. In the Create Virtual

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Server (Load Balancing) dialog box, enter the values shown in Table 5.1 for the Load Balancing Virtual Server parameters and click OK.

5. In the navigation pane, navigate to Traffic Management > Load Balancing > Virtual servers. Select the virtual server vs-NSact and click Open in the details pane. The Configure Virtual system (Load Balancing) dialog box, displays the list of configured services. Select services psvc1 and psvc2 and click OK.
6. In the navigation pane, expand Security > Protection Features > SureConnect. In the details pane, click Add. Create the policy with the values as given in the parameters table.
7. In the navigation pane, navigate to Traffic Management > Load Balancing > Virtual servers. Select the virtual server vs-NSact and click Open in the details pane. In the Configure Virtual system (Load Balancing) dialog box, click the Policies tab. Click **>>** to expand the features. Select **SureConnect**. When the list of SureConnect polices appear, select policyNS and click OK.
8. In the navigation pane, navigate to Traffic Management > Load Balancing > Virtual servers. Select the virtual server vs-NSact and click Open in the details pane. In the Configure Virtual system (Load Balancing) dialog box, on the Advanced tab, select SC and click OK.

Configuring the SureConnect Policies

You can configure the following SureConnect policies. The NetScaler appliance matches incoming requests in the order the policies are configured:

- Exact URL-based policies
- Wildcard rule-based policies

Configuring Exact URL Based Policies

Updated: 2013-09-03

When you configure an exact URL based policy, the NetScaler appliance matches the incoming request against the URL that has been configured in the policy. URL based policies take precedence over rule based policies.

To configure an exact URL based policy by using the command line interface

At the command prompt, type:

add sc policy <name> [-url <URL> | -rule <expression>] [-delay <usecs>] [-maxConn <positive_integer>] [-action (ACS <altContentSvcName> <altContentPath>) | NS | NOACTION)]

To configure an exact URL based policy by using the configuration utility

1. Navigate to Security > Protection Features > SureConnect.
2. In the details pane, click Add.
3. In the Create SureConnect Policy dialog box, set the following parameters:
   - Name*
   - URL (Make sure that the URL check box is selected)
   - Value*
   - Delay (microseconds)*
   - Maximum Client Connections
   - Action (Select from the Choose Action list.)
   - Alternate Service Name (if you select ACS as the Action)
   - Alternate Content Path (if you select ACS as the Action)

   *A required parameter
4. Click Create, and click Close. The URL based policy appears in the right pane, and a message displays in the status bar

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

that the policy is successfully configured.

Configuring Wildcard Rule-Based Policies

Updated: 2013-09-03

SureConnect matches the incoming requests to a defined rule, if you configure a rule-based policy.
To configure a SureConnect policy based on a wildcard rule by using the command line interface

1. Create the expression(s).
   Use the add expression command to create each expression.

2. Create the rule(s).

   Use the add sc policy command with the -rule expression_logic argument to specify the rule(s). In the -rule expression_logic argument, refer to the expression(s) you created in step 1.

   Repeat this command to create and name each rule.

The following example creates a rule “rule = = /*.cgi”:

```
add vserver vs-lb http 1.1.1.1 80
add expression expr1 url == /cgi-bin/*.cgi
add expression expr2 url == /index.html
add sc policy surecpolicy1 -rule (expr1||expr2) -delay 1000000 -action NS
bind lb vserver vs-lb -policyName surecpolicy1
```

To complete the SureConnect configuration, you will need to enter additional commands, beyond those shown in the example.

To configure a wildcard rule-based policy by using the configuration utility

1. Navigate to Security > Protection Features > SureConnect.
2. In the details pane, click Add.
3. In the Create SureConnect Policy dialog box, in the Name text box, type the name of the policy.
4. Under What to Monitor, click Expression, and then click Configure.
5. In the Create Expression dialog box, click Add.
6. In the Add Expression dialog box, enter an expression. For example, you can select an Expression Type of General, a Flow Type of REQ, a Protocol of HTTP, a Qualifier of URLQUERY, an Operator of CONTAINS, and in the Value text box, type AA. For more information about expressions, see "Policies and Expressions."
7. Click OK, and click Close.
8. In the Create Expression dialog box, click Create.

Examples of wildcard rules:

“/sports/*” matches all URLs under /sports

“/sports*” matches all URLs whose prefix matches “/sports”, starting at the beginning of the URL.

“/*.jsp” matches all URLs whose file extension is “jsp”

When configuring rule-based policies, first add the more specific rule-based policies, before adding more generic rules (for

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

example, add /cgi-bin/sports*.cgi before adding /cgi-bin/*.cgi).

Displaying the Configured SureConnect Policy

To view the SureConnect policy that you have configured, at the NetScaler command prompt, enter the show sc policy command.

Customizing the Alternate Content File

When SureConnect activates, it can display alternate content from one of the following files that you have configured:

- **progressbar.htm.** Displays the progress information.
- **alternatepage.htm.** Displays an alternate page.
- **barandpage.htm.** Displays both the progress information and an alternate page.

The alternate content files are JavaScript files. During SureConnect installation, these files are copied onto the server that contains the alternate content. These files can contain alternate content (including an alternate page) or references to other files that contain the alternate content.

This section describes the changes you can make to the alternate content file provided by the appliance.

```
//**** DEFINE YOUR VALUES HERE ****
var alt_url = "/Citrix NetScaler system /sample.gif";
var alt_url = "http://www.DomainName.com";
var Citrix NetScaler system _logo = "netscaler_logo.gif";
var our_logo = "netscaler_logo.gif";
var height = 450;
var width = 550;
var top = 200;
var left = 200;
var popunder = "no"; //specify yes for pop-under & no for pop-up
var shift_focus = "yes" //if you want to send pop-up to background on getting primary content else specify no
//**** YOUR DEFINITIONS ENDS HERE ****
```

You can make these changes:

- **var alt_url.** Specify the URL for the alternate content if a file provides the alternate content. For example:
  var alt_url = “/Citrix NetScaler system/sports.htm”
  Note: The alternate content file must be present in the /Citrix NetScaler system directory under the documents root of the Web server.
- **var our_logo.** Specify the image file of your organization logo.
- **var height.** Specify the height of the SureConnect window.
- **var width.** Specify the width of the SureConnect window.
- **var top and var left.** Specify the position of the SureConnect window.
- **var popunder.** Specifies the position of the alternate content window. Specify the value as NO to place the alternate content window above the original window. Specify the value as YES to place the alternate content window beneath the original window.
- **var shift_focus.** Specify the focus of the alternate content window. YES places the pop-up window in the background when getting the primary content. NO always keeps the pop-up window in focus, even when getting the primary content.

Note: For more information, see the README.txt file provided by the appliance with other alternate content files.

Configuring SureConnect for Citrix NetScaler Features

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updated: 2013-09-03

This section describes how SureConnect works in combination with the load balancing, content switching, cache redirection, and high availability features of the NetScaler appliance.

Configuring SureConnect for Load Balancing

You can use SureConnect in environments where the primary servers use the load balancing feature, with or without alternate servers. If the load balancing virtual server configured for SureConnect fails, the backup virtual server (if there is one) handles the traffic. Backup virtual servers do not support SureConnect policies.

Note: For information about load balancing, see "Load Balancing."

Configuring SureConnect for Cache Redirection

You can use SureConnect in environments where cache redirection is configured. The primary server is a load balancing virtual server bound to the cache redirection virtual server. Regardless of any rules configured for the cache redirection feature:

- You can configure any URL for SureConnect.
- Once SureConnect is activated for a client, requests from the client are always sent to the origin server.

Configuring SureConnect for High Availability

SureConnect is compatible with NetScaler appliances operating in high availability mode.

Note: If the optional vsr.htm file is used, it must be present in both nodes (primary and secondary) and must use the same name and directory.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Activating SureConnect

Sep 03, 2013

You can set the Citrix NetScaler appliance to activate SureConnect if either of two criteria match. Both criteria are arguments to the add sc policy command, as described here:

- -delay <microseconds>

  The first time the client requests the URL, the appliance records how long the server takes to respond. The appliance will not activate SureConnect until the second time the URL is requested. The first and second requests may be from the same or different clients.

  If you set -delay argument, SureConnect will be activated the second time the delay reaches the threshold you set.

- -maxConn <positive_integer>

  When the appliance receives a request, it checks the number of connections to the server for the configured URL. SureConnect is activated if the number of connections is greater than or equal to the value that you set for the -maxConn argument.

  If you will be providing alternate content to be displayed in the client's Web browser, you should configure the -action argument of the add sc policy command. This specifies for the NetScaler appliance whether the alternate content is coming from a dedicated alternate server (-action ACS) or the appliance (-action NS).

  When SureConnect is activated by the –maxConn argument, the SureConnect window and progress bar are displayed in the client's browser (with an alternate page, if configured).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SureConnect Environments

Sep 03, 2013

The following topics describe SureConnect environments.

- "Primary and Alternate Servers"
- "Configuration Checklist"
- "Example Configurations"

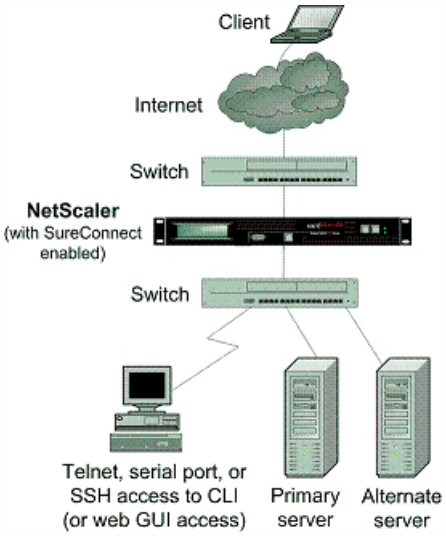
Primary and Alternate Servers

The SureConnect environment uses a dedicated server to provide alternate content when the requested content is not available. The alternate content may include an alternate page, plus optional components such as frame set, organization logo, and so on. The alternate and primary servers can be the same server.

You can configure SureConnect to display a progress bar when the requested content is not available (or the progress bar and an alternate page).

The following figure illustrates the SureConnect environment.

Figure 1. SureConnect - Primary and Alternate Servers



Configuration Checklist

Updated: 2013-09-03

Complete the following checklist before you start configuration:

Table 1. Configuration Checklist

▫	The same builds are running for the appliance and for the SureConnect files as suggested by appliance staff. Appliance Build Number: _____ SureConnect (sc_xx.exe) Build Number: _____

© 1999-2017 Citrix Systems, Inc. All rights reserved.

▫	The latest SureConnect files (style files) are extracted to: ● All primary servers (required for NS action). ● The alternate content server (required for ACS action).
▫	All customizations to the latest style and vsr.htm files are applied.
▫	The alternate content server is accessible from the Internet (required for ACS action).
▫	If the -redirectURL URL argument of the add vserver CLI command needs to be specified: ● The URL is up and running. ● This URL is not on the configured servers. ● This URL does not match any content in the vserver (that is, do not redirect a missing URL to itself). Redirecting a missing URL to itself can send some browsers into an infinite loop.
▫	All URLs to be configured for SureConnect are top-level URLs only. (Only the URLs that occupy the whole window or frame can be configured, not the embedded objects).

Following are the steps to configure SureConnect in a setup with a primary server and a dedicated alternate server:

- Enable the SureConnect feature
- Add the SureConnect policy
- Bind the SureConnect policy

You can optionally configure the following:

- Redirect the client to another URL if the primary server fails, or send a customized response to the client if the alternate server fails.
- If the servers do not provide alternate content, send a default or customized response.

To redirect the client to another URL

1. Enable the SureConnect feature.
2. Define the primary server and its service.

   You must identify the original server for which SureConnect support is being configured. At the NetScaler command prompt, type the following command:

   add service <serviceName> <IP> HTTP <port>
   where <serviceName> assigns a name for the service; <IP> is the server's IP address; and <port> is the port number that the service will use.

   Repeat use of the add service CLI command for each service that is to be added.

   You can also configure SureConnect on a load balancing virtual server. At the NetScaler command prompt, type the following command:

   add vserver <name> HTTP <IP> <port>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

3. Define and bind the SureConnect policy as follows. If you are configuring a rule-based policy, perform this step as described in "Configuring Wildcard Rule-Based Policies." To configure a URL-based policy, at the NetScaler command prompt, type the following command:

   add sc policy <name> [-url <URL>] [-delay <microsec>] [-maxConn <positiveInteger>]

   For a detailed description of the add sc policy command, see "Command Reference."

   To bind the SureConnect policy, at the NetScaler command prompt, type the following command:

   bind service <serviceName> -policyname <string>
   where <serviceName> is the name of the service defined in step 2, and <string> is the name of the SureConnect policy.

   Repeat the bind service command for each policy created.

   You must include the alternate content page in the altContSvcName argument, and in the altContPath argument of the add sc policy command.

   In the following example, the name of the alternate content file is /Citrix NetScaler system /barandpage.htm, and this file resides in svc2.

4. To save the configuration, at the NetScaler command prompt, type the following command:
   save config

Example Configurations

Updated: 2013-09-03

The following examples illustrate various SureConnect configurations.

The examples assume that monitoring of physical services is enabled. If the alternate system is down, SureConnect will deliver the alternate content from the system itself.

Example 1 - SureConnect Progress Bar and Alternate Page

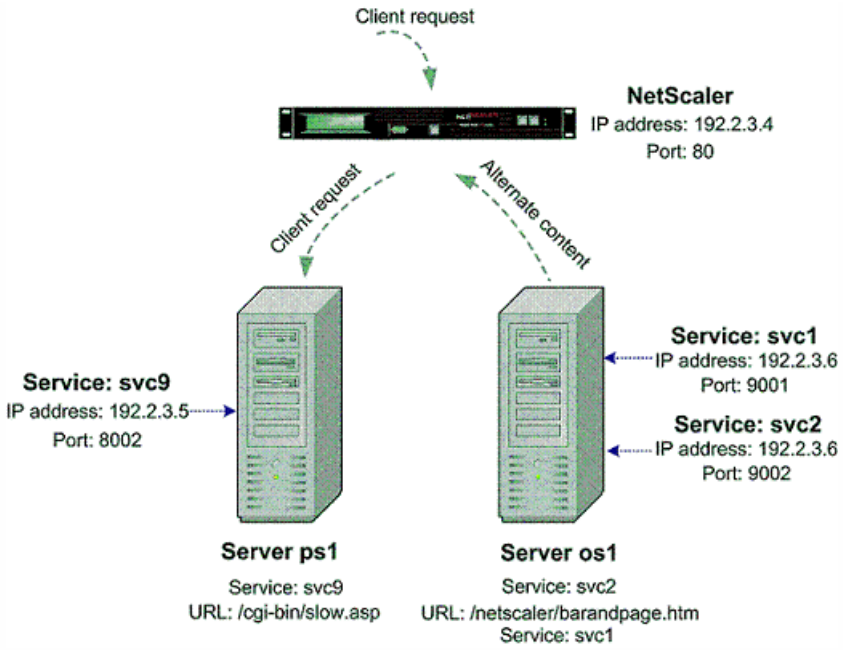
You can configure SureConnect to display both the progress bar and an alternate page to the user.

To bind a SureConnect policy to a load balancing virtual server, at the command prompt, type the following commands:

bind lb vserver <virtualServerName> -policyName <string>
where <virutalServerName> is the name of the load balancing virtual server defined in step 2 of the configuration process, and <string> is the name of the SureConnect policy defined in step 3.

Figure 2. SureConnect Configuration - Example 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.



At the NetScaler command prompt, type the following commands:

```
enable feature SC
show ns info
add service svc2 192.2.3.6 HTTP 9002
show server
show service svc2
add service svc9 192.2.3.5 HTTP 8002
add sc policy policy8 -url /cgi-bin/slow.asp
-delay 3000000 -action ACS svc2 /NetScaler 9000 system barandpage.htm
bind service svc9 -policyname policy8
set service svc9 -sc ON
save config
```

After you configure SureConnect, you can enter commands that show information to verify what you have configured.

Example 2 - SureConnect Progress Bar Only

In this example, SureConnect will display only the progress bar. The server orgsrvr with IP address 10.101.8.187 has service orgsvc. This server is connected to the appliance. The service is bound to the appliance. The progressbar.htm file specifies that only the progress bar will be displayed.

At the NetScaler command prompt, type the following commands:

```
enable feature SC
add service orgsvc 10.101.3.187 HTTP 80
add sc policy policy9 -url /cgi-bin/slow.asp
-delay 4000000 -action ACS orgsvc /NetScaler 9000 system / progressbar.htm
bind service orgsvc -policyname policy9
set service orgsvc -sc ON
save config
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Example 3 - SureConnect with Load Balancing

This example illustrates how to configure the load balancing feature so that SureConnect will display alternate contents from the primary server. For information about load balancing, see "Load Balancing."

In this example, two physical servers with IP 10.101.3.187 and 10.101.3.188 are being load balanced by the appliance. The name and location of the alternate page file is specified in the file alternatepage.htm, which resides on both servers.

The appliance has one configured virtual server address: 10.101.3.201. At the NetScaler command prompt, type the following commands:

```
enable feature SC LB
add service psvc1 10.101.3.187 HTTP 80
add service psvc2 10.101.3.188 HTTP 80
add vserver vs-SureC HTTP 10.101.3.201 80
bind lb vserver vs-SureC psvc1
bind lb vserver vs-SureC psvc2
add sc policy policy9 -url /cgi-bin/slow.asp -delay 4000000
-action ACS vs-SureC /NetScaler system /alternatepage.htm
bind lb vserver vs-SureC -policyName policy9
set lb vserver vs-SureC -sc ON
save config
```

Example 4 - SureConnect with Load Balancing (ACS Action)

This example illustrates how to configure the NetScaler appliance load balancing feature so that SureConnect will display alternate content from the alternate server. For information about load balancing, see "Load Balancing."

In this case, there are two physical servers, IP 10.101.3.187 and 10.101.3.188. Both are being load balanced by the appliance.

The name and location of the alternate page file are specified in file barandpage.htm, which resides on a third server not being load balanced.

The third server's IP address is 10.101.3.189. Because barandpage.htm is specified, the progress bar and alternate page will both be displayed.

The appliance has one configured virtual server "vsvr" whose IP address (Virtual Server) is 10.101.3.200.

At the NetScaler command prompt, type the following commands:

```
enable feature SC LB
add service psvc1 10.101.3.187 HTTP 80
add service psvc2 10.101.3.188 HTTP 80
add service alt-cont-svc 10.101.3.189 HTTP 80
add vserver vsvr HTTP 10.101.3.200 80
bind lb vserver vsvr psvc1
bind lb vserver vsvr psvc2
add sc policy policy10 -url /cgi-bin/slow.asp
-delay 4000000 -action ACS alt-cont-svc
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

/NetScaler 9000 system /barandpage.htm
bind lb vserver vsvr -policyName policy10
set lb vserver vsvr -sc ON
save config

Example 5 - SureConnect with Content Switching

This example illustrates how to configure SureConnect where the NetScaler content switching and load balancing features are being used. SureConnect is configured on a load balancing virtual server bound to a content switching virtual server.

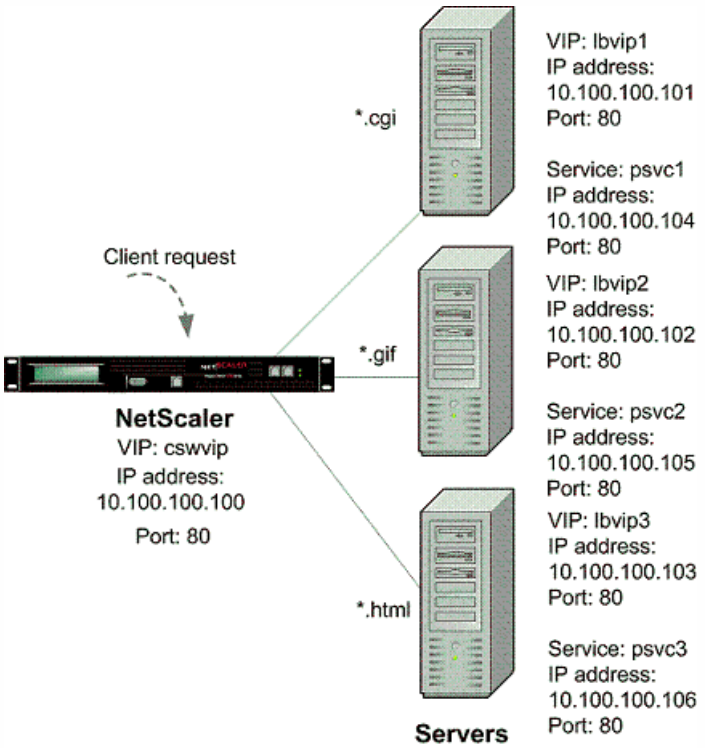
The alternate content is distributed under the content switching virtual server according to the content switching rules. For more information about load balancing and content switching, see "Load Balancing" and "Content Switching."

In this case, three physical services with IP addresses 10.100.100.104, 10.100.100.105, and 10.100.100.106 are bound to three load balancing virtual servers with IP addresses 10.100.100.101, 10.100.100.102, and 10.100.100.103. These three load balancing virtual servers are bound to a content switching virtual server with IP address 10.100.100.100.

In this setup, lbvip1 contains .cgi content, lbvip2 contains .gif content, and .lbvip3 contains .html content.

The name and location of the alternate page file is specified in the file alternatepage.htm, which resides on lbvip3. The embedded objects in this file must be distributed according to the content switching rules (any embedded gif will reside on lbvip2, any embedded htm will reside on lbvip3, and so on).

Figure 3. SureConnect Configuration - Example 5



At the NetScaler command prompt, type the following commands:

enable feature CS LB SC
add vserver cswvip HTTP 10.100.100.100 80 -type CONTENT
add vserver lbvip1 HTTP 10.100.100.101 80 -type ADDRESS

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
add vserver lbvip2 HTTP 10.100.100.102 80 -type ADDRESS
add vserver lbvip3 HTTP 10.100.100.103 80 -type ADDRESS
add service psvc1 10.100.100.104 HTTP 80
add service psvc2 10.100.100.105 HTTP 80
add service psvc3 10.100.100.106 HTTP 80
bind lb vserver lbvip1 psvc1
bind lb vserver lbvip2 psvc2
bind lb vserver lbvip3 psvc3
add cs policy CSWpolicy1 -url /*.cgi
bind cs vserver cswvip lbvip1 -policyName CSWpolicy1
add cs policy CSWpolicy2 -url /*.gif
bind cs vserver cswvip lbvip2 -policyName CSWpolicy2
add cs policy CSWpolicy3 -url /*.htm
bind cs vserver cswvip lbvip3 -policyName CSWpolicy3
add sc policy SCpol -url /cgi-bin/delay.cgi -delay 4000000 -action ACS cswvip /alternatepage.htm
bind lb vserver lbvip1 -policyName SCpol
set lb vserver lbvip1 -sc ON
save config
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Surge Protection

Mar 21, 2012

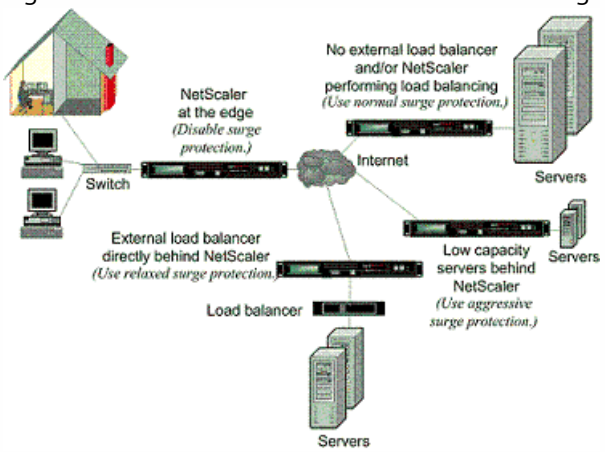
When a surge in client requests overloads a server, server response becomes slow, and the server is unable to respond to new requests. The Surge Protection feature ensures that connections to the server occur at a rate that the server can handle. The response rate depends on how surge protection is configured. The NetScaler appliance also tracks the number of connections to the server, and uses that information to adjust the rate at which it opens new server connections.

Surge protection is enabled by default. If you do not want to use surge protection, as will be the case with some special configurations, you must disable it.

The default surge protection settings are sufficient for most uses, but you can configure surge protection to tune it for your needs. First, you can set the throttle value to tell it how aggressively to manage connection attempts. Second you can set the base threshold value to control the maximum number of concurrent connections that the NetScaler appliance will allow before triggering surge protection. (The default base threshold value is set by the throttle value, but after setting the throttle value you can change it to any number you want.)

The following figure illustrates how surge protection is configured to handle traffic to a Web site.

Figure 1. A Functional Illustration of NetScaler Surge Protection



Note: If the NetScaler appliance is installed at the edge of the network, where it interacts with network devices on the client side of the Internet, the surge protection feature must be disabled. Surge protection must also be disabled if you enable USIP (Using Source IP) mode on your appliance.

The following example and illustration show the request and response rates for two cases. In one case, surge protection is disabled, and in the other it is enabled.

When surge protection is disabled and a surge in requests occurs, the server accepts as many requests as it can process concurrently, and then begins to drop requests. As the server becomes more overloaded, it goes down and the response rate is reduced to zero. When the server recovers from the crash, usually several minutes later, it sends resets for all pending requests, which is abnormal behavior, and also responds to new requests with resets. The process repeats for each surge in requests. Therefore, a server that is under DDoS attack and receives multiple surges of requests can become unavailable to legitimate users.

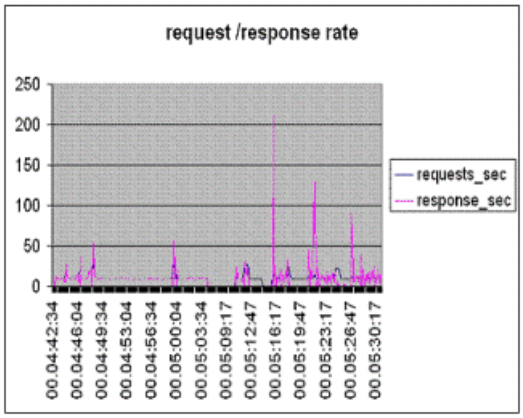
When surge protection is enabled and a surge in requests occurs, surge protection manages the rate of requests to the server, sending requests to the server only as fast as the server can handle those requests. This enables the server to respond to each request correctly in the order it was received. When the surge is over, the backlogged requests are cleared

© 1999-2017 Citrix Systems, Inc. All rights reserved.

as fast as the server can handle them, until the request rate matches the response rate.

The following figure compares the request and response scenarios when surge protection is enabled to that when it is disabled.

Figure 2. Request/Response Rate with and without Surge Protection



© 1999-2017 Citrix Systems, Inc. All rights reserved.

Disabling and Reenabling Surge Protection

Sep 03, 2013

The surge protection feature is enabled by default. When surge protection is enabled, it is active for any service that you add.

To disable or reenable surge protection by using the command line interface

At the command prompt, type one of the following sets of commands to disable or reenable surge protection and verify the configuration:

- disable ns feature SurgeProtection
- show ns feature

- enable ns feature SurgeProtection
- show ns feature

Example

```
disable ns feature SurgeProtection
 Done show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging            WL               ON
 2)   Surge Protection       SP               OFF
 .
 .
 .
 23)  HTML Injection         HTMLInjection    ON
 24)  NetScaler Push         push             OFF
 Done
 enable ns feature SurgeProtection
 Done
> show ns feature

      Feature                Acronym          Status
      -------                -------          ------
 1)   Web Logging            WL               ON
 2)   Surge Protection       SP               ON
 .
 .
 .

 23)  HTML Injection         HTMLInjection    ON
 24)  NetScaler Push         push             OFF
 Done
>
```

To disable or reenable surge protection by using the configuration utility

© 1999-2017 Citrix Systems, Inc. All rights reserved.

1. In the navigation pane, expand System, and then select Settings.
2. In the details pane, click Change Advanced Features.
3. In the Configure Advanced Features dialog box, clear the selection from the Surge Protection check box to disable the surge protection feature, or select the check box to enable the feature.
4. Click OK.
5. In the Enable/Disable Feature(s) dialog box, click Yes. A message appears in the status bar, stating that the feature has been enabled or disabled.

To disable or reenable surge protection for a particular service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services. The list of configured services is displayed in the details pane.
2. In the details pane, select the service for which you want to disable or reenable the surge protection feature, and then click Open.
3. In the Configure Service dialog box, click the Advanced tab and scroll down.
4. In the Others frame, clear the selection from the Surge Protection check box to disable the surge protection feature, or select the check box to enable the feature.
5. Click OK. A message appears in the status bar, stating that the feature has been enabled or disabled.
   Note: Surge protection works only when both the feature and the service setting are enabled.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

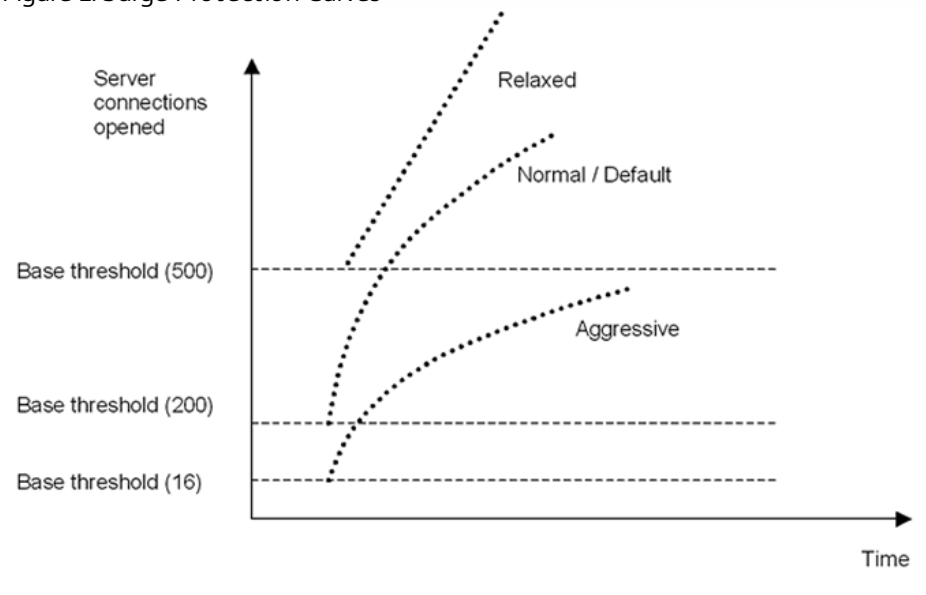
Setting Thresholds for Surge Protection

May 14, 2012

To set the rate at which the NetScaler appliance opens connections to the server, you must configure the threshold and throttle values for surge protection.

The following figure shows the surge protection curves that result from setting the throttle rate to relaxed, normal, or aggressive. Depending on the configuration of the server capacity, you can set base threshold values to generate appropriate surge protection curves.

Figure 1. Surge Protection Curves



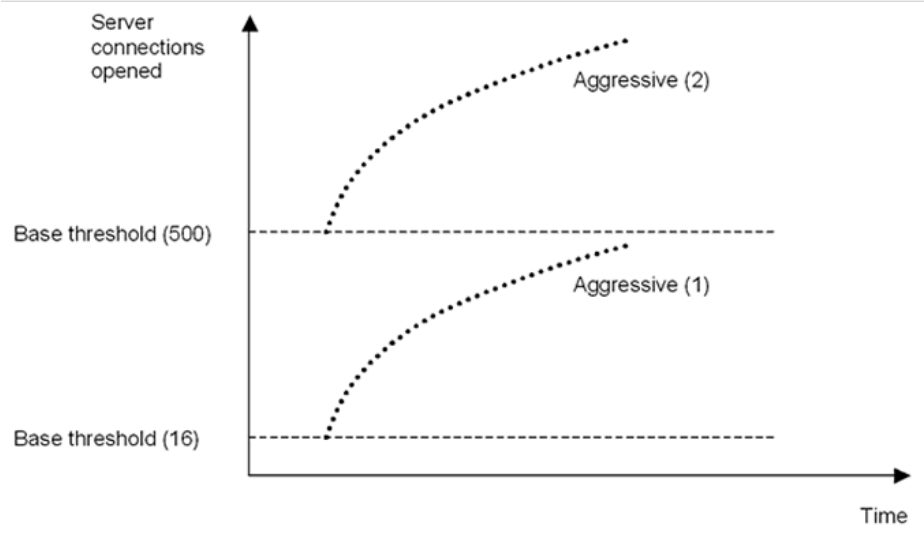
Your configuration settings affect the behavior of surge protection in the following manner:

- If you do not specify a throttle rate, it is set to normal (the default value), and the base threshold is set to 200, as shown in the preceding figure.
- If you specify a throttle rate (aggressive, normal, or relaxed) without specifying a base threshold, the curve reflects the default values of the base threshold for that throttle rate. For example, if you set the throttle rate to relaxed, the resulting curve will have the base threshold value of 500.
- If you specify only the base threshold, the entire surge protection curve shifts up or down, depending on the value you specify, as shown in the figure that follows.
- If you specify both a base threshold and a throttle rate, the resulting surge protection curve is based on the set throttle rate and adjusted according to the value set for the base threshold.

In the following figure, the lower curve (Aggressive 1) results when the throttle rate is set to aggressive but the base threshold is not set. The upper curve (Aggressive 2) results when the base threshold is set to 500, but the throttle rate is not set. The second upper curve (Aggressive 2) also results when the base threshold is set to 500, and the throttle rate is set to aggressive.

Figure 2. Aggressive Rate with the Default or a Set Base Threshold

© 1999-2017 Citrix Systems, Inc. All rights reserved.



To set the threshold for surge protection by using the configuration utility

1. In the navigation pane, expand System, and then select Settings.
2. In the details pane, click Global System Settings.
3. If you want to set a base threshold different from the default for the throttle rate, in the Configure Global Settings dialog box, Base Threshold text box, enter the maximum number of concurrent server connections allowed before surge protection is triggered. The base threshold is the maximum number of server connections that can be open before surge protection is activated. The maximum value for this setting is 32,767 server connections. The default setting for this value is controlled by the throttle rate you choose in the next step.
   Note: If you do not set an explicit value here, the default value will be used.
4. In the Throttle drop-down list, select a throttle rate. The throttle is the rate at which the NetScaler appliance allows connections to the server to be opened. The throttle can be set to the following values:
   Aggressive
   Choose this option when the connection-handling and surge-handling capacity of the server is low and the connection needs to be managed carefully. When you set the throttle to aggressive, the base threshold is set to a default value of 16, which means that surge protection is triggered whenever there are 17 or more concurrent connections to the server.
   Normal
   Choose this option when there is no external load balancer behind the NetScaler appliance or downstream. The base threshold is set to a value of 200, which means that surge protection is triggered whenever there are 201 or more concurrent connections to the server. Normal is the default throttle option.
   Relaxed
   Choose this option when the NetScaler appliance is performing load balancing between a large number of Web servers, and can therefore handle a high number of concurrent connections. The base threshold is set to a value of 500, which means that surge protection is triggered only when there are 501 or more concurrent connections to the server.
5. Click OK. A message appears in the status bar, stating that the global settings are configured.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Flushing the Surge Queue

Dec 04, 2013

When a physical server receives a surge of requests, it becomes slow to respond to the clients that are currently connected to it, which leaves users dissatisfied and disgruntled. Often, the overload also causes clients to receive error pages. To avoid such overloads, the NetScaler appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.

The appliance does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the appliance looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the appliance establishes a new connection with the server, and establishes a virtual link between client and the server. However, if the appliance cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the appliance cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The appliance maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which the appliance cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you may want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed; the other requests in the surge queue of the load balancing virtual server are not flushed.

Note: You cannot flush the surge queues of cache redirection, authentication, VPN or GSLB virtual servers or GSLB services.
Note: Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

To flush a surge queue by using the command line interface

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while executing the command, surge queue of the specified entity will be flushed. If more than one entity has the same name, the appliance flushes surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while executing the command, the appliance flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member (<serverName> and <port>) without specifying the name of the service group (<name>) and you cannot specify <port> without a <serverName>. Specify the <serverName> and <port> if you want to flush the surge queue for a specific service group member.
- If you execute the command without specifying any names, the appliance flushes the surge queues of all the entities present on the appliance.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the command prompt, type:

flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Examples

1.

flush ns surgeQ –name SVC1ANZGB –serverName 10.10.10.1 80

The above command flushes the surge queue of the service or virtual server that is named SVC1ANZGB and has IP address as 10.10.10

2.

flush ns surgeQ

The above command flushes all the surge queues on the appliance.

To flush a surge queue by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, select a virtual server and, in the Action list, select Flush Surge Queue.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NITRO API

Oct 14, 2015

The NetScaler NITRO protocol allows you to configure and monitor the NetScaler appliance programmatically by using Representational State Transfer (REST) interfaces. Therefore, NITRO applications can be developed in any programming language. Additionally, for applications that must be developed in Java or .NET or Python, NITRO APIs are exposed through relevant libraries that are packaged as separate Software Development Kits (SDKs).

> **Important**
>
> - XML API are deprecated from NetScaler 10.5 onwards.
>
> - Until specified otherwise, this NITRO documentation applies to NetScaler versions 11.0 and 10.5.

To use NITRO, you must have a basic understanding of the NetScaler appliance and you must make sure that the client application has the following:

- Access to a NetScaler appliance, version 9.2 or later.
- To use REST interfaces, you must have a system to generate HTTP or HTTPS requests (payload in JSON format) to the NetScaler appliance. You can use any programming language or tool.
- For Java clients, you must have a system where Java Development Kit (JDK) 1.5 or later is available. The JDK can be downloaded from http://www.oracle.com/technetwork/java/javase/downloads/index.html.
- For .NET clients, you must have a system with .NET framework 3.5 or later installed. The .NET framework can be downloaded from http://www.microsoft.com/downloads/en/default.aspx.
- The Python SDK is available from NetScaler 10.5 onwards. For Python clients, you must have a system with Python 2.7 or above version and the Requests library (available in <NITRO_SDK_HOME>/lib) installed. The NITRO library must be installed on the client path. For installation instructions, read the <NITRO_SDK_HOME>/README.txt file.

Obtaining the NITRO Package

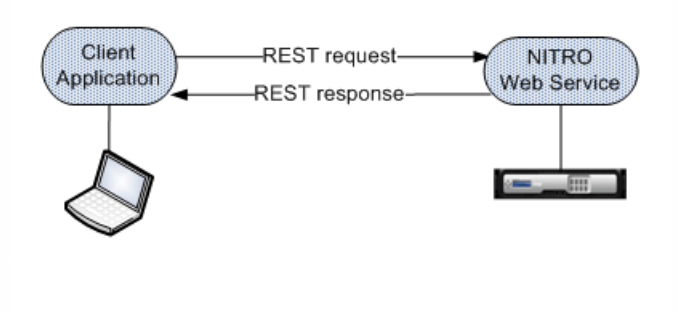
The NITRO package is available as a tar file on the **Downloads** page of the NetScaler appliance's configuration utility. You must download and un-tar the file to a folder on your local system. This folder contains the NITRO libraries in the **lib** subfolder. The libraries must be added to the client application classpath to access NITRO functionality.

Note: The REST package contains only documentation for using the REST interfaces.

How NITRO Works

The NITRO infrastructure consists of a client application and the NITRO Web service running on a NetScaler appliance. The communication between the client application and the NITRO web service is based on REST architecture using HTTP or HTTPS.

© 1999-2017 Citrix Systems, Inc. All rights reserved.



As shown in the above figure, a NITRO request is executed as follows:

1. The client application sends REST request message to the NITRO web service. When using the SDKs, an API call is translated into the appropriate REST request message.
2. The web service processes the REST request message.
3. The NITRO web service returns the corresponding REST response message to the client application. When using the SDKs, the REST response message is translated into the appropriate response for the API call.

To minimize traffic on the NetScaler network, you retrieve the whole state of a resource from the server, make modifications to the state of the resource locally, and then upload it back to the server in one network transaction. For example, to update a load balancing virtual server, you must retrieve the object, update the properties, and then upload the changed object in a single transaction.

Note: Local operations on a resource (changing its properties) do not affect its state on the server until the state of the object is explicitly uploaded.

NITRO APIs are synchronous in nature. This means that the client application waits for a response from the NITRO web service before executing another NITRO API.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

REST Web Services

Jan 27, 2016

REST (REpresentational State Transfer) is an architectural style based on simple HTTP requests and responses between the client and the server. REST is used to query or change the state of objects on the server side. In REST, the server side is modeled as a set of entities where each entity is identified by a unique URL.

For information on the NITRO SDKs, see Java, .NET, and Python API.

The general format for NITRO URLs is as follows:

- **For configurations.** http://<netscaler-ip-address>/nitro/v1/config/<resource-type>
- **For retrieving statistics.** http://<netscaler-ip-address>/nitro/v1/stat/<resource-type>

For example, for a load balancing virtual server, <resource-type> can be replaced by lbvserver.

> ## Important
>
> From NetScaler 10.1 version onwards, the following Content-Type is supported:
>
> Content-Type:application/json.
>
> However, content types such as "application/x-www-form-urlencoded" and of the form "application/vnd.com.citrix.netscaler" that were supported in NetScaler 9.3 and earlier versions can also be used. You must make sure that the payload is the same as used in earlier versions.
>
> The payloads provided in this documentation are applicable only for the "application/json" Content-Type.

Some points to remember:

- In addition to the CRUD operations (Create, Read, Update, and Delete), resources (such as lbvserver) can support other operations or actions. These operations use the HTTP POST method, with the URL specifying the operation to be performed and the request body specifying the parameters for that operation.
- All NITRO operations are logged in the /var/log/nitro.log file on the NetScaler appliance.

For more information on the REST objects and the usage, see the documentation provided in the <NITRO_SDK_HOME>/index.html file.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Connecting to the NetScaler Appliance

Feb 09, 2016

The first step towards using NITRO is to establish a session with the NetScaler appliance and then authenticate the session by using the NetScaler administrator's credentials. You must specify the username and password in the login object. The session ID that is created must be specified in the request header of all further operations in the session.

Note:

- You must have a user account on the appliance to log on to it. The configuration operations that you can perform are limited by the administrative roles assigned to your account.
- To ensure secure communication, use the HTTPS protocol in NITRO requests.
- Instead of creating a NITRO session, you can logon to the NetScaler appliance while performing individual operations. To do this, you must specify the username and password in the **request header** of the NITRO request as follows:
  X-NITRO-USER:<username>
  X-NITRO-PASS:<password>
  Content-Type:application/json

For example, to connect and create a session with a NetScaler appliance with NSIP address 10.102.29.60 by using the HTTP protocol:

- **Request:**

  HTTP Method

  POST

  URL

  http://10.102.29.60/nitro/v1/config/login

  Request Headers

  Content-Type:application/json

  Request Payload

```
{
   "login":
   {
      "username":"<username>",
      "password":"<password>"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  201 Created

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

  Response Header

  Set-Cookie:
  NITRO_AUTH_TOKEN=<tokenvalue>;

© 1999-2017 Citrix Systems, Inc. All rights reserved.

path=/nitro/v1

Modifying the Session Timeout

You can modify the timeout period by specifying a new timeout period (in seconds) in the login object. For example, to modify the timeout period to 60 minutes:

```
{
   "login":
   {
      "username":"<username>",
      "password":"<password>",
      "timeout":"3600"
   }
}
```

Some points to note with regards to session timeout for NetScaler 10.5 and later versions:

- When restricted timeout param is enabled, NITRO, by default, uses the timeout value that is configured for the logged in user. You can customize this value but it must be limited to the value specified for the user. If no value is specified for the user, the default timeout value of 15 minutes is used.
- When restricted timeout param is not enabled, NITRO uses the default value of 30 minutes as session timeout.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling NetScaler Features and Modes

Apr 10, 2015

Some NetScaler features and modes are disabled by default and therefore must be enabled before they can be configured. To enable a NetScaler feature or mode, specify the action as "enable" in the URL query string, and in the request payload, specify the feature or mode to be enabled.

For example, to enable the load balancing and content switching features:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/nsfeature?action=enable

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
   "nsfeature":
   {
      "feature":
      [
         "LB",
         "CS"
      ]
   }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

For example, to enable the L2 and fast ramp modes:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/nsmode?action=enable

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Content-Type:application/json

Request Payload

```
{
  "nsmode":
  {
    "mode":
    [
      "L2",
      "FR"
    ]
  }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Note: To disable a feature or mode, in the URL query string, specify the action as "disable".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Saving NetScaler Configurations

Apr 10, 2015

To make sure that the configurations persist on rebooting the appliance, you must save the NetScaler configurations. To save the configurations, specify the action as "save" in the URL query string.

To save the configurations:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/nsconfig?action=save

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

  ```
  {
    "nsconfig":
    {}
  }
  ```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

General API Usage

Apr 01, 2015

A NetScaler appliance has multiple features, and each feature has multiple resources. Each NetScaler resource, depending on the operation to be performed on it, has a unique URL associated with it. URLs for configuration operations have the following format:

http://<netscaler-ip-address>/nitro/v1/config/<resource_type>/<resource_name>.

For example, to access the lbvserver named MyFirstLbVServer on a NetScaler with IP 10.102.29.60, the URL is:

http://10.102.29.60/nitro/v1/config/lbvserver/MyFirstLbVServer.

This section explains, in general, the different types of operations that can be performed on the NetScaler appliance by using NITRO API.

Before going into the details of these operations, you must be aware of the following functionality:

- You can use the "action" query parameter in the URL to perform operations such as save, enable, disable, and kill sessions.
- You can use the "attrs", "filter", "args", "count" and other query parameters when retrieving NetScaler resources. For more information, see Retrieving Details of NetScaler Resources.
- You can perform bulk operations and thus minimize network traffic. To perform a bulk operation, specify the required parameters in the same request payload. You can also control the behavior of a bulk operation in case of failure of some operations. To do this, in the request header, specify the appropriate value for the X-NITRO-ONERROR parameter. For more information, see Performing Bulk Operations.
- NetScaler NITRO clearly shows the errors so that corrective actions can be taken. For more information, see Error Handling.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding a NetScaler Resource

Apr 10, 2015

To create a new resource (for example, an lbvserver) on the appliance, specify the resource name and other related arguments in the specific resource object.

For example, to create an load balancing virtual server named MyFirstLbVServer:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

  ```
  {
    "lbvserver":
    {
      "name":"MyFirstLbVServer",
      "servicetype":"http"
    }
  }
  ```

- **Response:**

  HTTP Status Code on Success

  200 Created

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling a NetScaler Resource

Apr 10, 2015

To enable a resource on the NetScaler appliance, specify the action as "enable" in the URL query string, and in the request payload, specify the resource to be enabled.

For example, to enable a load balancing virtual server named MyFirstLbVServer:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver?action=enable

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

  ```
  {
    "lbvserver":
    {
      "name":"MyFirstLbVServer"
    }
  }
  ```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Note: To disable a resource, in the URL query string, specify the action as "disable".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Getting the Count of NetScaler Resources

Apr 10, 2015

To get a count of a specific resource type, in the URL specify the count query parameter as "yes".

For example, to get a count of all the load balancing virtual servers:

- **Request:**

  HTTP Method

  GET

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver?count=yes

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Accept:application/json

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

  Response Header

  Content-Type:application/json

  Response Payload

```
{
  "lbvserver":
  [
    {
      "__count": 4
    }
  ]
}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Retrieving Details of NetScaler Resources

Apr 10, 2015

NITRO provides multiple approaches using which you can retrieve resources and their relevant details. The following table explains each of these approaches with the required URL.

Note: A sample format of the request and response is provided below the table.

Retrieving all details of all resources of a specific type	In the URL, specify the type of resource for which you want to retrieve the details. For example, to retrieve all details of load balancing virtual servers available on a NetScaler appliance. http://<netscaler-ip-address>/nitro/v1/config/lbvserver
Retrieving all details of a specific resource	In the URL, specify the name of resource for which you want to retrieve the details. For example, to retrieve all details of a load balancing virtual server named MyFirstLbVServer: http://<netscaler-ip-address>/nitro/v1/config/lbvserver/MyFirstLbVServer
Retrieving summary or detailed view of resources	In the URL, use the "view" query parameter to specify whether you want to view the summary (mandatory parameters) or detail (all parameters). For example, to view the mandatory parameters for all load balancing virtual servers: http://<netscaler-ip-address>/nitro/v1/config/lbvserver?view=summary Note: By default, the retrieved results are displayed in detail view (?view=detail).
Retrieving all details of resources that have multiple unique identifiers	In the URL, specify the type of resource and use the "args" query parameter to specify the unique attributes and the values for those attributes. For example, to get the application firewall profiles that have unique identifiers "name" and "starturl" as appfw1 and aa respectively. http://<netscaler-ip-address>/nitro/v1/config/appfwprofile? args=name:appfw1,starturl:aa
Retrieving specific details of all resources of a specific type	In the URL, specify the type of the resource and use the "attrs" query parameter to specify the resource details that you want to retrieve. For example, to retrieve the "name" and "lbmethod" of all load balancing virtual servers: http://<netscaler-ip-address>/nitro/v1/config/lbvserver?attrs=name,lbmethod
Retrieving specific details of a specific resource	In the URL, specify the type and name of the resource and use the "attrs" query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	For example, to retrieve the "name" and "lbmethod" of a load balancing virtual server named "MyFirstLbVServer": http://<netscaler-ip-address>/nitro/v1/config/lbvserver/MyFirstLbVServer?attrs=name,lbmethod
Filtering the retrieved resources	In the URL, specify the type of resource and use the "filter" query parameter to specify the attribute(s) and the value(s) of the attributes. The resources fetched will be filtered based on the filter criteria. Note: The filter query parameter supports the use of PCRE regular expressions. For example, to filter the load balancing virtual servers where the "lbmethod" is ROUNDROBIN. http://<netscaler-ip-address>/nitro/v1/config/lbvserver?filter=lbmethod:ROUNDROBIN
Retrieving resources in paginated manner	If the request is likely to result in a large number of resources, you can divide the results into pages and retrieve them page by page (paginated). For example, if you are retrieving all the 53 load balancing virtual servers of a NetScaler appliance, instead of retrieving all 53 in one response, you can configure the results to be divided into 6 pages each having 10 results. In the URL, specify the name of the resource and use the following query parameters: • "pageno" - The page number to be displayed. • "pagesize" - The number of resources to be displayed in each page. For example, to retrieve the load balancing virtual servers in a paginated form, first get a count (using the "count" query parameter shown in below row) of the load balancing virtual servers. Then, accordingly specify the number of results for each page and then specify the page number to be displayed. http://<netscaler-ip-address>/nitro/v1/config/lbvserver?pagesize=10&pageno=3

For example, to retrieve only the name and load balancing method of all load balancing virtual servers on a NetScaler:

- **Request:**

  HTTP Method

  GET

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver?attrs=name,lbmethod

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Accept:application/json

  Note: If the operation returns a list of objects, the Accept header does not require the resource type to be appended by "_list".

- **Response:**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success

200 OK

HTTP Status Code on Failure

4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Response Header

Content-Type:application/json

Note: For a single object, the Content-Type header does not require the resource type to be appended by "_list".

Response Payload

```
{
   lbvserver:
   {
      name: "test",
      lbmethod: "LEASTCONNECTION"
   }
   {
      name: "test1",
      lbmethod: "LEASTCONNECTION"
   }
}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Retrieving Statistics of NetScaler Resources

Apr 10, 2015

The NetScaler appliance collects statistics about the usage of its features and the corresponding resources. NITRO can retrieve these statistics.

- To get statistics of a feature, the URL format must be: http://<netscaler-ip-address>/nitro/v1/stat/<feature_name>.
- To get statistics of a resource, the URL format must be: http://<netscaler-ip-address>/nitro/v1/stat/<resource_type>/<resource_name>.
- To get statistics of the services and service groups that are bound to a load balancing virtual server, the URL format must be: http://<netscaler-ip-address>/nitro/v1/stat/lbvserver/<name>?statbindings=yes.

For example, to get the statistics of a load balancing virtual server named MyFirstLbVServer:

- **Request:**

HTTP Method

GET

URL

http://<netscaler-ip-address>/nitro/v1/stat/lbvserver/MyFirstLbVServer

Request Headers COPY

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Accept:application/json
```

- **Response:**

HTTP Status Code on Success

200 OK

HTTP Status Code on Failure

4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Response Payload COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{
"lbvserver":
[
{
"name":"MyFirstLbVServer",
"establishedconn":0,
"vslbhealth":0,
"primaryipaddress":"0.0.0.0",
...
}
]
}
```

> **Note**
>
> Not all NetScaler features and resources have statistic objects associated with them.

To get statistics of the bound entities, use statbindings=yes.

For example, to get the statistics of the services that are bound to a load balancing virtual server named MyFirstLbVServer:

- **Request**

HTTP Method

GET

URL

http://<netscaler-ip-address>/nitro/v1/stat/lbvserver/MyFirstLbVServer?statbindings=yes

Request Headers COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Accept:application/json
```

- **Response**

HTTP Status code on success
200 OK

HTTP Status code on failure
4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Response Header COPY

```
Content-Type:application/json
```

Response Payload COPY


```
{

  "lbvserver": [{

    "name": "MyFirstLbVServer",

    "sortorder": "descending",

    "vsvrsurgecount": "0",

    "establishedconn": "0",

     ...

     ...

    "service": [{
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "name": "s1",

        "throughput": "0",

        "throughputrate": 0,

        "avgsvrttfb": "0",

          "primaryipaddress": "1.2.3.5",

          "primaryport": 80,

          "servicetype": "HTTP",

          "state": "DOWN",

           "totalrequests": "0",

           "requestsrate": 0,

           ...



           ...

           }]

    }]

}
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Resetting Properties of a NetScaler Resource

Apr 10, 2015

To unset the value of an attribute of a NetScaler object or reset it to its default value (similar to the "unset" NetScaler CLI commands), specify the action as "unset" in the URL query string, and in the request payload, specify the value of the attributes to be unset as "true" (boolean).

For example, to unset the load balancing method and the comments attributes of a load balancer virtual server named MyFirstLbVServer:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver?action=unset

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
   "lbvserver":
   {
      "name":"MyFirstLbVServer",
      "lbmethod":"true",
      "comment":"true"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Renaming a NetScaler Resource

Apr 10, 2015

To change the name of an existing resource, specify the action as "rename" in the URL query string, and in the request payload, specify the existing name and the new name.

For example, to change the name of a load balancing virtual server from MyFirstLbVServer to MyServer:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver?action=rename

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
   "lbvserver":
   {
      "name":"MyFirstLbVServer",
      "newname":"MyServer"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updating a NetScaler Resource

Apr 10, 2015

To update the details of an existing resource on the NetScaler appliance, specify the name of the resource in the URL, and in the request payload,within the specific resource object, specify the name and the updated details of the resource.

For example, to change the load balancing method to ROUNDROBIN and update the comment property for a load balancing virtual server named MyFirstLbVServer:

- **Request:**

  HTTP Method

  PUT

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver/MyFirstLbVServer

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
   "lbvserver":
   {
      "name":"MyFirstLbVServer",
      "lbmethod":"ROUNDROBIN",
      "comment":"Updated comments"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding NetScaler Resources

Apr 15, 2015

NetScaler resources form relationships with each other through the process of binding. This is how services are associated with a load balancing virtual server, or how policies are bound to a load balancing virtual server. Each binding relationship is represented by its own object. A binding resource has properties representing the name of each NetScaler resource in the binding relationship. It can also have other properties related to that relationship (for example, the weight of the binding between an lbvserver resource and a service resource).

Read through the following examples to get a better understanding of the bind and unbind operation.

Example 1: To bind a service named "svc_prod" to a load balancing virtual server named "MyFirstLbVServer" and specify a weight for the binding:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver_service_binding

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
   "lbvserver_service_binding":
   {
      "servicename":"svc_prod",
      "weight":"20",
      "name":"MyFirstLbVServer"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  201 Created

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Example 2: To bind a policy to a policy label:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json

Request Payload

```
{
  "authenticationpolicylabel_authenticationpolicy_binding":
  {
    "policyname":"p1",
    "priority":"100",
    "gotopriorityexpression":"END",
    "labelname":"pl1"
  }
}
```

- **Response:**

  HTTP Status Code on Success

  201 Created

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

To unbind a resource, use the DELETE method and specify an "args" query string parameter in the URL that contains the attribute name and value in the relationship resource that designates the secondary resource.

For example, to unbind the service "svc_prod" from the load balancing virtual server "MyFirstLbVServer":

- **Request:**

  HTTP Method

  DELETE

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver_service_binding/MyFirstLbVServer?args=servicename:svc_prod

  Request Header

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Globally Binding NetScaler Resources

Apr 10, 2015

Some NetScaler resources can be bound globally to affect the whole system. For example, if a compression policy is bound to an load balancing virtual server, the policy affects only the traffic on that virtual server. However, if bound globally, it can affect any traffic on the NetScaler appliance regardless of the virtual server that handles the traffic.

The names of NITRO resources that can be used to bind resources globally have the pattern <featurename>global_<resourcetype>_binding. For example, the object aaaglobal_aaapreauthenticationpolicy_binding is used to bind preauthentication policies globally.

For example, to bind the policy named preautpol1 globally at priority 200:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding/preautpol1

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
  Content-Type:application/json

  Request Payload

```
{
   "aaaglobal_aaapreauthenticationpolicy_binding":
   {
      "policy":"preautpol1",
      "priority":"200"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

To unbind a global resource, in the URL use the args query parameter to specify the resource to be unbound.

For example, to unbind the policy named preautpol1:

- **Request:**

  HTTP Method

  DELETE

  URL

  http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding?args=policy:preautpol1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Header

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

- **Response:**

HTTP Status Code on Success

200 OK

HTTP Status Code on Failure

4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Deleting a NetScaler Resource

Nov 16, 2015

The usage of the delete operation depends on the unique identifiers (UIDs) of the resource being deleted.

- Deleting Resource with Single UID
- Deleting Resource with Multiple UIDs

Deleting Resource with Single UID

To delete a NetScaler resource that can be identified by a single identifier, specify the resource name in the URL.

For example, to delete a load balancing virtual server named MyFirstLbVServer:

- **Request:**

  HTTP Method

  DELETE

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver/MyFirstLbVServer

  Request Header

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Deleting Resource with Multiple UIDs

To delete a NetScaler resource that is identified using multiple identifiers, use the "args" query parameter to specify the unique attributes along with their values.

For example, consider a SNIP (10.102.29.71) that belongs to two traffic domains (TDs) 123 and 110. To delete the SNIP from one of the traffic domains, specify the IP address and the relevant TD in the URL as follows:

- **Request:**

  HTTP Method

  DELETE

  URL

  http://<netscaler-ip-address>/nitro/v1/config/nsip?args=ipaddress:10.102.29.71,td:123

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Header

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

- **Response:**

HTTP Status Code on Success

200 OK

HTTP Status Code on Failure

4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Performing Bulk Operations

Jun 21, 2017

You can create and update multiple resources simultaneously and thus minimize network traffic. For example, you can add/enable/disable multiple load balancing virtual servers in the same operation. To perform a bulk operation, specify the required parameters in the same request payload.

To account for the failure of some operations within the bulk operation, NITRO allows you to configure one of the following behaviors:

- **Exit.** When the first error is encountered, the execution stops. The NITRO operations that were executed before the error are committed. This is the default behavior.
- **Rollback.** When the first error is encountered, the execution stops. The NITRO operations that were executed before the error are rolled back. Rollback is only supported for add and bind NITRO operations.
- **Continue.** All the NITRO operations in the list are executed even if some operations fail.

You must specify the behavior of the bulk operation in the request header using the X-NITRO-ONERROR parameter.

For example, to add two load balancing virtual servers in one operation, and continue even if one of the add operation fails:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
  Content-Type:application/json

  Request Payload

```
{
  "lbvserver":
  [
    {
      "name":"new_lbvserver1",
      "servicetype":"http"
    },

    {
      "name":"new_lbvserver2",
      "servicetype":"http"
    }
  ]
}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Response:**

  HTTP Status Code on Success

  201 Created for the add operation and 200 OK for the update operation.

  HTTP Status Code on Failure

  207 Multi Status with error details in the response payload. For more information, see Error Handling.

Example for enabling multiple load balancing virtual servers in the same operation

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/lbvserver?action=enable

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
  Content-Type:application/json

  Request Payload

```
{
        "lbvserver":
   [
     {
        "name":"v1",
     },

     {
        "name":"v2",
     }
   ]
}
```

- **Response:**

  HTTP Status Code on Success

  201 Created for the add operation and 200 OK for the update operation.

  HTTP Status Code on Failure

  207 Multi Status with error details in the response payload. For more information, see Error Handling.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Performing File Operations

Sep 26, 2016

NetScaler operations such as configuring SSL certificates requires the input files to be available locally on the NetScaler appliance. NITRO allows you to perform file operations such as uploading files, retrieving files, retrieving file content, and deleting files of types.

Note:

- File size must be less than or equal to 2 MB.
- Use the "BASE64" value for the fileencoding attribute in the request payload. This is the only valid encoding currently supported.
- The filelocation path must be URL encoded. For example, if the path is /nsconfig/ssl, encode the / and use the file location as %2Fnsconfig%2Fssl.
- When uploading a file, make sure that each directory of the file path has the 755 (read, write, execute) permission. For example, to upload a file to the "/nsconfig/ssl/" directory, the following directories must have the 755 permission:
  - flash (because the "/nsconfig" folder is actually a link to "/flash/nsconfig/" directory)
  - nsconfig
  - ssl

Uploading a File

To upload a file to the NetScaler, specify a name for the file, the location where the file must be created on the NetScaler, and the content of the file.

For example, to upload a file named cert1.crt in the /nsconfig/ssl/ directory:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/systemfile

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
  "systemfile":
  {
    "filename": "cert1.crt",
    "filelocation": "/nsconfig/ssl/",
    "filecontent":"VGhpcyBpcyBteSBmaWxl",
    "fileencoding": "BASE64"
  }
}
```

- **Response:**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success

200 OK

HTTP Status Code on Failure

4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Retrieving the Files

To retrieve the files from a specific NetScaler directory, specify the directory path in the URL.

For example, to retrieve the files from the /nsconfig/ssl directory.

- **Request:**

  HTTP Method

  GET

  URL

  http://<netscaler-ip-address>/nitro/v1/config/systemfile?args=filelocation:%2Fnsconfig%2Fssl

  Request Header

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
  Accept:application/json

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

  Response Header

  Content-Type:application/json

  Response Payload


```
{
   "systemfile":
   [
      {
         "filename": "ns-root.key",
         "filelocation": "/nsconfig/ssl",
         "fileaccesstime": "Tue Jan 14 19:27:01 2014",
         "filemodifiedtime": "Tue Nov  5 17:16:00 2013"
      },
      {
         "filename": "ns-root.req",
         "filelocation": "/nsconfig/ssl",
         "fileaccesstime": "Tue Jan 14 19:27:01 2014",
         "filemodifiedtime": "Tue Nov  5 17:16:00 2013"
      }
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    ]
}
```

Retrieving Contents of a Specific File

To retrieve the contents of a file, specify the filename and its directory path in the URL.

For example, to retrieve the contents of the ns-root.key file from the /nsconfig/ssl directory.

- **Request:**

  HTTP Method

  GET

  URL

  http://<netscaler-ip-address>/nitro/v1/config/systemfile/ns-root.key?args=filelocation:%2Fnsconfig%2Fssl

  Request Header

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Accept:application/json

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

  Response Header

  Content-Type:application/json

  Response Payload


```
{
   "systemfile":
   [
      {
         "filename": "ns-root.key",
         "filelocation": "/nsconfig/ssl",
         "filecontent": "LS0tLS1CRUdJTiBSU0EgUFJJVkFU0tLQo=",
         "fileencoding": "BASE64",
         "fileaccesstime": "Tue Jan 14 19:27:01 2014",
         "filemodifiedtime": "Tue Nov  5 17:16:00 2013"
      }
   ]
}
```


Deleting a File

To delete a file from the NetScaler appliance, specify the filename and the directory path in the URL.

For example, to delete the ns-root.key file from the /nsconfig/ssl directory.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Request:**

  HTTP Method

  DELETE

  URL

  https://<netscaler-ip-address>/nitro/v1/config/systemfile/ns-root.key?args=filelocation:%2Fnsconfig%2Fssl

  Request Header

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Killing a System Session

Apr 10, 2015

A NetScaler administrator can kill any system session by specifying the action as "kill" in the URL query string and by specifying the required system session ID in the request payload.

For example, to kill a system session that has ID as 311:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/systemsession?action=kill

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

  ```
  {
    "systemsession":
    {
      "sid":"311"
    }
  }
  ```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Disconnecting from the NetScaler Appliance

Apr 10, 2015

Before disconnecting (logging out) from the NetScaler appliance, make sure that you have saved the NetScaler configurations.

To logout of the NetScaler appliance:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/logout

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

  ```
  {
     "logout":{}
  }
  ```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Usage Scenarios

Mar 30, 2015

In this section, we provide NITRO API specific to certain resources and scenarios. We will be adding more scenarios in future updates to this documentation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a NetScaler Cluster

Apr 10, 2015

You can use NITRO to add or create and manage a NetScaler cluster.

Cluster Instance Operations

All operations on a cluster instance must be performed on the clusterinstance object.

For example, to create a cluster instance with ID 1, connect to the NetScaler appliance that you are first adding to the cluster:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/clusterinstance

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
   "clusterinstance":
   {
      "clid":1,
      "preemption":"ENABLED"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  201 Created

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Cluster Node Operations

All operations on a cluster node must be performed on the clusternode object.

For example, to add a NetScaler appliance with NSIP address 10.102.29.60 to the cluster:

- **Request:**

  HTTP Method

  POST

  URL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/clusternode

Request Headers

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

Request Payload

```
{
   "clusternode":
   {
      "nodeid":1,
      "ipaddress":"10.102.29.60",
      "state":"ACTIVE",
      "backplane":"1/1/2"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  201 Created

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Add a Cluster IP Address

To define a cluster IP address, specify the required parameters in the nsip object.

For example, to configure a cluster IP address:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/nsip

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
   "nsip":
   {
      "ipaddress":"10.102.29.61",
      "netmask":"255.255.255.255",
      "type":"CLIP"
   }
}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Response:**

  HTTP Status Code on Success

  201 Created

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Add a Spotted IP Address

To configure an IP address as spotted, specify the required parameters in the nsip object. This configuration must be done on the cluster IP address.

For example, to configure a spotted SNIP address on a node with ID 1:

- **Request:**

  HTTP Method

  POST

  URL

  http://<cluster-ip-address>/nitro/v1/config/nsip

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
  Content-Type:application/json

  Request Payload

```
{
   "nsip":
   {
      "ipaddress":"10.102.29.77",
      "netmask":"255.255.255.0",
      "type":"SNIP",
      "ownernode":1
   }
}
```

- **Response:**

  HTTP Status Code on Success

  201 Created

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Join NetScaler Appliance to Cluster

To join an appliance to a cluster, specify the required parameters in the cluster object.

For example, to join a NetScaler appliance to a cluster:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/cluster

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
   "cluster":
   {
      "clip":"10.102.29.61",
      "password":"verysecret"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Linkset Operations

To configure a linkset, do the following:

1. Create a linkset by specifying the required parameters in the linkset object.
   For example, to add a linkset LS/1:
   - **Request:**

     HTTP Method

     POST

     URL

     http://<cluster-ip-address>/nitro/v1/config/linkset

     Request Headers

     Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

     Content-Type:application/json

     Request Payload

```
{
   "linkset":
   {
      "id":"LS/1"
   }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}
```

- **Response:**

  HTTP Status Code on Success

  201 Created

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

2. Bind the required interfaces to the linkset by specifying the interfaces in the linkset_interface_binding object.
   For example, to bind interfaces 1/1/2 and 2/1/2 to linkset LS/1:

- **Request:**

  HTTP Method

  PUT

  URL

  http://<cluster-ip-address>/nitro/v1/config/linkset_interface_binding/LS%2F1?action=bind

  Note: The linkset name (LS/1), must be URL encoded as LS%2F1.

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
   "linkset_interface_binding":
   {
      "id":"LS/1",
      "ifnum":"1/1/2 2/1/2"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Admin Partitions

Apr 10, 2015

To create an admin partition, you must perform a set of operations on the default partition. To understand this procedure, let us consider a company that has two departments each of which has an application that requires the NetScaler functionality. The NetScaler admin wants to have a different partition for each department so that there is isolation of users and configurations. The NetScaler admin must do the following (the sample shows configurations only for a single admin partition):

Note: For detailed information and best practices, see Admin Partitions.

1. Create a partition and allocate the required resources to that partition.
   - **Request:**

     HTTP Method

     POST

     URL

     http://<netscaler-ip-address>/nitro/v1/config/nspartition

     Request Headers

     Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

     Content-Type:application/json

     Request Payload

     ```
     {
       "nspartition":
       {
         "partitionname":"partition-dept1",
         "maxbandwidth":"10240",
         "minbandwidth":"10240",
         "maxconn":"1024",
         "maxmemlimit":"10"
       }
     }
     ```

   - **Response:**

     HTTP Status Code on Success

     200 Created

     HTTP Status Code on Failure

     4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

2. Associate the appropriate users with the partition.
   - **Request:**

     HTTP Method

     PUT

     URL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/systemuser_nspartition_binding/user1

Request Headers

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Content-Type:application/json

Request Payload

```
{
   "systemuser_nspartition_binding":
   {
      "username":"user1",
      "partitionname":"partition-dept1"
   }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

3. Associate an appropriate command policy to the admin partition user.
   - **Request:**

     HTTP Method

     PUT

     URL

     http://<netscaler-ip-address>/nitro/v1/config/systemuser_systemcmdpolicy_binding/user1

     Request Headers

     Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

     Content-Type:application/json

     Request Payload

```
{
   "systemuser_systemcmdpolicy_binding":
   {
      "username":"user1",
      "policyname":"partition-admin",
      "priority":"1"
   }
}
```

   - **Response:**

     HTTP Status Code on Success

     200 OK

     HTTP Status Code on Failure

     4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides

© 1999-2017 Citrix Systems, Inc. All rights reserved.

details of the error.

4. Specify the VLANs or bridgegroups to be associated with the partition. This step ensures network isolation of the traffic. Traffic received on the interfaces of the VLAN or bridgegroup is isolated from the traffic of other partitions.
   - **Request:**

     HTTP Method

     PUT

     URL

     http://<netscaler-ip-address>/nitro/v1/config/nspartition_vlan_binding/partition-dept1

     Request Headers

     Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

     Content-Type:application/json

     Request Payload

     ```
     {
       "nspartition_vlan_binding":
       {
         "partitionname":"partition-dept1",
         "vlan":"2"
       }
     }
     ```

   - **Response:**

     HTTP Status Code on Success

     200 OK

     HTTP Status Code on Failure

     4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

5. Save the configurations.
   - **Request:**

     HTTP Method

     POST

     URL

     http://<netscaler-ip-address>/nitro/v1/config/nsconfig?action=save

     Request Headers

     Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

     Content-Type:application/json

     Request Payload

     ```
     {
       "nsconfig":
       {
       }
     }
     ```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

The admin partition is created.

6. Now, to configure this admin partition, you must logout of the default partition and logon again. You are automatically taken to the admin partition to which you were first bound and once there you can configure the NetScaler.
   Note: If you want to configure another admin partition, perform the switch operation given in the next step before performing this step.
7. [Optional] If you are associated with multiple admin partitions, you can switch to the required partition.
   - **Request:**

     HTTP Method

     POST

     URL

     http://<netscaler-ip-address>/nitro/v1/config/nspartition?action=Switch

     Request Headers

     Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
     Content-Type:application/json

     Request Payload

     ```
     {
       "nspartition":
       {
         "partitionname":"partition-dept2"
       }
     }
     ```

   - **Response:**

     HTTP Status Code on Success

     200 OK

     HTTP Status Code on Failure

     4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

8. View the partitions that are available on the NetScaler appliance. If a user is associated with more than one partition, the response payload includes the "partitiontype" attribute the value of which indicates the partition to which the user is currently logged on.
   - **Request:**

     HTTP Method

     GET

     URL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nspartition

Request Headers

Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

Accept:application/json

- **Response:**

HTTP Status Code on Success

200 OK

HTTP Status Code on Failure

4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

Response Header

Content-Type:application/json

Response Payload


```
{
  "nspartition":
  {
    "partitionname":"partition-dept1",
    "partitionid": "2",
    "partitiontype": "Current Partition",
    "maxbandwidth":"10240",
    "minbandwidth":"10240",
    "maxconn":"1024",
    "maxmemlimit":"10"
  }
}
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing AppExpert Applications

Apr 10, 2015

To export an AppExpert application, specify the parameters needed for the export operation in the apptemplateinfo object. Optionally, you can specify basic information about the AppExpert application template, such as the author of the configuration, a summary of the template functionality, and the template version number, in the template_info object. This information is stored as part of the template file that is created.

For example, to export an AppExpert application named MyApp1:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/apptemplateinfo?action=export

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
  "apptemplateinfo":
  {
    "appname":"MyApp1",
    "apptemplatefilename":"BizAp.xml",
    "template_info":
    {
      "templateversion_major":"2",
      "templateversion_minor":"1",
      "author":"XYZ",
      "introduction":"Intro",
      "summary":"Summary"
    }
  }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

To import an AppExpert application, specify the parameters needed for the import operation in the apptemplateinfo object.

For example, to import an AppExpert application named MyApp1:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/apptemplateinfo?action=import

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
  "apptemplateinfo":
  {
     "apptemplatefilename":"BizAp.xml",
     "deploymentfilename":"BizAp_deployment.xml",
     "appname":"MyApp1"
  }
}
```

- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

To import an AppExpert application by specifying different deployment settings:

- **Request:**

  HTTP Method

  POST

  URL

  http://<netscaler-ip-address>/nitro/v1/config/apptemplateinfo?action=import

  Request Headers

  Cookie:NITRO_AUTH_TOKEN=<tokenvalue>

  Content-Type:application/json

  Request Payload

```
{
  "apptemplateinfo":
  {
    "apptemplatefilename":"BizAp.xml",
    "appname":"Myapp2",
     "deploymentinfo":
     {
        "appendpoint":
        [
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
            {
                "ipv46":"11.2.3.8",
                "port":80,
                "servicetype":"HTTP"
            }
        ],
        "service":
        [
            {
                "ip":"12.3.3.15",
                "port":80,
                "servicetype":"SSL"
            },
            {
                "ip":"14.5.5.16",
                "port":443,
                "servicetype":"SSL"
            }
        ]
    }
  }
}
```


- **Response:**

  HTTP Status Code on Success

  200 OK

  HTTP Status Code on Failure

  4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Handling

Apr 10, 2015

In case of a failed request, NITRO provides the required information through the HTTP status code and in the response header and response payload.

Error in a Single Resource Operation

The response of a single erroneous operation is as follows:

HTTP Status Code

4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors)

Response Header

Content-Type:application/json

Response Payload

```
{
   errorcode: <Error code>
   message: "<Error message>"
   severity: "ERROR"
}
```

Error in a Bulk Operation

When there is a failure in one of the bulk operations, the response payload gives a combination of success and failure (depends on the value set for X-NITRO-ONERROR in the request header).

HTTP Status Code

207 Multi Status

Response Header

Content-Type:application/json

Response Payload when X-NITRO-ONERROR is set to continue

When the first operation fails, the request is not terminated. The response payload shows the error details of the failed operation and the success status of the other operations.

```
{
   "errorcode": 1243,
   "message": "Bulk operation failed",
   "severity": "ERROR",
   "response":
   [
      {
         "errorcode": 273,
         "message": "Resource already exists",
         "severity": "ERROR"
      },
      {
         "errorcode": 0,
         "message": "Done",
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "severity": "NONE"
      }
    ]
}
```


Response Payload when X-NITRO-ONERROR is set to exit

When the first operation fails, the request is terminated. The response payload only shows the error details of the failed operation.


```
{
   "errorcode": 1243,
   "message": "Bulk operation failed",
   "severity": "ERROR",
   "response":
   [
      {
         "errorcode": 273,
         "message": "Resource already exists",
         "severity": "ERROR"
      }
   ]
}
```


Warnings in NITRO Operations

Warnings can be captured by specifying the "warning" query parameter as "yes" when performing any NITRO operation. For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://<netscaler-ip-address>/nitro/v1/config/login?warning=yes

If there are any warnings, the response is as follows:

HTTP Status Code

209 X-NITRO-WARNING

Response Header

Content-Type:application/json

Response Payload


```
{
   errorcode: <Code>
   message: "<Message>"
   severity: "WARNING"
}
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

API Reference

Jul 05, 2017

This documentation provides details of all operations that can be performed on the NetScaler appliance by using the REST API.

- Configuration
- Statistics
- Error Messages

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuration

Jul 06, 2017

The configuration API is arranged in a logical groups of objects. Choose an appropriate group to find more details.

Name	Description
AAA	Authentication, authorization, and accounting service configuration.
App	Application
Appflow	AppFlow configuration.
Application Firewall	Application Firewall configuration.
Appqoe	Application Level Quality of Experience configuration
Audit	Audit configuration.
Authentication	Authentication configuration.
Authorization	Authorization configuration. Authorization services check which resources users are authorized to access, and grant permissions accordingly.
Autoscale	AutoScale configuration.
Basic	Basic system configuration.
Ca	Content Accelerator Configuration. The Content Accelerator feature is used to cache static and dynamic web application data.
Integrated Caching	Integrated Caching Configuration. The Integrated Caching feature is used to cache static and dynamic web application data.
Cluster	Cluster commands.
Compression	Compression configuration. The system's feature for compressing HTTP responses to compression-aware browsers.
Cache Redirection	Cache redirection configuration. The cache redirection feature can transparently redirect cacheable HTTP requests to a cache and send non-cacheable or dynamic HTTP requests to the origin server. A cache stores or caches frequently requested web content and serves such web content to a client on behalf of the origin servers, alleviating the load on

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
Content Switching	Content Switching configuration. Content Switching feature that enables you to direct traffic to servers on the basis of content.
Db	All the commands associated with database user
Domain Name Service	Domain Name Service(DNS) configuration.
HTTP DoS Protection	HTTP DOS protection configuration.
Front-end-optimization	Front end optimization configuration. The system's feature to optimize Web content for performance.
Filter	Filter configuration. Protection from malicious attacks can be achieved for web sites at the content level with the system's content filtering feature. When this feature is enabled, the system inspects every incoming request with the user-configured rules based on any HTTP URL or headers.
Global Server Load Balancing	Global Server Load Balancing (GSLB) configuration. GSLB feature ensures that client requests are directed to a best performing site available in a global enterprise and distributed Internet environment.
High Availability	High-availability commands.
Ica	ICA configuration.
Ipsec	IPSEC configuration.
Load Balancing	Load Balancing configuration. The load balancing methods manage the selection of an appropriate physical server in a server farm.
LLDP	Link Layer Discovery Protocol.
Lsn	Large Scale NAT commands
Network	Networking commands.
NS	System/Global level configuration.
NTP	NTP configuration.
Policy	Policy configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
Priority Queuing	need. System provides priority-based queuing, which provides different levels of service to customers.
Protocol	Protocol Configuration.
Rdp	RDP configuration.
Reputation	Reputation services configuration.
Responder	Responder configuration.
Rewrite	Rewrite configuration.
Rise	RISE feature related configuration.
Router	Router configuration.
Sure Connect	SureConnect configuration. SureConnect allows the system to deliver alternate content to clients when the system is under heavy load for the original URL. The original content will be served to the client, automatically, after an estimated delay period.
Smpp	All the commands associated with SMPP.
SNMP	SNMP(Simple Network Management Protocol) configuration.
Spillover	Spillover policies and actions.
SSL	SSL Configuration
Stream	Stream configuration.
Subscriber	Subscriber configuration
System	System configuration.
Traffic Management	TM session/policy configuration.
Transform	URL Transformation configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tunnel	
Name	**Description**
Utility	Utilities.
SSL VPN	Virtual Private Network configuration.
Wf	Web Front configuration.
WebInterface	Web Interface configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AAA

May 20, 2016

Authentication, authorization, and accounting service configuration.

Name	Description
aaacertparams	Configuration for certificate parameter
aaaglobal_aaapreauthenticationpolicy_binding	Binding object showing the aaapreauthenticationpolicy that can be bound to aaaglobal
aaaglobal_authenticationnegotiateaction_binding	Binding object showing the authenticationnegotiateaction that can be bound to aaaglobal
aaaglobal_binding	Binding object showing the resources that can be bound to aaaglobal
aaagroup	Configuration for AAA group
aaagroup_aaauser_binding	Binding object showing the aaauser that can be bound to aaagroup
aaagroup_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to aaagroup
aaagroup_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to aaagroup
aaagroup_authorizationpolicy_binding	Binding object showing the authorizationpolicy that can be bound to aaagroup
aaagroup_binding	Binding object showing the resources that can be bound to aaagroup
aaagroup_intranetip6_binding	Binding object showing the intranetip6 that can be bound to aaagroup
aaagroup_intranetip_binding	Binding object showing the intranetip that can be bound to aaagroup
aaagroup_tmsessionpolicy_binding	Binding object showing the tmsessionpolicy that can be bound to aaagroup
aaagroup_vpnintranetapplication_binding	Binding object showing the vpnintranetapplication that can be bound to aaagroup
aaagroup_vpnsessionpolicy_binding	Binding object showing the vpnsessionpolicy that can be bound to aaagroup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_vpntrafficpolicy_binding	
Name	**Description**
aaagroup_vpnurl_binding	Binding object showing the vpnurl that can be bound to aaagroup
aaakcdaccount	Configuration for Kerberos constrained delegation account
aaaldapparams	Configuration for LDAP parameter
aaaparameter	Configuration for AAA parameter
aaapreauthenticationaction	Configuration for pre authentication action
aaapreauthenticationparameter	Configuration for pre authentication parameter
aaapreauthenticationpolicy	Configuration for pre authentication policy
aaapreauthenticationpolicy_aaaglobal_binding	Binding object showing the aaaglobal that can be bound to aaapreauthenticationpolicy
aaapreauthenticationpolicy_binding	Binding object showing the resources that can be bound to aaapreauthenticationpolicy
aaapreauthenticationpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to aaapreauthenticationpolicy
aaaradiusparams	Configuration for RADIUS parameter
aaasession	Configuration for active connection
aaatacacsparams	Configuration for tacacs parameters
aaauser	Configuration for AAA user
aaauser_aaagroup_binding	Binding object showing the aaagroup that can be bound to aaauser
aaauser_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to aaauser
aaauser_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to aaauser
aaauser_authorizationpolicy_binding	Binding object showing the authorizationpolicy that can be bound to aaauser

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_binding Name	Binding object showing the resources that can be bound to aaauser Description
aaauser_intranetip6_binding	Binding object showing the intranetip6 that can be bound to aaauser
aaauser_intranetip_binding	Binding object showing the intranetip that can be bound to aaauser
aaauser_tmsessionpolicy_binding	Binding object showing the tmsessionpolicy that can be bound to aaauser
aaauser_vpnintranetapplication_binding	Binding object showing the vpnintranetapplication that can be bound to aaauser
aaauser_vpnsessionpolicy_binding	Binding object showing the vpnsessionpolicy that can be bound to aaauser
aaauser_vpntrafficpolicy_binding	Binding object showing the vpntrafficpolicy that can be bound to aaauser
aaauser_vpnurl_binding	Binding object showing the vpnurl that can be bound to aaauser

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaacertparams

Jul 07, 2017

Configuration for certificate parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
usernamefield	<String>	Read-write	Client certificate field that contains the username, in the format <field>:<subfield>. .
groupnamefield	<String>	Read-write	Client certificate field that specifies the group, in the format <field>:<subfield>.
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups. Maximum length = 64
twofactor	<String>	Read-only	The state of the two-factor authentication. Possible values = ON, OFF

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/aaacertparams

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaacertparams":{
    "usernamefield":<String_value>,
    "groupnamefield":<String_value>,
    "defaultauthenticationgroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/aaacertparams?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaacertparams":{
    "usernamefield":true,
    "groupnamefield":true,
    "defaultauthenticationgroup":true
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaacertparams

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaacertparams": [ {
      "twofactor":<String_value>,
      "usernamefield":<String_value>,
      "groupnamefield":<String_value>,
      "defaultauthenticationgroup":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaaglobal_aaapreauthenticationpolicy_binding

May 20, 2016

Binding object showing the aaapreauthenticationpolicy that can be bound to aaaglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority of the bound policy.
policy	<String>	Read-write	Name of the policy to be unbound. Minimum length = 1
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
bindpolicytype	<Double>	Read-only	Bound policy type.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaaglobal_aaapreauthenticationpolicy_binding":{
    "policy":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaaglobal_aaapreauthenticationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaaglobal_aaapreauthenticationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaaglobal_aaapreauthenticationpolicy_binding": [ {
    "priority":<Double_value>,
    "policy":<String_value>,
    "builtin":<String[]_value>,
    "bindpolicytype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_aaapreauthenticationpolicy_binding?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "aaaglobal_aaapreauthenticationpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaaglobal_authenticationnegotiateaction_binding

May 20, 2016

Binding object showing the authenticationnegotiateaction that can be bound to aaaglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
windowsprofile	<String>	Read-write	Name of the negotiate profile to be bound. Minimum length = 1 Maximum length = 32
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaaglobal_authenticationnegotiateaction_binding":{
    "windowsprofile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding

Query-parameters:

filter

http://<netscaler-ip-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

address>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaaglobal_authenticationnegotiateaction_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaaglobal_authenticationnegotiateaction_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaaglobal_authenticationnegotiateaction_binding": [ {
    "windowsprofile":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_authenticationnegotiateaction_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "aaaglobal_authenticationnegotiateaction_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaaglobal_binding

May 20, 2016

Binding object which returns the resources bound to aaaglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
aaaglobal_authenticationnegotiateaction_binding	<aaaglobal_authenticationnegotiateaction_binding[]>	Read-only	authenticationnegotiateaction that can be bound to aaaglobal.
aaaglobal_aaapreauthenticationpolicy_binding	<aaaglobal_aaapreauthenticationpolicy_binding[]>	Read-only	aaapreauthenticationpolicy that can be bound to aaaglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaglobal_binding

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "aaaglobal_binding": [ {
    "aaaglobal_authenticationnegotiateaction_binding":<aaaglobal_authenticationnegotiateaction_binding[]_value>,
    "aaaglobal_aaapreauthenticationpolicy_binding":<aaaglobal_aaapreauthenticationpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup

May 20, 2016

Configuration for AAA group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
groupname	<String>	Read-write	Name for the group. Must begin with a letter, number, or the underscore character (_), and must consist only of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the group is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my aaa group? or ?my aaa group). Minimum length = 1
weight	<Double>	Read-write	Weight of this group with respect to other configured aaa groups (lower the number higher the weight). Default value: 0 Minimum value = 0 Maximum value = 65535
loggedin	<Boolean>	Read-write	Display only the group members who are currently logged in.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaagroup":{
    "groupname":<String_value>,
    "weight":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/aaagroup?**args=groupname:<String_value>,loggedin:<Boolean_value>,weight:<Double_value>**

Use this query-parameter to get aaagroup resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/aaagroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaagroup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/aaagroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup": [ {
groupname:<String_value>,loggedin:<Boolean_value>,weight:<Double_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup/groupname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/aaagroup/groupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/aaagroup/groupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup": [ {
groupname:<String_value>,loggedin:<Boolean_value>,weight:<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "aaagroup": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_aaauser_binding

Jul 07, 2017

Binding object showing the aaauser that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
username	<String>	Read-write	The user name.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_aaauser_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_aaauser_binding":{
    "groupname":<String_value>,
    "username":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_aaauser_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_aaauser_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_aaauser_binding": [ {
    "gotopriorityexpression":<String_value>,
    "username":<String_value>,
    "groupname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_aaauser_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaagroup_aaauser_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_auditnslogpolicy_binding

May 20, 2016

Binding object showing the auditnslogpolicy that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority to assign to the policy, as an integer. A lower number indicates a higher priority. Required when binding a group to a policy. Not relevant to any other type of group binding.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy name.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_auditnslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_auditnslogpolicy_binding":{
    "groupname":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_auditnslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_auditnslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_auditnslogpolicy_binding": [ {
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "policy":<String_value>,
    "groupname":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_auditnslogpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaagroup_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_auditsyslogpolicy_binding

May 20, 2016

Binding object showing the auditsyslogpolicy that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority to assign to the policy, as an integer. A lower number indicates a higher priority. Required when binding a group to a policy. Not relevant to any other type of group binding.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy name.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_auditsyslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_auditsyslogpolicy_binding":{
    "groupname":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_auditsyslogpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupname":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_auditsyslogpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaagroup_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_authorizationpolicy_binding

May 20, 2016

Binding object showing the authorizationpolicy that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority to assign to the policy, as an integer. A lower number indicates a higher priority. Required when binding a group to a policy. Not relevant to any other type of group binding.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy name.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_authorizationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_authorizationpolicy_binding":{
    "groupname":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_authorizationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_authorizationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_authorizationpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupname":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_authorizationpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaagroup_authorizationpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_binding

Jul 07, 2017

Binding object which returns the resources bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
groupname	<String>	Read-write	Name of the group. Minimum length = 1
aaagroup_auditnslogpolicy_binding	<aaagroup_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to aaagroup.
aaagroup_intranetip6_binding	<aaagroup_intranetip6_binding[]>	Read-only	intranetip6 that can be bound to aaagroup.
aaagroup_vpnsessionpolicy_binding	<aaagroup_vpnsessionpolicy_binding[]>	Read-only	vpnsessionpolicy that can be bound to aaagroup.
aaagroup_intranetip_binding	<aaagroup_intranetip_binding[]>	Read-only	intranetip that can be bound to aaagroup.
aaagroup_aaauser_binding	<aaagroup_aaauser_binding[]>	Read-only	aaauser that can be bound to aaagroup.
aaagroup_vpntrafficpolicy_binding	<aaagroup_vpntrafficpolicy_binding[]>	Read-only	vpntrafficpolicy that can be bound to aaagroup.
aaagroup_vpnintranetapplication_binding	<aaagroup_vpnintranetapplication_binding[]>	Read-only	vpnintranetapplication that can be bound to aaagroup.
aaagroup_authorizationpolicy_binding	<aaagroup_authorizationpolicy_binding[]>	Read-only	authorizationpolicy that can be bound to aaagroup.
aaagroup_auditsyslogpolicy_binding	<aaagroup_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to aaagroup.
aaagroup_vpnurl_binding	<aaagroup_vpnurl_binding[]>	Read-only	vpnurl that can be bound to aaagroup.
aaagroup_tmsessionpolicy_binding	<aaagroup_tmsessionpolicy_binding[]>	Read-only	tmsessionpolicy that can be bound to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_binding/groupname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "aaagroup_binding": [ {
    "groupname":<String_value>,
    "aaagroup_vpntrafficpolicy_binding":<aaagroup_vpntrafficpolicy_binding[]_value>,
    "aaagroup_authorizationpolicy_binding":<aaagroup_authorizationpolicy_binding[]_value>,
    "aaagroup_intranetip_binding":<aaagroup_intranetip_binding[]_value>,
    "aaagroup_intranetip6_binding":<aaagroup_intranetip6_binding[]_value>,
    "aaagroup_auditnslogpolicy_binding":<aaagroup_auditnslogpolicy_binding[]_value>,
    "aaagroup_vpnurl_binding":<aaagroup_vpnurl_binding[]_value>,
    "aaagroup_auditsyslogpolicy_binding":<aaagroup_auditsyslogpolicy_binding[]_value>,
    "aaagroup_aaauser_binding":<aaagroup_aaauser_binding[]_value>,
    "aaagroup_tmsessionpolicy_binding":<aaagroup_tmsessionpolicy_binding[]_value>,
    "aaagroup_vpnsessionpolicy_binding":<aaagroup_vpnsessionpolicy_binding[]_value>,
    "aaagroup_vpnintranetapplication_binding":<aaagroup_vpnintranetapplication_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_vpnurl_binding

May 20, 2016

Binding object showing the vpnurl that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
urlname	<String>	Read-write	The intranet url.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_vpnurl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_vpnurl_binding":{
    "groupname":<String_value>,
    "urlname":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_vpnurl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_vpnurl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_vpnurl_binding": [ {
    "urlname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "groupname":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnurl_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaagroup_vpnurl_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_intranetip6_binding

May 20, 2016

Binding object showing the intranetip6 that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
intranetip6	<String>	Read-write	The Intranet IP6(s) bound to the group.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
numaddr	<Double>	Read-write	Numbers of ipv6 address bound starting with intranetip6.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_intranetip6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_intranetip6_binding":{
    "groupname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "intranetip6":<String_value>,
    "numaddr":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_intranetip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_intranetip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_intranetip6_binding": [ {
    "intranetip6":<String_value>,
    "gotopriorityexpression":<String_value>,
    "numaddr":<Double_value>,
    "groupname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_intranetip6_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaagroup_intranetip6_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_intranetip_binding

May 20, 2016

Binding object showing the intranetip that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
intranetip	<String>	Read-write	The Intranet IP(s) bound to the group.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
netmask	<String>	Read-write	The netmask for the Intranet IP.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_intranetip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_intranetip_binding":{
    "groupname":<String_value>,
    "intranetip":<String_value>,
    "netmask":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_intranetip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_intranetip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_intranetip_binding": [ {
    "gotopriorityexpression":<String_value>,
    "intranetip":<String_value>,
    "groupname":<String_value>,
    "netmask":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_intranetip_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaagroup_intranetip_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_tmsessionpolicy_binding

May 20, 2016

Binding object showing the tmsessionpolicy that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority to assign to the policy, as an integer. A lower number indicates a higher priority. Required when binding a group to a policy. Not relevant to any other type of group binding.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	The policy name.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_tmsessionpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_tmsessionpolicy_binding":{
    "groupname":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_tmsessionpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_tmsessionpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_tmsessionpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupname":<String_value>,
    "acttype":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_tmsessionpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"aaagroup_tmsessionpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_vpnintranetapplication_binding

May 20, 2016

Binding object showing the vpnintranetapplication that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
intranetapplication	<String>	Read-write	Bind the group to the specified intranet VPN application.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_vpnintranetapplication_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_vpnintranetapplication_binding":{
    "groupname":<String_value>,
    "intranetapplication":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_vpnintranetapplication_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_vpnintranetapplication_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_vpnintranetapplication_binding": [ {
    "gotopriorityexpression":<String_value>,
    "groupname":<String_value>,
    "intranetapplication":<String_value>,
    "acttype":<Double_value>
}]}
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnintranetapplication_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"aaagroup_vpnintranetapplication_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_vpnsessionpolicy_binding

Jul 07, 2017

Binding object showing the vpnsessionpolicy that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority to assign to the policy, as an integer. A lower number indicates a higher priority. Required when binding a group to a policy. Not relevant to any other type of group binding.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	The policy name.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_vpnsessionpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_vpnsessionpolicy_binding":{
    "groupname":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_vpnsessionpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_vpnsessionpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_vpnsessionpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupname":<String_value>,
    "acttype":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpnsessionpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"aaagroup_vpnsessionpolicy_binding": [ { "__count": "#no"} ] }
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

aaagroup_vpntrafficpolicy_binding

May 20, 2016

Binding object showing the vpntrafficpolicy that can be bound to aaagroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority to assign to the policy, as an integer. A lower number indicates a higher priority. Required when binding a group to a policy. Not relevant to any other type of group binding.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy name.
groupname	<String>	Read-write	Name of the group that you are binding. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaagroup_vpntrafficpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaagroup_vpntrafficpolicy_binding":{
    "groupname":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaagroup_vpntrafficpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaagroup_vpntrafficpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaagroup_vpntrafficpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupname":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaagroup_vpntrafficpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaagroup_vpntrafficpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaakcdaccount

Jul 07, 2017

Configuration for Kerberos constrained delegation account resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
kcdaccount	<String>	Read-write	The name of the KCD account. Minimum length = 1
keytab	<String>	Read-write	The path to the keytab file. If specified other parameters in this command need not be given.
realmstr	<String>	Read-write	Kerberos Realm.
delegateduser	<String>	Read-write	Username that can perform kerberos constrained delegation.
kcdpassword	<String>	Read-write	Password for Delegated User.
usercert	<String>	Read-write	SSL Cert (including private key) for Delegated User.
cacert	<String>	Read-write	CA Cert for UserCert or when doing PKINIT backchannel.
userrealm	<String>	Read-write	Realm of the user.
enterpriserealm	<String>	Read-write	Enterprise Realm of the user. This should be given only in certain KDC deployments where KDC expects Enterprise username instead of Principal Name.
servicespn	<String>	Read-write	Service SPN. When specified, this will be used to fetch kerberos tickets. If not specified, Netscaler will construct SPN using service fqdn.
principle	<String>	Read-only	SPN extracted from keytab file.
kcdspn	<String>	Read-only	Host SPN extracted from keytab file.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaakcdaccount":{
    "kcdaccount":<String_value>,
    "keytab":<String_value>,
    "realmstr":<String_value>,
    "delegateduser":<String_value>,
    "kcdpassword":<String_value>,
    "usercert":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cacert":<String_value>,
    "userrealm":<String_value>,
    "enterpriserealm":<String_value>,
    "servicespn":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount/kcdaccount_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaakcdaccount":{
    "kcdaccount":<String_value>,
    "keytab":<String_value>,
    "realmstr":<String_value>,
    "delegateduser":<String_value>,
    "kcdpassword":<String_value>,
    "usercert":<String_value>,
    "cacert":<String_value>,
    "userrealm":<String_value>,
    "enterpriserealm":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "servicespn":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaakcdaccount":{
    "kcdaccount":<String_value>,
    "keytab":true,
    "usercert":true,
    "cacert":true,
    "userrealm":true,
    "enterpriserealm":true,
    "servicespn":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaakcdaccount resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaakcdaccount resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaakcdaccount": [ {
    "kcdaccount":<String_value>,
    "keytab":<String_value>,
    "principle":<String_value>,
    "kcdspn":<String_value>,
    "realmstr":<String_value>,
    "delegateduser":<String_value>,
    "kcdpassword":<String_value>,
    "usercert":<String_value>,
    "cacert":<String_value>,
    "userrealm":<String_value>,
    "enterpriserealm":<String_value>,
    "servicespn":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount/kcdaccount_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount/kcdaccount_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount/kcdaccount_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaakcdaccount": [ {
      "kcdaccount":<String_value>,
      "keytab":<String_value>,
      "principle":<String_value>,
      "kcdspn":<String_value>,
      "realmstr":<String_value>,
      "delegateduser":<String_value>,
      "kcdpassword":<String_value>,
      "usercert":<String_value>,
      "cacert":<String_value>,
      "userrealm":<String_value>,
      "enterpriserealm":<String_value>,
      "servicespn":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaakcdaccount?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaakcdaccount": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaaldapparams

May 20, 2016

Configuration for LDAP parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
serverip	<String>	Read-write	IP address of your LDAP server.
serverport	<Integer>	Read-write	Port number on which the LDAP server listens for connections. Default value: 389 Minimum value = 1
authtimeout	<Double>	Read-write	Maximum number of seconds that the NetScaler appliance waits for a response from the LDAP server. Default value: 3 Minimum value = 1
ldapbase	<String>	Read-write	Base (the server and location) from which LDAP search commands should start. If the LDAP server is running locally, the default value of base is dc=netscaler, dc=com.
ldapbinddn	<String>	Read-write	Complete distinguished name (DN) string used for binding to the LDAP server.
ldapbinddnpassword	<String>	Read-write	Password for binding to the LDAP server. Minimum length = 1
ldaploginname	<String>	Read-write	Name attribute that the NetScaler appliance uses to query the external LDAP server or an Active Directory.
searchfilter	<String>	Read-write	String to be combined with the default LDAP user search string to form the value to use when executing an LDAP search. For example, the following values: vpnallowed=true, ldaploginame=""samaccount"" when combined with the user-supplied username ""bob"", yield the following LDAP search string: ""(&(vpnallowed=true)(samaccount=bob)"". Minimum length = 1
groupattrname	<String>	Read-write	Attribute name used for group extraction from the LDAP server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sectype	<String>	Read-write	Type of security used for communications between the NetScaler appliance and the LDAP server. For the PLAINTEXT setting, no encryption is required. Default value: PLAINTEXT Possible values = PLAINTEXT, TLS, SSL
svrtype	<String>	Read-write	The type of LDAP server. Default value: AAA_LDAP_SERVER_TYPE_DEFAULT Possible values = AD, NDS
ssonameattribute	<String>	Read-write	Attribute used by the NetScaler appliance to query an external LDAP server or Active Directory for an alternative username. This alternative username is then used for single sign-on (SSO).
passwdchange	<String>	Read-write	Accept password change requests. Default value: DISABLED Possible values = ENABLED, DISABLED
nestedgroupextraction	<String>	Read-write	Queries the external LDAP server to determine whether the specified group belongs to another group. Default value: OFF Possible values = ON, OFF
maxnestinglevel	<Double>	Read-write	Number of levels up to which the system can query nested LDAP groups. Default value: 2 Minimum value = 2
groupnameidentifier	<String>	Read-write	LDAP-group attribute that uniquely identifies the group. No two groups on one LDAP server can have the same group name identifier.
groupsearchattribute	<String>	Read-write	LDAP-group attribute that designates the parent group of the specified group. Use this attribute to search for a group?s parent group.
groupsearchsubattribute	<String>	Read-write	LDAP-group subattribute that designates the parent group of the specified group. Use this attribute to search for a group?s parent group.
groupsearchfilter	<String>	Read-write	Search-expression that can be specified for sending group-search requests to the LDAP server.
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups. Maximum length = 64
groupauthname	<String>	Read-only	To associate AAA users with an AAA group, use the command "bind AAA group ... -username ...". You can bind different policies to each AAA group. Use the command "bind AAA group ... -policy ...".
			Indicates that a variable is a built-in (SYSTEM INTERNAL) type.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Permissions)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaldapparams

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaaldapparams":{
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "ldapbase":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ldapbinddn":<String_value>,
    "ldapbinddnpassword":<String_value>,
    "ldaploginname":<String_value>,
    "searchfilter":<String_value>,
    "groupattrname":<String_value>,
    "subattributename":<String_value>,
    "sectype":<String_value>,
    "svrtype":<String_value>,
    "ssonameattribute":<String_value>,
    "passwdchange":<String_value>,
    "nestedgroupextraction":<String_value>,
    "maxnestinglevel":<Double_value>,
    "groupnameidentifier":<String_value>,
    "groupsearchattribute":<String_value>,
    "groupsearchsubattribute":<String_value>,
    "groupsearchfilter":<String_value>,
    "defaultauthenticationgroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaldapparams?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaaldapparams":{
    "serverip":true,
    "serverport":true,
    "authtimeout":true,
    "ldapbase":true,
    "ldapbinddn":true,
    "ldapbinddnpassword":true,
    "ldaploginname":true,
    "searchfilter":true,
    "groupattrname":true,
    "subattributename":true,
    "sectype":true,
    "svrtype":true,
    "ssonameattribute":true,
    "passwdchange":true,
    "nestedgroupextraction":true,
    "maxnestinglevel":true,
    "groupnameidentifier":true,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "groupsearchattribute":true,
    "groupsearchsubattribute":true,
    "groupsearchfilter":true,
    "defaultauthenticationgroup":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaldapparams

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaaldapparams": [ {
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "ldapbinddn":<String_value>,
    "ldaploginname":<String_value>,
    "ldapbase":<String_value>,
    "sectype":<String_value>,
    "svrtype":<String_value>,
    "ssonameattribute":<String_value>,
    "searchfilter":<String_value>,
    "groupattrname":<String_value>,
    "subattributename":<String_value>,
    "groupauthname":<String_value>,
    "passwdchange":<String_value>,
    "nestedgroupextraction":<String_value>,
    "maxnestinglevel":<Double_value>,
    "groupnameidentifier":<String_value>,
    "groupsearchattribute":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "groupsearchsubattribute":<String_value>,
    "groupsearchfilter":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaaparameter

May 20, 2016

Configuration for AAA parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
enablestaticpagecaching	<String>	Read-write	The default state of VPN Static Page caching. If nothing is specified, the default value is set to YES. Default value: YES Possible values = YES, NO
enableenhancedauthfeedback	<String>	Read-write	Enhanced auth feedback provides more information to the end user about the reason for an authentication failure. The default value is set to NO. Default value: NO Possible values = YES, NO
defaultauthtype	<String>	Read-write	The default authentication server type. Default value: LOCAL Possible values = LOCAL, LDAP, RADIUS, TACACS, CERT
maxaaausers	<Double>	Read-write	Maximum number of concurrent users allowed to log on to VPN simultaneously. Minimum value = 1
maxloginattempts	<Double>	Read-write	Maximum Number of login Attempts. Minimum value = 1
failedlogintimeout	<Double>	Read-write	Number of minutes an account will be locked if user exceeds maximum permissible attempts. Minimum value = 1
aaadnatip	<String>	Read-write	Source IP address to use for traffic that is sent to the authentication server.
enablesessionstickiness	<String>	Read-write	Enables/Disables stickiness to authentication servers. Default value: NO Possible values = YES, NO
			Audit log level, which specifies the types of events to log for cli executed commands. Available values function as follows: * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			* WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. Default value: DEFAULT_LOGLEVEL_AAA Possible values = EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"aaaparameter":{
    "enablestaticpagecaching":<String_value>,
    "enableenhancedauthfeedback":<String_value>,
    "defaultauthtype":<String_value>,
    "maxaaausers":<Double_value>,
    "maxloginattempts":<Double_value>,
    "failedlogintimeout":<Double_value>,
    "aaadnatip":<String_value>,
    "enablesessionstickiness":<String_value>,
    "aaasessionloglevel":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaaparameter":{
    "enablestaticpagecaching":true,
    "enableenhancedauthfeedback":true,
    "defaultauthtype":true,
    "maxaaausers":true,
    "aaadnatip":true,
    "maxloginattempts":true,
    "enablesessionstickiness":true,
    "aaasessionloglevel":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaaparameter": [ {
      "enablestaticpagecaching":<String_value>,
      "enableenhancedauthfeedback":<String_value>,
      "defaultauthtype":<String_value>,
      "maxaaausers":<Double_value>,
      "aaadnatip":<String_value>,
      "maxloginattempts":<Double_value>,
      "failedlogintimeout":<Double_value>,
      "enablesessionstickiness":<String_value>,
      "aaasessionloglevel":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaapreauthenticationaction

Jul 07, 2017

Configuration for pre authentication action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the preauthentication action. Must begin with a letter, number, or the underscore character (_), and must consist only of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after preauthentication action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my aaa action? or ?my aaa action). Minimum length = 1
preauthenticationaction	<String>	Read-write	Allow or deny logon after endpoint analysis (EPA) results. Possible values = ALLOW, DENY
killprocess	<String>	Read-write	String specifying the name of a process to be terminated by the endpoint analysis (EPA) tool.
deletefiles	<String>	Read-write	String specifying the path(s) and name(s) of the files to be deleted by the endpoint analysis (EPA) tool.
defaultepagroup	<String>	Read-write	This is the default group that is chosen when the EPA check succeeds. Maximum length = 64
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaapreauthenticationaction":{
    "name":<String_value>,
    "preauthenticationaction":<String_value>,
    "killprocess":<String_value>,
    "deletefiles":<String_value>,
    "defaultepagroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

delete

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaapreauthenticationaction":{
    "name":<String_value>,
    "preauthenticationaction":<String_value>,
    "killprocess":<String_value>,
    "deletefiles":<String_value>,
    "defaultepagroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"aaapreauthenticationaction":{
    "name":<String_value>,
    "killprocess":true,
    "deletefiles":true,
    "defaultepagroup":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaapreauthenticationaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaapreauthenticationaction resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaapreauthenticationaction": [ {
      "name":<String_value>,
      "preauthenticationaction":<String_value>,
      "killprocess":<String_value>,
      "deletefiles":<String_value>,
      "builtin":<String[]_value>,
      "defaultepagroup":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaapreauthenticationaction": [ {
    "name":<String_value>,
    "preauthenticationaction":<String_value>,
    "killprocess":<String_value>,
    "deletefiles":<String_value>,
    "builtin":<String[]_value>,
    "defaultepagroup":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaapreauthenticationaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaapreauthenticationparameter

May 20, 2016

Configuration for pre authentication parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
preauthenticationaction	<String>	Read-write	Deny or allow login on the basis of end point analysis results. Possible values = ALLOW, DENY
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, to be evaluated by the EPA tool.
killprocess	<String>	Read-write	String specifying the name of a process to be terminated by the EPA tool.
deletefiles	<String>	Read-write	String specifying the path(s) to and name(s) of the files to be deleted by the EPA tool, as a string of between 1 and 1023 characters.
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaapreauthenticationparameter":{
    "preauthenticationaction":<String_value>,
    "rule":<String_value>,
    "killprocess":<String_value>,
    "deletefiles":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaapreauthenticationparameter":{
    "rule":true,
    "preauthenticationaction":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "killprocess":true,
    "deletefiles":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaapreauthenticationparameter": [ {
    "preauthenticationaction":<String_value>,
    "rule":<String_value>,
    "killprocess":<String_value>,
    "deletefiles":<String_value>,
    "builtin":<String[]_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaapreauthenticationpolicy

May 20, 2016

Configuration for pre authentication policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the preauthentication policy. Must begin with a letter, number, or the underscore character (_), and must consist only of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the preauthentication policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my policy? or ?my policy?). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, defining connections that match the policy.
reqaction	<String>	Read-write	Name of the action that the policy is to invoke when a connection matches the policy. Minimum length = 1
hits	<Double>	Read-only	No of hits.
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaapreauthenticationpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaapreauthenticationpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy?**filter=property-name1:property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

val1,property-name2:property-val2

Use this query-parameter to get the filtered set of aaapreauthenticationpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaapreauthenticationpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaapreauthenticationpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>,
    "hits":<Double_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy/name_value<String>

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaapreauthenticationpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>,
    "hits":<Double_value>,
    "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "aaapreauthenticationpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaapreauthenticationpolicy_aaaglobal_binding

May 20, 2016

Binding object showing the aaaglobal that can be bound to aaapreauthenticationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the preauthentication policy whose properties you want to view. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy_aaaglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy_aaaglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaapreauthenticationpolicy_aaaglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy_aaaglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaapreauthenticationpolicy_aaaglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaapreauthenticationpolicy_aaaglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy_aaaglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"aaapreauthenticationpolicy_aaaglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaapreauthenticationpolicy_binding

May 20, 2016

Binding object which returns the resources bound to aaapreauthenticationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the preauthentication policy whose properties you want to view. Minimum length = 1
aaapreauthenticationpolicy_vpnvserver_binding	<aaapreauthenticationpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to aaapreauthenticationpolicy.
aaapreauthenticationpolicy_aaaglobal_binding	<aaapreauthenticationpolicy_aaaglobal_binding[]>	Read-only	aaaglobal that can be bound to aaapreauthenticationpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "aaapreauthenticationpolicy_binding": [ {
    "name":<String_value>,
    "aaapreauthenticationpolicy_vpnvserver_binding":<aaapreauthenticationpolicy_vpnvserver_binding[]_value>,
    "aaapreauthenticationpolicy_aaaglobal_binding":<aaapreauthenticationpolicy_aaaglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaapreauthenticationpolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to aaapreauthenticationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the preauthentication policy whose properties you want to view. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaapreauthenticationpolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaapreauthenticationpolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaapreauthenticationpolicy_vpnvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaapreauthenticationpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"aaapreauthenticationpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaaradiusparams

May 20, 2016

Configuration for RADIUS parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
serverip	<String>	Read-write	IP address of your RADIUS server. Minimum length = 1
serverport	<Integer>	Read-write	Port number on which the RADIUS server listens for connections. Default value: 1812 Minimum value = 1
authtimeout	<Double>	Read-write	Maximum number of seconds that the NetScaler appliance waits for a response from the RADIUS server. Default value: 3 Minimum value = 1
radkey	<String>	Read-write	The key shared between the RADIUS server and clients. Required for allowing the NetScaler appliance to communicate with the RADIUS server. Minimum length = 1
radnasip	<String>	Read-write	Send the NetScaler IP (NSIP) address to the RADIUS server as the Network Access Server IP (NASIP) part of the Radius protocol. Possible values = ENABLED, DISABLED
radnasid	<String>	Read-write	Send the Network Access Server ID (NASID) for your NetScaler appliance to the RADIUS server as the nasid part of the Radius protocol.
radvendorid	<Double>	Read-write	Vendor ID for RADIUS group extraction. Minimum value = 1
radattributetype	<Double>	Read-write	Attribute type for RADIUS group extraction. Minimum value = 1
radgroupsprefix	<String>	Read-write	Prefix string that precedes group names within a RADIUS attribute for RADIUS group extraction.
radgroupseparator	<String>	Read-write	Group separator string that delimits group names within a RADIUS attribute for RADIUS group extraction.
passencoding	<String>	Read-write	Enable password encoding in RADIUS packets that the NetScaler appliance sends to the RADIUS server. Default value: pap

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ipvendorid	<Double>	Read-write	Vendor ID attribute in the RADIUS response. If the attribute is not vendor-encoded, it is set to 0.
ipattributetype	<Double>	Read-write	IP attribute type in the RADIUS response. Minimum value = 1
accounting	<String>	Read-write	Configure the RADIUS server state to accept or refuse accounting messages. Possible values = ON, OFF
pwdvendorid	<Double>	Read-write	Vendor ID of the password in the RADIUS response. Used to extract the user password. Minimum value = 1
pwdattributetype	<Double>	Read-write	Attribute type of the Vendor ID in the RADIUS response. Minimum value = 1
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups. Maximum length = 64
callingstationid	<String>	Read-write	Send Calling-Station-ID of the client to the RADIUS server. IP Address of the client is sent as its Calling-Station-ID. Default value: DISABLED Possible values = ENABLED, DISABLED
groupauthname	<String>	Read-only	To associate AAA users with an AAA group, use the command "bind AAA group ... -username ...". You can bind different policies to each AAA group. Use the command "bind AAA group ... -policy ...".
ipaddress	<String>	Read-only	IP Address.
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaradiusparams

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaaradiusparams":{
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "radkey":<String_value>,
    "radnasip":<String_value>,
    "radnasid":<String_value>,
    "radvendorid":<Double_value>,
    "radattributetype":<Double_value>,
    "radgroupsprefix":<String_value>,
    "radgroupseparator":<String_value>,
    "passencoding":<String_value>,
    "ipvendorid":<Double_value>,
    "ipattributetype":<Double_value>,
    "accounting":<String_value>,
    "pwdvendorid":<Double_value>,
    "pwdattributetype":<Double_value>,
    "defaultauthenticationgroup":<String_value>,
    "callingstationid":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaradiusparams?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaaradiusparams":{
    "serverip":true,
    "serverport":true,
    "authtimeout":true,
    "radnasip":true,
    "radnasid":true,
    "radvendorid":true,
    "radattributetype":true,
    "radgroupsprefix":true,
    "radgroupseparator":true,
    "passencoding":true,
    "ipvendorid":true,
    "ipattributetype":true,
    "accounting":true,
    "pwdvendorid":true,
    "pwdattributetype":true,
    "defaultauthenticationgroup":true,
    "callingstationid":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaaradiusparams

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaaradiusparams": [ {
      "serverip":<String_value>,
      "serverport":<Integer_value>,
      "radkey":<String_value>,
      "groupauthname":<String_value>,
      "authtimeout":<Double_value>,
      "radnasip":<String_value>,
      "radnasid":<String_value>,
      "ipaddress":<String_value>,
      "radvendorid":<Double_value>,
      "radattributetype":<Double_value>,
      "radgroupsprefix":<String_value>,
      "radgroupseparator":<String_value>,
      "passencoding":<String_value>,
      "ipvendorid":<Double_value>,
      "ipattributetype":<Double_value>,
      "accounting":<String_value>,
      "pwdvendorid":<Double_value>,
      "pwdattributetype":<Double_value>,
      "defaultauthenticationgroup":<String_value>,
      "callingstationid":<String_value>,
      "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaasession

May 20, 2016

Configuration for active connection resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	Name of the AAA user. Minimum length = 1
groupname	<String>	Read-write	Name of the AAA group. Minimum length = 1
iip	<String>	Read-write	IP address or the first address in the intranet IP range. Minimum length = 1
netmask	<String>	Read-write	Subnet mask for the intranet IP range. Minimum length = 1
all	<Boolean>	Read-write	Terminate all active AAA-TM/VPN sessions.
publicip	<String>	Read-only	Client's public IP address.
publicport	<Integer>	Read-only	Client's public port. Range 1 - 65535
ipaddress	<String>	Read-only	NetScaler's IP address.
port	<Integer>	Read-only	NetScaler's port. Range 1 - 65535
privateip	<String>	Read-only	Client's private/mapped IP address.
privateport	<Integer>	Read-only	Client's private/mapped port. Range 1 - 65535
destip	<String>	Read-only	Destination IP address.
destport	<Integer>	Read-only	Destination port. Range 1 - 65535
intranetip	<String>	Read-only	Specifies the Intranet IP.
intranetip6	<String>	Read-only	Specifies the Intranet IP6.
peid	<Double>	Read-only	Core id of the session owner.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

KILL | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

kill

URL: http://<netscaler-ip-address>/nitro/v1/config/aaasession?**action=kill**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaasession":{
    "username":<String_value>,
    "groupname":<String_value>,
    "iip":<String_value>,
    "netmask":<String_value>,
    "all":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaasession

Query-parameters:

args

http://<netscaler-ip-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

address>/nitro/v1/config/aaasession?**args=username:<String_value>,groupname:<String_value>,iip:<String_value>,netmask:<String_value>**

Use this query-parameter to get aaasession resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/aaasession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/aaasession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaasession resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/aaasession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaasession?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaasession resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaasession": [ {
username:<String_value>,groupname:<String_value>,iip:<String_value>,netmask:<String_value>      "publicip":<String_value>,
      "publicport":<Integer_value>,
      "ipaddress":<String_value>,
      "port":<Integer_value>,
      "privateip":<String_value>,
      "privateport":<Integer_value>,
      "destip":<String_value>,
      "destport":<Integer_value>,
      "intranetip":<String_value>,
      "intranetip6":<String_value>,
      "peid":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaasession?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaasession": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaatacacsparams

May 20, 2016

Configuration for tacacs parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
serverip	<String>	Read-write	IP address of your TACACS+ server. Minimum length = 1
serverport	<Integer>	Read-write	Port number on which the TACACS+ server listens for connections. Default value: 49 Minimum value = 1
authtimeout	<Double>	Read-write	Maximum number of seconds that the NetScaler appliance waits for a response from the TACACS+ server. Default value: 3 Minimum value = 1
tacacssecret	<String>	Read-write	Key shared between the TACACS+ server and clients. Required for allowing the NetScaler appliance to communicate with the TACACS+ server. Minimum length = 1
authorization	<String>	Read-write	Use streaming authorization on the TACACS+ server. Possible values = ON, OFF
accounting	<String>	Read-write	Send accounting messages to the TACACS+ server. Possible values = ON, OFF
auditfailedcmds	<String>	Read-write	The option for sending accounting messages to the TACACS+ server. Possible values = ON, OFF
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups. Maximum length = 64
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/aaatacacsparams

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaatacacsparams":{
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "tacacssecret":<String_value>,
    "authorization":<String_value>,
    "accounting":<String_value>,
    "auditfailedcmds":<String_value>,
    "defaultauthenticationgroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/aaatacacsparams?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaatacacsparams":{
    "serverip":true,
    "serverport":true,
    "authtimeout":true,
    "tacacssecret":true,
    "authorization":true,
    "accounting":true,
    "auditfailedcmds":true,
    "defaultauthenticationgroup":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaatacacsparams

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "aaatacacsparams": [ {
      "serverip":<String_value>,
      "serverport":<Integer_value>,
      "authtimeout":<Double_value>,
      "tacacssecret":<String_value>,
      "authorization":<String_value>,
      "accounting":<String_value>,
      "auditfailedcmds":<String_value>,
      "defaultauthenticationgroup":<String_value>,
      "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser

Jul 07, 2017

Configuration for AAA user resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	Name for the user. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the user is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my aaa user" or "my aaa user"). Minimum length = 1
password	<String>	Read-write	Password with which the user logs on. Required for any user account that does not exist on an external authentication server. If you are not using an external authentication server, all user accounts must have a password. If you are using an external authentication server, you must provide a password for local user accounts that do not exist on the authentication server. Minimum length = 1
loggedin	<Boolean>	Read-write	Show whether the user is logged in or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNLOCK | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaauser":{
    "username":<String_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaauser":{
    "username":<String_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unlock

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser?**action=unlock**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"aaauser":{
    "username":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/aaauser?**args=username:<String_value>,loggedin:<Boolean_value>**

Use this query-parameter to get aaauser resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/aaauser?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of aaauser resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/aaauser?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser": [ {
username:<String_value>,loggedin:<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser/username_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/aaauser/username_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/aaauser/username_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "aaauser": [ {
username:<String_value>,loggedin:<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_aaagroup_binding

May 20, 2016

Binding object showing the aaagroup that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
groupname	<String>	Read-write	The group name.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_aaagroup_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_aaagroup_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_aaagroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_aaagroup_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_aaagroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser_aaagroup_binding": [ {
      "gotopriorityexpression":<String_value>,
      "username":<String_value>,
      "groupname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_aaagroup_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_aaagroup_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_auditnslogpolicy_binding

May 20, 2016

Binding object showing the auditnslogpolicy that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy Name.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaauser_auditnslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaauser_auditnslogpolicy_binding":{
    "username":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_auditnslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_auditnslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser_auditnslogpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "username":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_auditnslogpolicy_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_auditsyslogpolicy_binding

May 20, 2016

Binding object showing the auditsyslogpolicy that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy Name.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaauser_auditsyslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaauser_auditsyslogpolicy_binding":{
    "username":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser_auditsyslogpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "username":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_auditsyslogpolicy_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_authorizationpolicy_binding

May 20, 2016

Binding object showing the authorizationpolicy that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy Name.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaauser_authorizationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaauser_authorizationpolicy_binding":{
    "username":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_authorizationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_authorizationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser_authorizationpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "username":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_authorizationpolicy_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_authorizationpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_binding

Jul 07, 2017

Binding object which returns the resources bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	Name of the user who has the account. Minimum length = 1
aaauser_auditnslogpolicy_binding	<aaauser_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to aaauser.
aaauser_intranetip6_binding	<aaauser_intranetip6_binding[]>	Read-only	intranetip6 that can be bound to aaauser.
aaauser_vpnsessionpolicy_binding	<aaauser_vpnsessionpolicy_binding[]>	Read-only	vpnsessionpolicy that can be bound to aaauser.
aaauser_intranetip_binding	<aaauser_intranetip_binding[]>	Read-only	intranetip that can be bound to aaauser.
aaauser_vpntrafficpolicy_binding	<aaauser_vpntrafficpolicy_binding[]>	Read-only	vpntrafficpolicy that can be bound to aaauser.
aaauser_vpnintranetapplication_binding	<aaauser_vpnintranetapplication_binding[]>	Read-only	vpnintranetapplication that can be bound to aaauser.
aaauser_authorizationpolicy_binding	<aaauser_authorizationpolicy_binding[]>	Read-only	authorizationpolicy that can be bound to aaauser.
aaauser_auditsyslogpolicy_binding	<aaauser_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to aaauser.
aaauser_aaagroup_binding	<aaauser_aaagroup_binding[]>	Read-only	aaagroup that can be bound to aaauser.
aaauser_vpnurl_binding	<aaauser_vpnurl_binding[]>	Read-only	vpnurl that can be bound to aaauser.
aaauser_tmsessionpolicy_binding	<aaauser_tmsessionpolicy_binding[]>	Read-only	tmsessionpolicy that can be bound to aaauser.

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_binding/username_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "aaauser_binding": [ {
    "username":<String_value>,
    "aaauser_intranetip_binding":<aaauser_intranetip_binding[]_value>,
    "aaauser_auditsyslogpolicy_binding":<aaauser_auditsyslogpolicy_binding[]_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "aaauser_vpnintranetapplication_binding":<aaauser_vpnintranetapplication_binding[]_value>,
    "aaauser_authorizationpolicy_binding":<aaauser_authorizationpolicy_binding[]_value>,
    "aaauser_vpnurl_binding":<aaauser_vpnurl_binding[]_value>,
    "aaauser_vpntrafficpolicy_binding":<aaauser_vpntrafficpolicy_binding[]_value>,
    "aaauser_aaagroup_binding":<aaauser_aaagroup_binding[]_value>,
    "aaauser_vpnsessionpolicy_binding":<aaauser_vpnsessionpolicy_binding[]_value>,
    "aaauser_intranetip6_binding":<aaauser_intranetip6_binding[]_value>,
    "aaauser_tmsessionpolicy_binding":<aaauser_tmsessionpolicy_binding[]_value>,
    "aaauser_auditnslogpolicy_binding":<aaauser_auditnslogpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_intranetip6_binding

May 20, 2016

Binding object showing the intranetip6 that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
intranetip6	<String>	Read-write	The Intranet IP6 bound to the user.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
numaddr	<Double>	Read-write	Numbers of ipv6 address bound starting with intranetip6.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaauser_intranetip6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaauser_intranetip6_binding":{
    "username":<String_value>,
    "gotopriorityexpression":<String_value>,
    "intranetip6":<String_value>,
    "numaddr":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_intranetip6_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_intranetip6_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_intranetip6_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_intranetip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_intranetip6_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_intranetip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser_intranetip6_binding": [ {
    "intranetip6":<String_value>,
    "gotopriorityexpression":<String_value>,
    "numaddr":<Double_value>,
    "username":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_intranetip6_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_intranetip6_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_intranetip_binding

May 20, 2016

Binding object showing the intranetip that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
intranetip	<String>	Read-write	The Intranet IP bound to the user.
netmask	<String>	Read-write	The netmask for the Intranet IP.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaauser_intranetip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaauser_intranetip_binding":{
    "username":<String_value>,
    "intranetip":<String_value>,
    "netmask":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_intranetip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_intranetip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser_intranetip_binding": [ {
    "gotopriorityexpression":<String_value>,
    "username":<String_value>,
    "intranetip":<String_value>,
    "netmask":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_intranetip_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_intranetip_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_tmsessionpolicy_binding

May 20, 2016

Binding object showing the tmsessionpolicy that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	The policy Name.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaauser_tmsessionpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaauser_tmsessionpolicy_binding":{
    "username":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_tmsessionpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_tmsessionpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser_tmsessionpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "username":<String_value>,
    "acttype":<Double_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_tmsessionpolicy_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_tmsessionpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_vpnintranetapplication_binding

May 20, 2016

Binding object showing the vpnintranetapplication that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
intranetapplication	<String>	Read-write	Name of the intranet VPN application to which the policy applies.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaauser_vpnintranetapplication_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaauser_vpnintranetapplication_binding":{
    "username":<String_value>,
    "intranetapplication":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_vpnintranetapplication_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_vpnintranetapplication_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "aaauser_vpnintranetapplication_binding": [ {
    "gotopriorityexpression":<String_value>,
    "username":<String_value>,
    "intranetapplication":<String_value>,
    "acttype":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnintranetapplication_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_vpnintranetapplication_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_vpnsessionpolicy_binding

May 20, 2016

Binding object showing the vpnsessionpolicy that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	The policy Name.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaauser_vpnsessionpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaauser_vpnsessionpolicy_binding":{
    "username":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_vpnsessionpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_vpnsessionpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser_vpnsessionpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "username":<String_value>,
    "acttype":<Double_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnsessionpolicy_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_vpnsessionpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_vpntrafficpolicy_binding

May 20, 2016

Binding object showing the vpntrafficpolicy that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The policy Name.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaauser_vpntrafficpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaauser_vpntrafficpolicy_binding":{
    "username":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_vpntrafficpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_vpntrafficpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser_vpntrafficpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "username":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpntrafficpolicy_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_vpntrafficpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaauser_vpnurl_binding

May 20, 2016

Binding object showing the vpnurl that can be bound to aaauser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
urlname	<String>	Read-write	The intranet url.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
username	<String>	Read-write	User account to which to bind the policy. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/aaauser_vpnurl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"aaauser_vpnurl_binding":{
    "username":<String_value>,
    "urlname":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of aaauser_vpnurl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the aaauser_vpnurl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaauser_vpnurl_binding": [ {
    "urlname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "username":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/aaauser_vpnurl_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"aaauser_vpnurl_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

App

May 20, 2016

Application

Name	Description
application	Configuration for application

© 1999-2017 Citrix Systems, Inc. All rights reserved.

application

May 20, 2016

Configuration for application resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
apptemplatefilename	<String>	Read-write	Name of the AppExpert application template file.
appname	<String>	Read-write	Name to assign to the application on the NetScaler appliance. If you do not provide a name, the appliance assigns the application the name of the template file.
deploymentfilename	<String>	Read-write	Name of the deployment file.

Operations (click to see Properties)

IMPORT | EXPORT | DELETE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/application?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"application":{
    "apptemplatefilename":<String_value>,
    "appname":<String_value>,
    "deploymentfilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

export

URL: http://<netscaler-ip-address>/nitro/v1/config/application?**action=export**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"application":{
    "appname":<String_value>,
    "apptemplatefilename":<String_value>,
    "deploymentfilename":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/application

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Appflow

May 20, 2016

AppFlow configuration.

Name	Description
appflowaction	Configuration for AppFlow action
appflowcollector	Configuration for AppFlow collector
appflowglobal_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to appflowglobal
appflowglobal_binding	Binding object showing the resources that can be bound to appflowglobal
appflowparam	Configuration for AppFlow parameter
appflowpolicy	Configuration for AppFlow policy
appflowpolicy_appflowglobal_binding	Binding object showing the appflowglobal that can be bound to appflowpolicy
appflowpolicy_appflowpolicylabel_binding	Binding object showing the appflowpolicylabel that can be bound to appflowpolicy
appflowpolicy_binding	Binding object showing the resources that can be bound to appflowpolicy
appflowpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to appflowpolicy
appflowpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to appflowpolicy
appflowpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to appflowpolicy
appflowpolicylabel	Configuration for AppFlow policy label
appflowpolicylabel_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to appflowpolicylabel
appflowpolicylabel_binding	Binding object showing the resources that can be bound to appflowpolicylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowaction

May 20, 2016

Configuration for AppFlow action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow action" or 'my appflow action').
collectors	<String[]>	Read-write	Name(s) of collector(s) to be associated with the AppFlow action. Minimum length = 1
clientsidemeasurements	<String>	Read-write	On enabling this option, the NetScaler will collect the time required to load and render the mainpage on the client. Default value: DISABLED Possible values = ENABLED, DISABLED
pagetracking	<String>	Read-write	On enabling this option, the NetScaler will start tracking the page for waterfall chart by inserting a NS_ESNS cookie in the response. Default value: DISABLED Possible values = ENABLED, DISABLED
comment	<String>	Read-write	Any comments about this action. In the CLI, if including spaces between words, enclose the comment in quotation marks. (The quotation marks are not required in the configuration utility.). Maximum length = 256
newname	<String>	Read-write	New name for the AppFlow action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow action" or 'my appflow action'). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
referencecount	<Double>	Read-only	The number of references to the action.
description	<String>	Read-only	Description of the action.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"appflowaction":{
    "name":<String_value>,
    "collectors":<String[]_value>,
    "clientsidemeasurements":<String_value>,
    "pagetracking":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowaction":{
    "name":<String_value>,
    "collectors":<String[]_value>,
    "clientsidemeasurements":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "pagetracking":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowaction":{
    "name":<String_value>,
    "clientsidemeasurements":true,
    "pagetracking":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowaction?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowaction":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appflowaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appflowaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appflowaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowaction": [ {
    "name":<String_value>,
    "hits":<Double_value>,
    "collectors":<String[]_value>,
    "clientsidemeasurements":<String_value>,
    "pagetracking":<String_value>,
    "referencecount":<Double_value>,
    "description":<String_value>,
    "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appflowaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appflowaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowaction": [ {
    "name":<String_value>,
    "hits":<Double_value>,
    "collectors":<String[]_value>,
    "clientsidemeasurements":<String_value>,
    "pagetracking":<String_value>,
    "referencecount":<Double_value>,
    "description":<String_value>,
    "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowcollector

May 20, 2016

Configuration for AppFlow collector resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the collector. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Only four collectors can be configured. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow collector" or 'my appflow collector'). Minimum length = 1 Maximum length = 127
ipaddress	<String>	Read-write	IPv4 address of the collector.
port	<Integer>	Read-write	UDP port on which the collector listens. Default value: 4739
netprofile	<String>	Read-write	Netprofile to associate with the collector. The IP address defined in the profile is used as the source IP address for AppFlow traffic for this collector. If you do not set this parameter, the NetScaler IP (NSIP) address is used as the source IP address. Maximum length = 128
newname	<String>	Read-write	New name for the collector. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at(@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow coll" or 'my appflow coll'). Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowcollector

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowcollector":{
    "name":<String_value>,
    "ipaddress":<String_value>,
    "port":<Integer_value>,
    "netprofile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowcollector/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowcollector?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowcollector":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowcollector

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/appflowcollector?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowcollector?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appflowcollector resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appflowcollector?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowcollector?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowcollector resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowcollector": [ {
      "name":<String_value>,
      "ipaddress":<String_value>,
      "port":<Integer_value>,
      "netprofile":<String_value>
}]}
```

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowcollector/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appflowcollector/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appflowcollector/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowcollector": [ {
      "name":<String_value>,
      "ipaddress":<String_value>,
      "port":<Integer_value>,
      "netprofile":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowcollector?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "appflowcollector": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowglobal_appflowpolicy_binding

Jul 07, 2017

Binding object showing the appflowpolicy that can be bound to appflowglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the AppFlow policy.
labelname	<String>	Read-write	Name of the label to invoke if the current policy evaluates to TRUE.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority.
type	<String>	Read-write	Global bind point for which to show detailed information about the policies bound to the bind point. Possible values = REQ_OVERRIDE, REQ_DEFAULT, OVERRIDE, DEFAULT, OTHERTCP_REQ_OVERRIDE, OTHERTCP_REQ_DEFAULT, MSSQL_REQ_OVERRIDE, MSSQL_REQ_DEFAULT, MYSQL_REQ_OVERRIDE, MYSQL_REQ_DEFAULT, ICA_REQ_OVERRIDE, ICA_REQ_DEFAULT, ORACLE_REQ_OVERRIDE, ORACLE_REQ_DEFAULT
labeltype	<String>	Read-write	Type of policy label to invoke. Specify vserver for a policy label associated with a virtual server, or policylabel for a user-defined policy label. Possible values = vserver, policylabel
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
flowtype	<Double>	Read-only	Flow type of the bound AppFlow policy.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

Name	Data Type	Permissions	Description

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appflowglobal_appflowpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appflowglobal_appflowpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowglobal_appflowpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowglobal_appflowpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowglobal_appflowpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowglobal_appflowpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowglobal_appflowpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowglobal_appflowpolicy_binding resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowglobal_appflowpolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "type":<String_value>,
    "labeltype":<String_value>,
    "numpol":<Double_value>,
    "flowtype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowglobal_appflowpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "appflowglobal_appflowpolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowglobal_binding

May 20, 2016

Binding object which returns the resources bound to appflowglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
appflowglobal_appflowpolicy_binding	<appflowglobal_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to appflowglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "appflowglobal_binding": [ {
      "appflowglobal_appflowpolicy_binding":<appflowglobal_appflowpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowparam

May 20, 2016

Configuration for AppFlow parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
templaterefresh	<Double>	Read-write	Refresh interval, in seconds, at which to export the template data. Because data transmission is in UDP, the templates must be resent at regular intervals. Default value: 600 Minimum value = 60 Maximum value = 3600
appnamerefresh	<Double>	Read-write	Interval, in seconds, at which to send Appnames to the configured collectors. Appname refers to the name of an entity (virtual server, service, or service group) in the NetScaler appliance. Default value: 600 Minimum value = 60 Maximum value = 3600
flowrecordinterval	<Double>	Read-write	Interval, in seconds, at which to send flow records to the configured collectors. Default value: 60 Minimum value = 60 Maximum value = 3600
securityinsightrecordinterval	<Double>	Read-write	Interval, in seconds, at which to send security insight flow records to the configured collectors. Default value: 600 Minimum value = 60 Maximum value = 3600
udppmtu	<Double>	Read-write	MTU, in bytes, for IPFIX UDP packets. Default value: 1472 Minimum value = 128 Maximum value = 1472
httpurl	<String>	Read-write	Include the http URL that the NetScaler appliance received from the client. Default value: DISABLED Possible values = ENABLED, DISABLED
aaausername	<String>	Read-write	Enable AppFlow AAA Username logging. Default value: DISABLED Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
httpcookie	<String>	Read-write	received from the client. Default value: DISABLED Possible values = ENABLED, DISABLED
httpreferer	<String>	Read-write	Include the web page that was last visited by the client. Default value: DISABLED Possible values = ENABLED, DISABLED
httpmethod	<String>	Read-write	Include the method that was specified in the HTTP request that the appliance received from the client. Default value: DISABLED Possible values = ENABLED, DISABLED
httphost	<String>	Read-write	Include the host identified in the HTTP request that the appliance received from the client. Default value: DISABLED Possible values = ENABLED, DISABLED
httpuseragent	<String>	Read-write	Include the client application through which the HTTP request was received by the NetScaler appliance. Default value: DISABLED Possible values = ENABLED, DISABLED
clienttrafficonly	<String>	Read-write	Generate AppFlow records for only the traffic from the client. Default value: NO Possible values = YES, NO
httpcontenttype	<String>	Read-write	Include the HTTP Content-Type header sent from the server to the client to determine the type of the content sent. Default value: DISABLED Possible values = ENABLED, DISABLED
httpauthorization	<String>	Read-write	Include the HTTP Authorization header information. Default value: DISABLED Possible values = ENABLED, DISABLED
httpvia	<String>	Read-write	Include the httpVia header which contains the IP address of proxy server through which the client accessed the server. Default value: DISABLED Possible values = ENABLED, DISABLED
httpxforwardedfor	<String>	Read-write	Include the httpXForwardedFor header, which contains the original IP Address of the client using a proxy server to access the server. Default value: DISABLED Possible values = ENABLED, DISABLED
httplocation	<String>	Read-write	Include the HTTP location headers returned from the HTTP responses. Default value: DISABLED Possible values = ENABLED, DISABLED
			Include the Set-cookie header sent from the server to the client in response to a HTTP request.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = ENABLED, DISABLED
httpsetcookie2	<String>	Read-write	Include the Set-cookie header sent from the server to the client in response to a HTTP request. Default value: DISABLED Possible values = ENABLED, DISABLED
connectionchaining	<String>	Read-write	Enable connection chaining so that the client server flows of a connection are linked. Also the connection chain ID is propagated across NetScalers, so that in a multi-hop environment the flows belonging to the same logical connection are linked. This id is also logged as part of appflow record. Default value: DISABLED Possible values = ENABLED, DISABLED
httpdomain	<String>	Read-write	Include the http domain request to be exported. Default value: DISABLED Possible values = ENABLED, DISABLED
skipcacheredirectionhttptransaction	<String>	Read-write	Skip Cache http transaction. This HTTP transaction is specific to Cache Redirection module. In Case of Cache Miss there will be another HTTP transaction initiated by the cache server. Default value: DISABLED Possible values = ENABLED, DISABLED
identifiername	<String>	Read-write	Include the stream identifier name to be exported. Default value: DISABLED Possible values = ENABLED, DISABLED
identifiersessionname	<String>	Read-write	Include the stream identifier session name to be exported. Default value: DISABLED Possible values = ENABLED, DISABLED
observationdomainid	<Double>	Read-write	An observation domain groups a set of NetScalers based on deployment: cluster, HA etc. A unique Observation Domain ID is required to be assigned to each such group. Default value: 0 Minimum value = 1000
observationdomainname	<String>	Read-write	Name of the Observation Domain defined by the observation domain ID. Maximum length = 127
securityinsighttraffic	<String>	Read-write	Flag to determine whether security insight traffic needs to be exported or not. Default value: DISABLED Possible values = ENABLED, DISABLED
builtin	<String[]>	Read-only	Flag to determine if the appflow param is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowparam":{
    "templaterefresh":<Double_value>,
    "appnamerefresh":<Double_value>,
    "flowrecordinterval":<Double_value>,
    "securityinsightrecordinterval":<Double_value>,
    "udppmtu":<Double_value>,
    "httpurl":<String_value>,
    "aaausername":<String_value>,
    "httpcookie":<String_value>,
    "httpreferer":<String_value>,
    "httpmethod":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "httphost":<String_value>,
    "httpuseragent":<String_value>,
    "clienttrafficonly":<String_value>,
    "httpcontenttype":<String_value>,
    "httpauthorization":<String_value>,
    "httpvia":<String_value>,
    "httpxforwardedfor":<String_value>,
    "httplocation":<String_value>,
    "httpsetcookie":<String_value>,
    "httpsetcookie2":<String_value>,
    "connectionchaining":<String_value>,
    "httpdomain":<String_value>,
    "skipcacheredirectionhttptransaction":<String_value>,
    "identifiername":<String_value>,
    "identifiersessionname":<String_value>,
    "observationdomainid":<Double_value>,
    "observationdomainname":<String_value>,
    "securityinsighttraffic":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowparam":{
    "templaterefresh":true,
    "appnamerefresh":true,
    "flowrecordinterval":true,
    "securityinsightrecordinterval":true,
    "udppmtu":true,
    "httpurl":true,
    "aaausername":true,
    "httpcookie":true,
    "httpreferer":true,
    "httpmethod":true,
    "httphost":true,
    "httpuseragent":true,
    "clienttrafficonly":true,
    "httpcontenttype":true,
    "httpauthorization":true,
    "httpvia":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "httpxforwardedfor":true,
    "httplocation":true,
    "httpsetcookie":true,
    "httpsetcookie2":true,
    "connectionchaining":true,
    "httpdomain":true,
    "skipcacheredirectionhttptransaction":true,
    "identifiername":true,
    "identifiersessionname":true,
    "observationdomainid":true,
    "observationdomainname":true,
    "securityinsighttraffic":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowparam": [ {
    "templaterefresh":<Double_value>,
    "appnamerefresh":<Double_value>,
    "flowrecordinterval":<Double_value>,
    "securityinsightrecordinterval":<Double_value>,
    "udppmtu":<Double_value>,
    "httpurl":<String_value>,
    "aaausername":<String_value>,
    "httpcookie":<String_value>,
    "httpreferer":<String_value>,
    "httpmethod":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "httphost":<String_value>,
    "httpuseragent":<String_value>,
    "clienttrafficonly":<String_value>,
    "httpcontenttype":<String_value>,
    "httpauthorization":<String_value>,
    "httpvia":<String_value>,
    "httpxforwardedfor":<String_value>,
    "httplocation":<String_value>,
    "httpsetcookie":<String_value>,
    "httpsetcookie2":<String_value>,
    "connectionchaining":<String_value>,
    "httpdomain":<String_value>,
    "skipcacheredirectionhttptransaction":<String_value>,
    "identifiername":<String_value>,
    "identifiersessionname":<String_value>,
    "observationdomainid":<Double_value>,
    "observationdomainname":<String_value>,
    "builtin":<String[]_value>,
    "securityinsighttraffic":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowpolicy

May 20, 2016

Configuration for AppFlow policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow policy" or 'my appflow policy').
rule	<String>	Read-write	Expression or other value against which the traffic is evaluated. Must be a Boolean, default syntax expression. Note: Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the action to be associated with this policy.
comment	<String>	Read-write	Any comments about this policy.
newname	<String>	Read-write	New name for the policy. Must begin with an ASCII alphabetic or underscore (_)character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow policy" or 'my appflow policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
description	<String>	Read-only	Description of the policy.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"appflowpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowpolicy":{
    "name":<String_value>,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowpolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appflowpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "description":<String_value>,
      "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "appflowpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "description":<String_value>,
    "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicy": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowpolicylabel

May 20, 2016

Configuration for AppFlow policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the AppFlow policy label. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow policylabel" or 'my appflow policylabel'). Minimum length = 1
policylabeltype	<String>	Read-write	Type of traffic evaluated by the policies bound to the policy label. Default value: HTTP Possible values = HTTP, OTHERTCP
newname	<String>	Read-write	New name for the policy label. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my appflow policylabel" or 'my appflow policylabel'). Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to the policy label.
hits	<Double>	Read-only	Number of times the policy label was invoked.
policyname	<String>	Read-only	Name of the AppFlow policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
labeltype	<String>	Read-only	Type of policy label to be invoked. Possible values = vserver, policylabel
invoke_labelname	<String>	Read-only	Name of the label to invoke if the current policy evaluates to TRUE.
flowtype	<Double>	Read-only	Flowtype of the bound AppFlow policy.
description	<String>	Read-only	Description of the policylabel.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appflowpolicylabel":{
    "labelname":<String_value>,
    "policylabeltype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"appflowpolicylabel":{
    "labelname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appflowpolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowpolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicylabel": [ {
    "labelname":<String_value>,
    "policylabeltype":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "flowtype":<Double_value>,
    "description":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicylabel": [ {
    "labelname":<String_value>,
    "policylabeltype":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "flowtype":<Double_value>,
    "description":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicylabel": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowpolicylabel_appflowpolicy_binding

Jul 07, 2017

Binding object showing the appflowpolicy that can be bound to appflowpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotoprioritiyexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the AppFlow policy.
labeltype	<String>	Read-write	Type of policy label to be invoked. Possible values = vserver, policylabel
labelname	<String>	Read-write	Name of the policy label to which to bind the policy. Minimum length = 1
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appflowpolicylabel_appflowpolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicylabel_appflowpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowpolicylabel_appflowpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicylabel_appflowpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "invoke":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel_appflowpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appflowpolicylabel_appflowpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowpolicylabel_binding

Jul 07, 2017

Binding object which returns the resources bound to appflowpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the policy label about which to display detailed information. Minimum length = 1
appflowpolicylabel_appflowpolicy_binding	<appflowpolicylabel_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to appflowpolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "appflowpolicylabel_binding": [ {
    "labelname":<String_value>,
    "appflowpolicylabel_appflowpolicy_binding":<appflowpolicylabel_appflowpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowpolicy_appflowglobal_binding

May 20, 2016

Binding object showing the appflowglobal that can be bound to appflowpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy about which to display detailed information.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_appflowglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_appflowglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicy_appflowglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_appflowglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowpolicy_appflowglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicy_appflowglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_appflowglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appflowpolicy_appflowglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowpolicy_appflowpolicylabel_binding

May 20, 2016

Binding object showing the appflowpolicylabel that can be bound to appflowpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy about which to display detailed information.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then

© 1999-2017 Citrix Systems, Inc. All rights reserved.

using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_appflowpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_appflowpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicy_appflowpolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_appflowpolicylabel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowpolicy_appflowpolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicy_appflowpolicylabel_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_appflowpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appflowpolicy_appflowpolicylabel_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowpolicy_binding

May 20, 2016

Binding object which returns the resources bound to appflowpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy about which to display detailed information.
appflowpolicy_csvserver_binding	<appflowpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to appflowpolicy.
appflowpolicy_lbvserver_binding	<appflowpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to appflowpolicy.
appflowpolicy_appflowglobal_binding	<appflowpolicy_appflowglobal_binding[]>	Read-only	appflowglobal that can be bound to appflowpolicy.
appflowpolicy_vpnvserver_binding	<appflowpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to appflowpolicy.
appflowpolicy_appflowpolicylabel_binding	<appflowpolicy_appflowpolicylabel_binding[]>	Read-only	appflowpolicylabel that can be bound to appflowpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "appflowpolicy_binding": [ {
    "name":<String_value>,
    "appflowpolicy_appflowpolicylabel_binding":<appflowpolicy_appflowpolicylabel_binding[]_value>,
    "appflowpolicy_csvserver_binding":<appflowpolicy_csvserver_binding[]_value>,
    "appflowpolicy_vpnvserver_binding":<appflowpolicy_vpnvserver_binding[]_value>,
    "appflowpolicy_lbvserver_binding":<appflowpolicy_lbvserver_binding[]_value>,
    "appflowpolicy_appflowglobal_binding":<appflowpolicy_appflowglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowpolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to appflowpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy about which to display detailed information.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicy_csvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appflowpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowpolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to appflowpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy about which to display detailed information.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicy_lbvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appflowpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflowpolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to appflowpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy about which to display detailed information.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appflowpolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appflowpolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflowpolicy_vpnvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appflowpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appflowpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Firewall

Jul 07, 2017

Application Firewall configuration.

Name	Description
appfwarchive	Configuration for archive
appfwconfidfield	Configuration for configured confidential form fields
appfwcustomsettings	Configuration for application firewall custom settings XML configuration
appfwfieldtype	Configuration for application firewall form field type
appfwglobal_appfwpolicy_binding	Binding object showing the appfwpolicy that can be bound to appfwglobal
appfwglobal_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to appfwglobal
appfwglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to appfwglobal
appfwglobal_binding	Binding object showing the resources that can be bound to appfwglobal
appfwhtmlerrorpage	Configuration for HTML error page
appfwjsoncontenttype	Configuration for JSON content type
appfwlearningdata	Configuration for learning data
appfwlearningsettings	Configuration for learning settings
appfwpolicy	Configuration for application firewall policy
appfwpolicy_appfwglobal_binding	Binding object showing the appfwglobal that can be bound to appfwpolicy
appfwpolicy_appfwpolicylabel_binding	Binding object showing the appfwpolicylabel that can be bound to appfwpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicy_binding	
Name	**Description**
appfwpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to appfwpolicy
appfwpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to appfwpolicy
appfwpolicylabel	Configuration for application firewall policy label
appfwpolicylabel_appfwpolicy_binding	Binding object showing the appfwpolicy that can be bound to appfwpolicylabel
appfwpolicylabel_binding	Binding object showing the resources that can be bound to appfwpolicylabel
appfwpolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to appfwpolicylabel
appfwprofile	Configuration for application firewall profile
appfwprofile_binding	Binding object showing the resources that can be bound to appfwprofile
appfwprofile_contenttype_binding	Binding object showing the contenttype that can be bound to appfwprofile
appfwprofile_cookieconsistency_binding	Binding object showing the cookieconsistency that can be bound to appfwprofile
appfwprofile_creditcardnumber_binding	Binding object showing the creditcardnumber that can be bound to appfwprofile
appfwprofile_crosssitescripting_binding	Binding object showing the crosssitescripting that can be bound to appfwprofile
appfwprofile_csrftag_binding	Binding object showing the csrftag that can be bound to appfwprofile
appfwprofile_denyurl_binding	Binding object showing the denyurl that can be bound to appfwprofile
appfwprofile_excluderescontenttype_binding	Binding object showing the excluderescontenttype that can be bound to appfwprofile
appfwprofile_fieldconsistency_binding	Binding object showing the fieldconsistency that can be bound to appfwprofile
appfwprofile_fieldformat_binding	Binding object showing the fieldformat that can be bound to appfwprofile
appfwprofile_safeobject_binding	Binding object showing the safeobject that can be bound to appfwprofile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_sqlinjection_binding	
Name	**Description**
appfwprofile_starturl_binding	Binding object showing the starturl that can be bound to appfwprofile
appfwprofile_trustedlearningclients_binding	Binding object showing the trustedlearningclients that can be bound to appfwprofile
appfwprofile_xmlattachmenturl_binding	Binding object showing the xmlattachmenturl that can be bound to appfwprofile
appfwprofile_xmldosurl_binding	Binding object showing the xmldosurl that can be bound to appfwprofile
appfwprofile_xmlsqlinjection_binding	Binding object showing the xmlsqlinjection that can be bound to appfwprofile
appfwprofile_xmlvalidationurl_binding	Binding object showing the xmlvalidationurl that can be bound to appfwprofile
appfwprofile_xmlwsiurl_binding	Binding object showing the xmlwsiurl that can be bound to appfwprofile
appfwprofile_xmlxss_binding	Binding object showing the xmlxss that can be bound to appfwprofile
appfwsettings	Configuration for AS settings
appfwsignatures	Configuration for application firewall signatures XML configuration
appfwtransactionrecords	Configuration for Application firewall transaction record
appfwwsdl	Configuration for WSDL file
appfwxmlcontenttype	Configuration for XML Content type
appfwxmlerrorpage	Configuration for xml error page
appfwxmlschema	Configuration for XML schema

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwarchive

May 20, 2016

Configuration for archive resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of tar archive. Minimum length = 1 Maximum length = 31
target	<String>	Read-write	Path to the file to be exported. Minimum length = 1 Maximum length = 2047
src	<String>	Read-write	Indicates the source of the tar archive file as a URL of the form <protocol>://<host>[:<port>][/<path>] <protocol> is http or https. <host> is the DNS name or IP address of the http or https server. <port> is the port number of the server. If omitted, the default port for http or https will be used. <path> is the path of the file on the server. Import will fail if an https server requires client certificate authentication. . Minimum length = 1 Maximum length = 2047
comment	<String>	Read-write	Comments associated with this archive. Maximum length = 128
response	<String>	Read-only	.

Operations (click to see Properties)

EXPORT | IMPORT | DELETE | GET (ALL)

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

export

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwarchive?**action=export**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwarchive":{
    "name":<String_value>,
    "target":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Import

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwarchive?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwarchive":{
    "src":<String_value>,
    "name":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwarchive/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwarchive

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwarchive": [ {
    "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwconfidfield

May 20, 2016

Configuration for configured confidential form fields resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
fieldname	<String>	Read-write	Name of the form field to designate as confidential. Minimum length = 1
url	<String>	Read-write	URL of the web page that contains the web form. Minimum length = 1
isregex	<String>	Read-write	Method of specifying the form field name. Available settings function as follows: * REGEX. Form field is a regular expression. * NOTREGEX. Form field is a literal string. Default value: NOTREGEX Possible values = REGEX, NOTREGEX
comment	<String>	Read-write	Any comments to preserve information about the form field designation.
state	<String>	Read-write	Enable or disable the confidential field designation. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwconfidfield

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwconfidfield":{
    "fieldname":<String_value>,
    "url":<String_value>,
    "isregex":<String_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwconfidfield/fieldname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwconfidfield

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwconfidfield":{
    "fieldname":<String_value>,
    "url":<String_value>,
    "comment":<String_value>,
    "isregex":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwconfidfield?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwconfidfield":{
    "fieldname":<String_value>,
    "url":<String_value>,
    "comment":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "isregex":true,
    "state":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwconfidfield

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwconfidfield?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwconfidfield?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwconfidfield resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwconfidfield?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwconfidfield?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwconfidfield resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwconfidfield": [ {
      "fieldname":<String_value>,
      "url":<String_value>,
      "isregex":<String_value>,
      "comment":<String_value>,
      "state":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwconfidfield?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwconfidfield": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwcustomsettings

Jul 07, 2017

Configuration for application firewall custom settings XML configuration resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	. Minimum length = 1 Maximum length = 31
target	<String>	Read-write	. Minimum length = 1 Maximum length = 2047

Operations (click to see Properties)

EXPORT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

export

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwcustomsettings?**action=export**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwcustomsettings":{
    "name":<String_value>,
    "target":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwfieldtype

May 20, 2016

Configuration for application firewall form field type resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the field type. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the field type is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my field type" or 'my field type'). Minimum length = 1
regex	<String>	Read-write	PCRE - format regular expression defining the characters and length allowed for this field type. Minimum length = 1
priority	<Double>	Read-write	Positive integer specifying the priority of the field type. A lower number specifies a higher priority. Field types are checked in the order of their priority numbers. Minimum value = 0 Maximum value = 64000
comment	<String>	Read-write	Comment describing the type of field that this field type is intended to match.
builtin	<String[]>	Read-only	Flag to determine if fieldtype is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwfieldtype":{
    "name":<String_value>,
    "regex":<String_value>,
    "priority":<Double_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwfieldtype":{
    "name":<String_value>,
    "regex":<String_value>,
    "priority":<Double_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwfieldtype resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwfieldtype resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwfieldtype": [ {
      "name":<String_value>,
      "regex":<String_value>,
      "priority":<Double_value>,
      "comment":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwfieldtype": [ {
      "name":<String_value>,
      "regex":<String_value>,
      "priority":<Double_value>,
      "comment":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwfieldtype?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "appfwfieldtype": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwglobal_appfwpolicy_binding

May 20, 2016

Binding object showing the appfwpolicy that can be bound to appfwglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the policy.
labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
state	<String>	Read-write	Enable or disable the binding to activate or deactivate the policy. This is applicable to classic policies only. Possible values = ENABLED, DISABLED
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
type	<String>	Read-write	Bind point to which to policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, NONE
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = reqvserver, policylabel
flowtype	<Double>	Read-only	flowtype of the bound application firewall policy.
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwglobal_appfwpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwglobal_appfwpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "state":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_appfwpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_appfwpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_appfwpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwglobal_appfwpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_appfwpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwglobal_appfwpolicy_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwglobal_appfwpolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "state":<String_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "flowtype":<Double_value>,
    "numpol":<Double_value>,
    "policytype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_appfwpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "appfwglobal_appfwpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwglobal_auditnslogpolicy_binding

May 20, 2016

Binding object showing the auditnslogpolicy that can be bound to appfwglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	Name of the policy.
labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
state	<String>	Read-write	Enable or disable the binding to activate or deactivate the policy. This is applicable to classic policies only. Possible values = ENABLED, DISABLED
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is smaller than the current policy's priority number. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = REQ_OVERRIDE, REQ_DEFAULT, NONE
labeltype	<String>	Read-write	Type of policy label to invoke if the current policy evaluates to TRUE and the invoke parameter is set. Available settings function as follows: * reqvserver. Invoke the unnamed policy label associated with the specified request virtual server. * policylabel. Invoke the specified user-defined policy label. Possible values = reqvserver, policylabel
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwglobal_auditnslogpolicy_binding

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwglobal_auditnslogpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "state":<String_value>,
    "gotoprioritiyexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of appfwglobal_auditnslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwglobal_auditnslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwglobal_auditnslogpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "state":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "type":<String_value>,
    "labeltype":<String_value>,
    "policytype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_auditnslogpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "appfwglobal_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwglobal_auditsyslogpolicy_binding

May 20, 2016

Binding object showing the auditsyslogpolicy that can be bound to appfwglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	Name of the policy.
labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
state	<String>	Read-write	Enable or disable the binding to activate or deactivate the policy. This is applicable to classic policies only. Possible values = ENABLED, DISABLED
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is smaller than the current policy's priority number. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
type	<String>	Read-write	Bind point to which to policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, NONE
			If the current policy evaluates to TRUE, terminate evaluation of policies bound

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			server or evaluate the specified policy label.
labeltype	<String>	Read-write	Type of policy label to invoke if the current policy evaluates to TRUE and the invoke parameter is set. Available settings function as follows: * reqvserver. Invoke the unnamed policy label associated with the specified request virtual server. * policylabel. Invoke the specified user-defined policy label. Possible values = reqvserver, policylabel
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwglobal_auditsyslogpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "state":<String_value>,
    "gotoprioritiyexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of appfwglobal_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwglobal_auditsyslogpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "state":<String_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "policytype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "appfwglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwglobal_binding

May 20, 2016

Binding object which returns the resources bound to appfwglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
appfwglobal_auditsyslogpolicy_binding	<appfwglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to appfwglobal.
appfwglobal_auditnslogpolicy_binding	<appfwglobal_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to appfwglobal.
appfwglobal_appfwpolicy_binding	<appfwglobal_appfwpolicy_binding[]>	Read-only	appfwpolicy that can be bound to appfwglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "appfwglobal_binding": [ {
      "appfwglobal_appfwpolicy_binding":<appfwglobal_appfwpolicy_binding[]_value>,
      "appfwglobal_auditsyslogpolicy_binding":<appfwglobal_auditsyslogpolicy_binding[]_value>,
      "appfwglobal_auditnslogpolicy_binding":<appfwglobal_auditnslogpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwhtmlerrorpage

Jul 07, 2017

Configuration for HTML error page resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the XML error object to remove. Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL (protocol, host, path, and name) for the location at which to store the imported HTML error object. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
comment	<String>	Read-write	Any comments to preserve information about the HTML error object. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrite any existing HTML error object of the same name.
response	<String>	Read-only	.

Operations (click to see Properties)

DELETE | IMPORT | CHANGE | GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwhtmlerrorpage/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwhtmlerrorpage?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwhtmlerrorpage":{
    "src":<String_value>,
    "name":<String_value>,
    "comment":<String_value>,
    "overwrite":<Boolean_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

change

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwhtmlerrorpage?**action=update**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwhtmlerrorpage":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwhtmlerrorpage

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "appfwhtmlerrorpage": [ {
    "name":<String_value>,
    "response":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwhtmlerrorpage/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwhtmlerrorpage/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwhtmlerrorpage/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwhtmlerrorpage": [ {
    "name":<String_value>,
    "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwjsoncontenttype

May 20, 2016

Configuration for JSON content type resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
jsoncontenttypevalue	<String>	Read-write	Content type to be classified as JSON. Minimum length = 1
isregex	<String>	Read-write	Is json content type a regular expression?. Default value: NOTREGEX Possible values = REGEX, NOTREGEX
builtin	<String[]>	Read-only	Flag to determine if jsoncontenttype is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwjsoncontenttype":{
    "jsoncontenttypevalue":<String_value>,
    "isregex":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype/jsoncontenttypevalue_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwjsoncontenttype resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwjsoncontenttype resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwjsoncontenttype": [ {
    "jsoncontenttypevalue":<String_value>,
    "isregex":<String_value>,
    "builtin":<String[]_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype/jsoncontenttypevalue_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype/jsoncontenttypevalue_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype/jsoncontenttypevalue_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwjsoncontenttype": [ {
      "jsoncontenttypevalue":<String_value>,
      "isregex":<String_value>,
      "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwjsoncontenttype?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwjsoncontenttype": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwlearningdata

Jul 07, 2017

Configuration for learning data resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
profilename	<String>	Read-write	Name of the profile.
starturl	<String>	Read-write	Start URL configuration. Minimum length = 1
cookieconsistency	<String>	Read-write	Cookie Name. Minimum length = 1
fieldconsistency	<String>	Read-write	Form field name. Minimum length = 1
formactionurl_ffc	<String>	Read-write	Form action URL.
contenttype	<String>	Read-write	Content Type Name. Minimum length = 1
crosssitescripting	<String>	Read-write	Cross-site scripting. Minimum length = 1
formactionurl_xss	<String>	Read-write	Form action URL. Minimum length = 1
as_scan_location_xss	<String>	Read-write	Location of cross-site scripting exception - form field, header or cookie. Possible values = FORMFIELD, HEADER, COOKIE
as_value_type_xss	<String>	Read-write	XSS value type. (Tag | Attribute | Pattern). Possible values = Tag, Attribute, Pattern
as_value_expr_xss	<String>	Read-write	XSS value expressions consistituting expressions for Tag, Attribute or Pattern.
sqlinjection	<String>	Read-write	Form field name. Minimum length = 1
formactionurl_sql	<String>	Read-write	Form action URL. Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = FORMFIELD, HEADER, COOKIE
as_value_type_sql	<String>	Read-write	SQL value type. Keyword, SpecialString or Wildchar. Possible values = Keyword, SpecialString, Wildchar
as_value_expr_sql	<String>	Read-write	SQL value expressions consistituting expressions for Keyword, SpecialString or Wildchar.
fieldformat	<String>	Read-write	Field format name. Minimum length = 1
formactionurl_ff	<String>	Read-write	Form action URL. Minimum length = 1
csrftag	<String>	Read-write	CSRF Form Action URL. Minimum length = 1
csrfformoriginurl	<String>	Read-write	CSRF Form Origin URL. Minimum length = 1
creditcardnumber	<String>	Read-write	The object expression that is to be excluded from safe commerce check. Minimum length = 1
creditcardnumberurl	<String>	Read-write	The url for which the list of credit card numbers are needed to be bypassed from inspection. Minimum length = 1
xmldoscheck	<String>	Read-write	XML Denial of Service check, one of MaxAttributes MaxAttributeNameLength MaxAttributeValueLength MaxElementNameLength MaxFileSize MinFileSize MaxCDATALength MaxElements MaxElementDepth MaxElementChildren NumDTDs NumProcessingInstructions NumExternalEntities MaxEntityExpansions MaxEntityExpansionDepth MaxNamespaces MaxNamespaceUriLength MaxSOAPArraySize MaxSOAPArrayRank . Minimum length = 1
xmlwsicheck	<String>	Read-write	Web Services Interoperability Rule ID. Minimum length = 1
			XML Attachment Content-Type.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
totalxmlrequests	<Boolean>	Read-write	Total XML requests.
securitycheck	<String>	Read-write	Name of the security check. Possible values = startURL, cookieConsistency, fieldConsistency, crossSiteScripting, SQLInjection, fieldFormat, CSRFtag, XMLDoSCheck, XMLWSICheck, XMLAttachmentCheck, TotalXMLRequests, creditCardNumber, ContentType
target	<String>	Read-write	Target filename for data to be exported. Minimum length = 1 Maximum length = 127
url	<String>	Read-only	Learnt url.
name	<String>	Read-only	Learnt field name.
fieldtype	<String>	Read-only	Learnt field type.
fieldformatminlength	<Double>	Read-only	The minimum allowed length for data in this form field.
fieldformatmaxlength	<Double>	Read-only	The maximum allowed length for data in this form field.
fieldformatcharmappcre	<String>	Read-only	Form field value allowed character map.
value_type	<String>	Read-only	Learnt field value type.
value	<String>	Read-only	Learnt field value.
hits	<Double>	Read-only	Learnt entity hit count.
data	<String>	Read-only	Learned data.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

DELETE | RESET | EXPORT | GET (ALL) | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwlearningdata

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

reset

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwlearningdata?**action=reset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"appfwlearningdata":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

export

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwlearningdata?**action=export**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwlearningdata":{
    "profilename":<String_value>,
    "securitycheck":<String_value>,
    "target":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwlearningdata

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningdata?**args=profilename:<String_value>,securitycheck:<String_value>,**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get appfwlearningdata resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningdata?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningdata?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwlearningdata resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningdata?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningdata?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwlearningdata resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwlearningdata": [ {
profilename:<String_value>,securitycheck:<String_value>,      "url":<String_value>,
      "name":<String_value>,
      "fieldtype":<String_value>,
      "fieldformatminlength":<Double_value>,
      "fieldformatmaxlength":<Double_value>,
      "fieldformatcharmappcre":<String_value>,
      "value_type":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "value":<String_value>,
    "hits":<Double_value>,
    "data":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwlearningdata?**args=profilename:<String_value>,securitycheck:<String_value>,&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwlearningdata": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwlearningsettings

May 20, 2016

Configuration for learning settings resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
profilename	<String>	Read-write	Name of the profile. Minimum length = 1
starturlminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn start URLs. Default value: 1 Minimum value = 1
starturlpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular start URL pattern for the learning engine to learn that start URL. Default value: 0 Minimum value = 0 Maximum value = 100
cookieconsistencyminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn cookies. Default value: 1 Minimum value = 1
cookieconsistencypercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular cookie pattern for the learning engine to learn that cookie. Default value: 0 Minimum value = 0 Maximum value = 100
csrftagminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn cross-site request forgery (CSRF) tags. Default value: 1 Minimum value = 1
csrftagpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular CSRF tag for the learning engine to learn that CSRF tag. Default value: 0 Minimum value = 0 Maximum value = 100

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
fieldconsistencyminthreshold	<Double>	Read-write	engine must observe to learn field consistency information. Default value: 1 Minimum value = 1
fieldconsistencypercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular field consistency pattern for the learning engine to learn that field consistency pattern. Default value: 0 Minimum value = 0 Maximum value = 100
crosssitescriptingminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn HTML cross-site scripting patterns. Default value: 1 Minimum value = 1
crosssitescriptingpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular cross-site scripting pattern for the learning engine to learn that cross-site scripting pattern. Default value: 0 Minimum value = 0 Maximum value = 100
sqlinjectionminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn HTML SQL injection patterns. Default value: 1 Minimum value = 1
sqlinjectionpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular HTML SQL injection pattern for the learning engine to learn that HTML SQL injection pattern. Default value: 0 Minimum value = 0 Maximum value = 100
fieldformatminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn field formats. Default value: 1 Minimum value = 1
fieldformatpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular web form field pattern for the learning engine to recommend a field format for that form field. Default value: 0 Minimum value = 0 Maximum value = 100
creditcardnumberminthreshold	<Double>	Read-write	Minimum threshold to learn Credit Card information. Default value: 1 Minimum value = 1
			Minimum threshold in percent to learn Credit Card information. Default value: 0

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum value = 100
contenttypeminthreshold	<Double>	Read-write	Minimum threshold to learn Content Type information. Default value: 1 Minimum value = 1
contenttypepercentthreshold	<Double>	Read-write	Minimum threshold in percent to learn Content Type information. Default value: 0 Minimum value = 0 Maximum value = 100
xmlwsiminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn web services interoperability (WSI) information. Default value: 1 Minimum value = 1
xmlwsipercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular pattern for the learning engine to learn a web services interoperability (WSI) pattern. Default value: 0 Minimum value = 0 Maximum value = 100
xmlattachmentminthreshold	<Double>	Read-write	Minimum number of application firewall sessions that the learning engine must observe to learn XML attachment patterns. Default value: 1 Minimum value = 1
xmlattachmentpercentthreshold	<Double>	Read-write	Minimum percentage of application firewall sessions that must contain a particular XML attachment pattern for the learning engine to learn that XML attachment pattern. Default value: 0 Minimum value = 0 Maximum value = 100
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwlearningsettings":{
    "profilename":<String_value>,
    "starturlminthreshold":<Double_value>,
    "starturlpercentthreshold":<Double_value>,
    "cookieconsistencyminthreshold":<Double_value>,
    "cookieconsistencypercentthreshold":<Double_value>,
    "csrftagminthreshold":<Double_value>,
    "csrftagpercentthreshold":<Double_value>,
    "fieldconsistencyminthreshold":<Double_value>,
    "fieldconsistencypercentthreshold":<Double_value>,
    "crosssitescriptingminthreshold":<Double_value>,
    "crosssitescriptingpercentthreshold":<Double_value>,
    "sqlinjectionminthreshold":<Double_value>,
    "sqlinjectionpercentthreshold":<Double_value>,
    "fieldformatminthreshold":<Double_value>,
    "fieldformatpercentthreshold":<Double_value>,
    "creditcardnumberminthreshold":<Double_value>,
    "creditcardnumberpercentthreshold":<Double_value>,
    "contenttypeminthreshold":<Double_value>,
    "contenttypepercentthreshold":<Double_value>,
    "xmlwsiminthreshold":<Double_value>,
    "xmlwsipercentthreshold":<Double_value>,
    "xmlattachmentminthreshold":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "xmlattachmentpercentthreshold":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwlearningsettings":{
    "profilename":<String_value>,
    "starturlminthreshold":true,
    "starturlpercentthreshold":true,
    "cookieconsistencyminthreshold":true,
    "cookieconsistencypercentthreshold":true,
    "csrftagminthreshold":true,
    "csrftagpercentthreshold":true,
    "fieldconsistencyminthreshold":true,
    "fieldconsistencypercentthreshold":true,
    "crosssitescriptingminthreshold":true,
    "crosssitescriptingpercentthreshold":true,
    "sqlinjectionminthreshold":true,
    "sqlinjectionpercentthreshold":true,
    "fieldformatminthreshold":true,
    "fieldformatpercentthreshold":true,
    "creditcardnumberminthreshold":true,
    "creditcardnumberpercentthreshold":true,
    "contenttypeminthreshold":true,
    "contenttypepercentthreshold":true,
    "xmlwsiminthreshold":true,
    "xmlwsipercentthreshold":true,
    "xmlattachmentminthreshold":true,
    "xmlattachmentpercentthreshold":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwlearningsettings resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwlearningsettings resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwlearningsettings": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "profilename":<String_value>,
    "starturlminthreshold":<Double_value>,
    "starturlpercentthreshold":<Double_value>,
    "cookieconsistencyminthreshold":<Double_value>,
    "cookieconsistencypercentthreshold":<Double_value>,
    "csrftagminthreshold":<Double_value>,
    "csrftagpercentthreshold":<Double_value>,
    "fieldconsistencyminthreshold":<Double_value>,
    "fieldconsistencypercentthreshold":<Double_value>,
    "crosssitescriptingminthreshold":<Double_value>,
    "crosssitescriptingpercentthreshold":<Double_value>,
    "sqlinjectionminthreshold":<Double_value>,
    "sqlinjectionpercentthreshold":<Double_value>,
    "fieldformatminthreshold":<Double_value>,
    "fieldformatpercentthreshold":<Double_value>,
    "creditcardnumberminthreshold":<Double_value>,
    "creditcardnumberpercentthreshold":<Double_value>,
    "contenttypeminthreshold":<Double_value>,
    "contenttypepercentthreshold":<Double_value>,
    "xmlwsiminthreshold":<Double_value>,
    "xmlwsipercentthreshold":<Double_value>,
    "xmlattachmentminthreshold":<Double_value>,
    "xmlattachmentpercentthreshold":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings/profilename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings/profilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings/profilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwlearningsettings": [ {
    "profilename":<String_value>,
    "starturlminthreshold":<Double_value>,
    "starturlpercentthreshold":<Double_value>,
    "cookieconsistencyminthreshold":<Double_value>,
    "cookieconsistencypercentthreshold":<Double_value>,
    "csrftagminthreshold":<Double_value>,
    "csrftagpercentthreshold":<Double_value>,
    "fieldconsistencyminthreshold":<Double_value>,
    "fieldconsistencypercentthreshold":<Double_value>,
    "crosssitescriptingminthreshold":<Double_value>,
    "crosssitescriptingpercentthreshold":<Double_value>,
    "sqlinjectionminthreshold":<Double_value>,
    "sqlinjectionpercentthreshold":<Double_value>,
    "fieldformatminthreshold":<Double_value>,
    "fieldformatpercentthreshold":<Double_value>,
    "creditcardnumberminthreshold":<Double_value>,
    "creditcardnumberpercentthreshold":<Double_value>,
    "contenttypeminthreshold":<Double_value>,
    "contenttypepercentthreshold":<Double_value>,
    "xmlwsiminthreshold":<Double_value>,
    "xmlwsipercentthreshold":<Double_value>,
    "xmlattachmentminthreshold":<Double_value>,
    "xmlattachmentpercentthreshold":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwlearningsettings?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "appfwlearningsettings": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicylabel_binding

May 20, 2016

Binding object which returns the resources bound to appfwpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the application firewall policy label.
appfwpolicylabel_policybinding_binding	<appfwpolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to appfwpolicylabel.
appfwpolicylabel_appfwpolicy_binding	<appfwpolicylabel_appfwpolicy_binding[]>	Read-only	appfwpolicy that can be bound to appfwpolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "appfwpolicylabel_binding": [ {
    "labelname":<String_value>,
    "appfwpolicylabel_policybinding_binding":<appfwpolicylabel_policybinding_binding[]_value>,
    "appfwpolicylabel_appfwpolicy_binding":<appfwpolicylabel_appfwpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicy

May 20, 2016

Configuration for application firewall policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the policy. Must begin with a letter, number, or the underscore character \(_\), and must contain only letters, numbers, and the hyphen \(-\), period \(.\) pound \(\#\), space \(\), at (@), equals \(=\), colon \(:\), and underscore characters. Can be changed after the policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks \(for example, "my policy" or 'my policy'\). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a NetScaler default syntax expression, that the policy uses to determine whether to filter the connection through the application firewall with the designated profile.
profilename	<String>	Read-write	Name of the application firewall profile to use if the policy matches. Minimum length = 1
comment	<String>	Read-write	Any comments to preserve information about the policy for later reference.
logaction	<String>	Read-write	Where to log information for connections that match this policy.
newname	<String>	Read-write	New name for the policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of Undef hits.
policytype	<String>	Read-only	.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
[illegible]	<Double>	[illegible]	[illegible]

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwpolicy":{
    "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "rule":<String_value>,
    "profilename":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "profilename":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwpolicy":{
    "name":<String_value>,
    "comment":true,
    "logaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwpolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "appfwpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "profilename":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "policytype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "appfwpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "profilename":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "policytype":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwwsdl

May 20, 2016

Configuration for WSDL file resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the WSDL file to remove. Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL (protocol, host, path, and name) of the WSDL file to be imported is stored. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
comment	<String>	Read-write	Any comments to preserve information about the WSDL. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrite any existing WSDL of the same name.
response	<String>	Read-only	.

Operations (click to see Properties)

DELETE | IMPORT | GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwwsdl/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwwsdl?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwwsdl":{
    "src":<String_value>,
    "name":<String_value>,
    "comment":<String_value>,
    "overwrite":<Boolean_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwwsdl

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwwsdl": [ {
      "name":<String_value>,
      "response":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwwsdl/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwwsdl/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwwsdl/name_value<String>?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwwsdl": [ {
    "name":<String_value>,
    "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicylabel

May 20, 2016

Configuration for application firewall policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name for the policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the policy label is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy label" or 'my policy label').
policylabeltype	<String>	Read-write	Type of transformations allowed by the policies bound to the label. Always http_req for application firewall policy labels. Possible values = http_req
newname	<String>	Read-write	The new name of the application firewall policylabel. Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Positive integer specifying the priority of the policy. A lower number specifies a higher priority. Must be unique within a group of policies that are bound to the same bind point or label. Policies are evaluated in the order of their priority numbers.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label to invoke if the current policy evaluates to TRUE and the invoke parameter is set. Available settings function as follows: * reqvserver. Invoke the unnamed policy label associated with the specified request virtual server. * policylabel. Invoke the specified user-defined policy label. Possible values = reqvserver, policylabel
invoke_labelname	<String>	Read-only	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

description	<String>	Read-only	Description of the policylabel.
Name	**Data Type**	**Permissions**	**Description**
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"appfwpolicylabel":{
    "labelname":<String_value>,
    "policylabeltype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwpolicylabel":{
    "labelname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwpolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwpolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "appfwpolicylabel": [ {
    "labelname":<String_value>,
    "policylabeltype":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "description":<String_value>,
    "policytype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "appfwpolicylabel": [ {
      "labelname":<String_value>,
      "policylabeltype":<String_value>,
      "numpol":<Double_value>,
      "hits":<Double_value>,
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "labeltype":<String_value>,
      "invoke_labelname":<String_value>,
      "description":<String_value>,
      "policytype":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicylabel": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicylabel_appfwpolicy_binding

May 20, 2016

Binding object showing the appfwpolicy that can be bound to appfwpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Positive integer specifying the priority of the policy. A lower number specifies a higher priority. Must be unique within a group of policies that are bound to the same bind point or label. Policies are evaluated in the order of their priority numbers.
gotoprioritiyexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the application firewall policy to bind to the policy label.
labeltype	<String>	Read-write	Type of policy label to invoke if the current policy evaluates to TRUE and the invoke parameter is set. Available settings function as follows: * reqvserver. Invoke the unnamed policy label associated with the specified request virtual server. * policylabel. Invoke the specified user-defined policy label. Possible values = reqvserver, policylabel
labelname	<String>	Read-write	Name of the application firewall policy label.
invoke_labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwpolicylabel_appfwpolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicylabel_appfwpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwpolicylabel_appfwpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "appfwpolicylabel_appfwpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel_appfwpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwpolicylabel_appfwpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicylabel_policybinding_binding

May 20, 2016

Binding object showing the policybinding that can be bound to appfwpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Positive integer specifying the priority of the policy. A lower number specifies a higher priority. Must be unique within a group of policies that are bound to the same bind point or label. Policies are evaluated in the order of their priority numbers.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the application firewall policy to bind to the policy label.
labeltype	<String>	Read-write	Type of policy label to invoke if the current policy evaluates to TRUE and the invoke parameter is set. Available settings function as follows: * reqvserver. Invoke the unnamed policy label associated with the specified request virtual server. * policylabel. Invoke the specified user-defined policy label. Possible values = reqvserver, policylabel
labelname	<String>	Read-write	Name of the application firewall policy label.
invoke_labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicylabel_policybinding_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwpolicylabel_policybinding_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicylabel_policybinding_binding": [ {
      "priority":<Double_value>,
      "gotoprioritiyexpression":<String_value>,
      "policyname":<String_value>,
      "labeltype":<String_value>,
      "labelname":<String_value>,
      "invoke_labelname":<String_value>,
      "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwpolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicy_appfwglobal_binding

Jul 07, 2017

Binding object showing the appfwglobal that can be bound to appfwpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_appfwglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_appfwglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicy_appfwglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_appfwglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwpolicy_appfwglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicy_appfwglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_appfwglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwpolicy_appfwglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicy_appfwpolicylabel_binding

May 20, 2016

Binding object showing the appfwpolicylabel that can be bound to appfwpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_appfwpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_appfwpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicy_appfwpolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_appfwpolicylabel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwpolicy_appfwpolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicy_appfwpolicylabel_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_appfwpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwpolicy_appfwpolicylabel_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to appfwpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
appfwpolicy_appfwglobal_binding	<appfwpolicy_appfwglobal_binding[]>	Read-only	appfwglobal that can be bound to appfwpolicy.
appfwpolicy_lbvserver_binding	<appfwpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to appfwpolicy.
appfwpolicy_csvserver_binding	<appfwpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to appfwpolicy.
appfwpolicy_appfwpolicylabel_binding	<appfwpolicy_appfwpolicylabel_binding[]>	Read-only	appfwpolicylabel that can be bound to appfwpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "appfwpolicy_binding": [ {
      "name":<String_value>,
      "appfwpolicy_appfwpolicylabel_binding":<appfwpolicy_appfwpolicylabel_binding[]_value>,
      "appfwpolicy_csvserver_binding":<appfwpolicy_csvserver_binding[]_value>,
      "appfwpolicy_appfwglobal_binding":<appfwpolicy_appfwglobal_binding[]_value>,
      "appfwpolicy_lbvserver_binding":<appfwpolicy_lbvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicy_csvserver_binding

Jul 07, 2017

Binding object showing the csvserver that can be bound to appfwpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicy_csvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to appfwpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicy_lbvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile

May 20, 2016

Configuration for application firewall profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the profile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.), pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) characters. Cannot be changed after the profile is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my profile" or 'my profile'). Minimum length = 1
defaults	<String>	Read-write	Default configuration to apply to the profile. Basic defaults are intended for standard content that requires little further configuration, such as static web site content. Advanced defaults are intended for specialized content that requires significant specialized configuration, such as heavily scripted or dynamic content. CLI users: When adding an application firewall profile, you can set either the defaults or the type, but not both. To set both options, create the profile by using the add appfw profile command, and then use the set appfw profile command to configure the other option. Possible values = basic, advanced
starturlaction	<String[]>	Read-write	One or more Start URL actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -startURLaction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = none, block, learn, log, stats
contenttypeaction	<String[]>	Read-write	One or more Content-type actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -contentTypeaction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -contentTypeaction none". Possible values = none, block, learn, log, stats
inspectcontenttypes	<String[]>	Read-write	One or more InspectContentType lists. * application/x-www-form-urlencoded * multipart/form-data * text/x-gwt-rpc CLI users: To enable, type "set appfw profile -InspectContentTypes" followed by the content types to be inspected. Possible values = none, application/x-www-form-urlencoded, multipart/form-data, text/x-gwt-rpc
starturlclosure	<String>	Read-write	Toggle the state of Start URL Closure. Default value: OFF Possible values = ON, OFF
denyurlaction	<String[]>	Read-write	One or more Deny URL actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. NOTE: The Deny URL check takes precedence over the Start URL check. If you enable blocking for the Deny URL check, the application firewall blocks any URL that is explicitly blocked by a Deny URL, even if the same URL would otherwise be allowed by the Start URL check. CLI users: To enable one or more actions, type "set appfw profile -denyURLaction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -denyURLaction none". Possible values = none, block, learn, log, stats
			Enable validation of Referer headers. Referer validation ensures that a web form that a user sends to your web site originally came from your web site,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
refererheadercheck	<String>	Read-write	Although this parameter is part of the Start URL check, referer validation protects against cross-site request forgery (CSRF) attacks, not Start URL attacks. Default value: OFF Possible values = OFF, if_present, AlwaysExceptStartURLs, AlwaysExceptFirstRequest
cookieconsistencyaction	<String[]>	Read-write	One or more Cookie Consistency actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -cookieConsistencyAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -cookieConsistencyAction none". Default value: none Possible values = none, block, learn, log, stats
cookietransforms	<String>	Read-write	Perform the specified type of cookie transformation. Available settings function as follows: * Encryption - Encrypt cookies. * Proxying - Mask contents of server cookies by sending proxy cookie to users. * Cookie flags - Flag cookies as HTTP only to prevent scripts on user's browser from accessing and possibly modifying them. CAUTION: Make sure that this parameter is set to ON if you are configuring any cookie transformations. If it is set to OFF, no cookie transformations are performed regardless of any other settings. Default value: OFF Possible values = ON, OFF
cookieencryption	<String>	Read-write	Type of cookie encryption. Available settings function as follows: * None - Do not encrypt cookies. * Decrypt Only - Decrypt encrypted cookies, but do not encrypt cookies. * Encrypt Session Only - Encrypt session cookies, but not permanent cookies. * Encrypt All - Encrypt all cookies. Default value: none Possible values = none, decryptOnly, encryptSessionOnly, encryptAll
cookieproxying	<String>	Read-write	Cookie proxy setting. Available settings function as follows: * None - Do not proxy cookies. * Session Only - Proxy session cookies by using the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: none Possible values = none, sessionOnly
addcookieflags	<String>	Read-write	Add the specified flags to cookies. Available settings function as follows: * None - Do not add flags to cookies. * HTTP Only - Add the HTTP Only flag to cookies, which prevents scripts from accessing cookies. * Secure - Add Secure flag to cookies. * All - Add both HTTPOnly and Secure flags to cookies. Default value: none Possible values = none, httpOnly, secure, all
fieldconsistencyaction	<String[]>	Read-write	One or more Form Field Consistency actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -fieldConsistencyaction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -fieldConsistencyAction none". Default value: none Possible values = none, block, learn, log, stats
csrftagaction	<String[]>	Read-write	One or more Cross-Site Request Forgery (CSRF) Tagging actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -CSRFTagAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -CSRFTagAction none". Default value: none Possible values = none, block, learn, log, stats
crosssitescriptingaction	<String[]>	Read-write	One or more Cross-Site Scripting (XSS) actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			profile -crossSiteScriptingAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -crossSiteScriptingAction none". Possible values = none, block, learn, log, stats
crosssitescriptingtransformunsafehtml	<String>	Read-write	Transform cross-site scripts. This setting configures the application firewall to disable dangerous HTML instead of blocking the request. CAUTION: Make sure that this parameter is set to ON if you are configuring any cross-site scripting transformations. If it is set to OFF, no cross-site scripting transformations are performed regardless of any other settings. Default value: OFF Possible values = ON, OFF
crosssitescriptingcheckcompleteurls	<String>	Read-write	Check complete URLs for cross-site scripts, instead of just the query portions of URLs. Default value: OFF Possible values = ON, OFF
sqlinjectionaction	<String[]>	Read-write	One or more HTML SQL Injection actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -SQLInjectionAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -SQLInjectionAction none". Possible values = none, block, learn, log, stats
sqlinjectiontransformspecialchars	<String>	Read-write	Transform injected SQL code. This setting configures the application firewall to disable SQL special strings instead of blocking the request. Since most SQL servers require a special string to activate an SQL keyword, in most cases a request that contains injected SQL code is safe if special strings are disabled. CAUTION: Make sure that this parameter is set to ON if you are configuring any SQL injection transformations. If it is set to OFF, no SQL injection transformations are performed regardless of any other settings. Default value: OFF Possible values = ON, OFF
sqlinjectiononlycheckfieldswithsqlchars	<String>	Read-write	Check only form fields that contain SQL special strings (characters) for injected SQL code. Most SQL servers require a special string to activate an SQL request, so SQL code without a special string is harmless to most SQL servers. Default value: ON

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sqlinjectiontype	<String>	Read-write	Available SQL injection types. -SQLSplChar : Checks for SQL Special Chars -SQLKeyword : Checks for SQL Keywords -SQLSplCharANDKeyword : Checks for both and blocks if both are found -SQLSplCharORKeyword : Checks for both and blocks if anyone is found. Default value: SQLSplCharANDKeyword Possible values = SQLSplChar, SQLKeyword, SQLSplCharORKeyword, SQLSplCharANDKeyword
sqlinjectionchecksqlwildchars	<String>	Read-write	Check for form fields that contain SQL wild chars . Default value: OFF Possible values = ON, OFF
fieldformataction	<String[]>	Read-write	One or more Field Format actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of suggested web form fields and field format assignments. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -fieldFormatAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -fieldFormatAction none". Possible values = none, block, learn, log, stats
defaultfieldformattype	<String>	Read-write	Designate a default field type to be applied to web form fields that do not have a field type explicitly assigned to them. Minimum length = 1
defaultfieldformatminlength	<Double>	Read-write	Minimum length, in characters, for data entered into a field that is assigned the default field type. To disable the minimum and maximum length settings and allow data of any length to be entered into the field, set this parameter to zero (0). Default value: 0 Minimum value = 0 Maximum value = 65535
defaultfieldformatmaxlength	<Double>	Read-write	Maximum length, in characters, for data entered into a field that is assigned the default field type. Default value: 65535 Minimum value = 1 Maximum value = 65535
			One or more Buffer Overflow actions. Available settings function as follows: * Block - Block connections that violate this security check.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
bufferoverflowaction	<String[]>	Read-write	* Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -bufferOverflowAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -bufferOverflowAction none". Possible values = none, block, learn, log, stats
bufferoverflowmaxurllength	<Double>	Read-write	Maximum length, in characters, for URLs on your protected web sites. Requests with longer URLs are blocked. Default value: 1024 Minimum value = 0 Maximum value = 65535
bufferoverflowmaxheaderlength	<Double>	Read-write	Maximum length, in characters, for HTTP headers in requests sent to your protected web sites. Requests with longer headers are blocked. Default value: 4096 Minimum value = 0 Maximum value = 65535
bufferoverflowmaxcookielength	<Double>	Read-write	Maximum length, in characters, for cookies sent to your protected web sites. Requests with longer cookies are blocked. Default value: 4096 Minimum value = 0 Maximum value = 65535
creditcardaction	<String[]>	Read-write	One or more Credit Card actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -creditCardAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -creditCardAction none". Default value: none Possible values = none, block, learn, log, stats
creditcard	<String[]>	Read-write	Credit card types that the application firewall should protect. Possible values = visa, mastercard, discover, amex, jcb, dinersclub
creditcardmaxallowed	<Double>	Read-write	Maximum number of credit card numbers that can appear on a web page served by your protected web sites. Pages that contain more credit card numbers are blocked if 'block' action is enabled. It doesn't apply to x-out. Minimum value = 0

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
creditcardxout	<String>	Read-write	Mask any credit card number detected in a response by replacing each digit, except the digits in the final group, with the letter "X.". Default value: OFF Possible values = ON, OFF
dosecurecreditcardlogging	<String>	Read-write	Setting this option logs credit card numbers in the response when the match is found. Default value: ON Possible values = ON, OFF
streaming	<String>	Read-write	Setting this option converts content-length form submission requests (requests with content-type "application/x-www-form-urlencoded" or "multipart/form-data") to chunked requests when atleast one of the following protections : SQL injection protection, XSS protection, form field consistency protection, starturl closure, CSRF tagging is enabled. Please make sure that the backend server accepts chunked requests before enabling this option. Default value: OFF Possible values = ON, OFF
trace	<String>	Read-write	Toggle the state of trace. Default value: OFF Possible values = ON, OFF
requestcontenttype	<String>	Read-write	Default Content-Type header for requests. A Content-Type header can contain 0-255 letters, numbers, and the hyphen (-) and underscore (_) characters. Minimum length = 1 Maximum length = 255
responsecontenttype	<String>	Read-write	Default Content-Type header for responses. A Content-Type header can contain 0-255 letters, numbers, and the hyphen (-) and underscore (_) characters. Minimum length = 1 Maximum length = 255
xmldosaction	<String[]>	Read-write	One or more XML Denial-of-Service (XDoS) actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLDoSAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = none, block, learn, log, stats
xmlformataction	<String[]>	Read-write	One or more XML Format actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLFormatAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLFormatAction none". Possible values = none, block, learn, log, stats
xmlsqlinjectionaction	<String[]>	Read-write	One or more XML SQL Injection actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLSQLInjectionAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLSQLInjectionAction none". Possible values = none, block, learn, log, stats
xmlsqlinjectiononlycheckfieldswithsqlchars	<String>	Read-write	Check only form fields that contain SQL special characters, which most SQL servers require before accepting an SQL command, for injected SQL. Default value: ON Possible values = ON, OFF
xmlsqlinjectiontype	<String>	Read-write	Available SQL injection types. -SQLSplChar : Checks for SQL Special Chars -SQLKeyword : Checks for SQL Keywords -SQLSplCharANDKeyword : Checks for both and blocks if both are found -SQLSplCharORKeyword : Checks for both and blocks if anyone is found. Default value: SQLSplCharANDKeyword Possible values = SQLSplChar, SQLKeyword, SQLSplCharORKeyword, SQLSplCharANDKeyword
xmlsqlinjectionchecksqlwildchars	<String>	Read-write	Check for form fields that contain SQL wild chars . Default value: OFF Possible values = ON, OFF
			Parse comments in XML Data and exempt those sections of the request that are from the XML SQL Injection check. You must configure the type of comments that the application firewall is to detect and exempt from this security check. Available settings function as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
xmlsqlinjectionparsecomments	<String>	Read-write	* ANSI - Exempt content that is part of an ANSI (Mozilla-style) comment. * Nested - Exempt content that is part of a nested (Microsoft-style) comment. * ANSI Nested - Exempt content that is part of any type of comment. Default value: checkall Possible values = checkall, ansi, nested, ansinested
xmlxssaction	<String[]>	Read-write	One or more XML Cross-Site Scripting actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLXSSAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLXSSAction none". Possible values = none, block, learn, log, stats
xmlwsiaction	<String[]>	Read-write	One or more Web Services Interoperability (WSI) actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLWSIAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLWSIAction none". Possible values = none, block, learn, log, stats
xmlattachmentaction	<String[]>	Read-write	One or more XML Attachment actions. Available settings function as follows: * Block - Block connections that violate this security check. * Learn - Use the learning engine to generate a list of exceptions to this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLAttachmentAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLAttachmentAction none". Possible values = none, block, learn, log, stats
			One or more XML Validation actions. Available settings function as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
xmlvalidationaction	<String[]>	Read-write	* Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLValidationAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLValidationAction none". Possible values = none, block, learn, log, stats
xmlerrorobject	<String>	Read-write	Name to assign to the XML Error Object, which the application firewall displays when a user request is blocked. Must begin with a letter, number, or the underscore character \(_\), and must contain only letters, numbers, and the hyphen \(-\), period \(.\) pound \(\#\), space \(\), at (@), equals \(=\), colon \(:\), and underscore characters. Cannot be changed after the XML error object is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks \(for example, "my XML error object" or 'my XML error object'\). Minimum length = 1
customsettings	<String>	Read-write	Object name for custom settings. This check is applicable to Profile Type: HTML, XML. . Minimum length = 1
signatures	<String>	Read-write	Object name for signatures. This check is applicable to Profile Type: HTML, XML. . Minimum length = 1
xmlsoapfaultaction	<String[]>	Read-write	One or more XML SOAP Fault Filtering actions. Available settings function as follows: * Block - Block connections that violate this security check. * Log - Log violations of this security check. * Stats - Generate statistics for this security check. * None - Disable all actions for this security check. * Remove - Remove all violations for this security check. CLI users: To enable one or more actions, type "set appfw profile -XMLSOAPFaultAction" followed by the actions to be enabled. To turn off all actions, type "set appfw profile -XMLSOAPFaultAction none". Possible values = none, block, log, remove, stats
usehtmlerrorobject	<String>	Read-write	Send an imported HTML Error object to a user when a request is blocked, instead of redirecting the user to the designated Error URL. Default value: OFF Possible values = ON, OFF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
htmlerrorobject	<String>	Read-write	Name to assign to the HTML Error Object. Must begin with a letter, number, or the underscore character \(_\), and must contain only letters, numbers, and the hyphen \(-\), period \(.\) pound \(\#\), space \(\), at (@), equals \(=\), colon \(:\), and underscore characters. Cannot be changed after the HTML error object is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks \(for example, "my HTML error object" or 'my HTML error object'\). Minimum length = 1
logeverypolicyhit	<String>	Read-write	Log every profile match, regardless of security checks results. Default value: OFF Possible values = ON, OFF
stripcomments	<String>	Read-write	Strip HTML comments. This check is applicable to Profile Type: HTML. . Default value: OFF Possible values = ON, OFF
striphtmlcomments	<String>	Read-write	Strip HTML comments before forwarding a web page sent by a protected web site in response to a user request. Default value: none Possible values = none, all, exclude_script_tag
stripxmlcomments	<String>	Read-write	Strip XML comments before forwarding a web page sent by a protected web site in response to a user request. Default value: none Possible values = none, all
exemptclosureurlsfromsecuritychecks	<String>	Read-write	Exempt URLs that pass the Start URL closure check from SQL injection, cross-site script, field format and field consistency security checks at locations other than headers. Default value: ON Possible values = ON, OFF
defaultcharset	<String>	Read-write	Default character set for protected web pages. Web pages sent by your protected web sites in response to user requests are assigned this character set if the page does not already specify a character set. The character sets supported by the application firewall are: * iso-8859-1 (English US) * big5 (Chinese Traditional) * gb2312 (Chinese Simplified) * sjis (Japanese Shift-JIS) * euc-jp (Japanese EUC-JP)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			* utf-8 (Unicode) * euc-kr (Korean). Minimum length = 1 Maximum length = 31
postbodylimit	<Double>	Read-write	Maximum allowed HTTP post body size, in bytes. Default value: 20000000
fileuploadmaxnum	<Double>	Read-write	Maximum allowed number of file uploads per form-submission request. The maximum setting (65535) allows an unlimited number of uploads. Default value: 65535 Minimum value = 0 Maximum value = 65535
canonicalizehtmlresponse	<String>	Read-write	Perform HTML entity encoding for any special characters in responses sent by your protected web sites. Default value: ON Possible values = ON, OFF
enableformtagging	<String>	Read-write	Enable tagging of web form fields for use by the Form Field Consistency and CSRF Form Tagging checks. Default value: ON Possible values = ON, OFF
sessionlessfieldconsistency	<String>	Read-write	Perform sessionless Field Consistency Checks. Default value: OFF Possible values = OFF, ON, postOnly
sessionlessurlclosure	<String>	Read-write	Enable session less URL Closure Checks. This check is applicable to Profile Type: HTML. . Default value: OFF Possible values = ON, OFF
semicolonfieldseparator	<String>	Read-write	Allow ';' as a form field separator in URL queries and POST form bodies. . Default value: OFF Possible values = ON, OFF
excludefileuploadfromchecks	<String>	Read-write	Exclude uploaded files from Form checks. Default value: OFF Possible values = ON, OFF
sqlinjectionparsecomments	<String>	Read-write	Parse HTML comments and exempt them from the HTML SQL Injection check. You must specify the type of comments that the application firewall is to detect and exempt from this security check. Available settings function as follows: * Check all - Check all content. * ANSI - Exempt content that is part of an ANSI (Mozilla-style) comment. * Nested - Exempt content that is part of a nested (Microsoft-style) comment. * ANSI Nested - Exempt content that is part of any type of

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = checkall, ansi, nested, ansinested
invalidpercenthandling	<String>	Read-write	Configure the method that the application firewall uses to handle percent-encoded names and values. Available settings function as follows: * apache_mode - Apache format. * asp_mode - Microsoft ASP format. * secure_mode - Secure format. Default value: secure_mode Possible values = apache_mode, asp_mode, secure_mode
type	<String[]>	Read-write	Application firewall profile type, which controls which security checks and settings are applied to content that is filtered with the profile. Available settings function as follows: * HTML - HTML-based web sites. * XML - XML-based web sites and services. * HTML XML (Web 2.0) - Sites that contain both HTML and XML content, such as ATOM feeds, blogs, and RSS feeds. Default value: HTML Possible values = HTML, XML
checkrequestheaders	<String>	Read-write	Check request headers as well as web forms for injected SQL and cross-site scripts. Default value: OFF Possible values = ON, OFF
optimizepartialreqs	<String>	Read-write	Optimize handle of HTTP partial requests i.e. those with range headers. Available settings are as follows: * ON - Partial requests by the client result in partial requests to the backend server in most cases. * OFF - Partial requests by the client are changed to full requests to the backend server. Default value: ON Possible values = ON, OFF
urldecoderequestcookies	<String>	Read-write	URL Decode request cookies before subjecting them to SQL and cross-site scripting checks. Default value: OFF Possible values = ON, OFF
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
archivename	<String>	Read-write	Source for tar archive. Minimum length = 1 Maximum length = 31
csrftag	<String>	Read-only	The web form originating URL.
state	<String>	Read-only	Enabled. Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

builtin **Name**	<Boolean> **Data Type**	Read-only **Permissions**	Indicates that a profile is a built-in entity. **Description**
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RESTORE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"appfwprofile":{
    "name":<String_value>,
    "defaults":<String_value>,
    "starturlaction":<String[]_value>,
    "contenttypeaction":<String[]_value>,
    "inspectcontenttypes":<String[]_value>,
    "starturlclosure":<String_value>,
    "denyurlaction":<String[]_value>,
    "refererheadercheck":<String_value>,
    "cookieconsistencyaction":<String[]_value>,
    "cookietransforms":<String_value>,
    "cookieencryption":<String_value>,
    "cookieproxying":<String_value>,
    "addcookieflags":<String_value>,
    "fieldconsistencyaction":<String[]_value>,
    "csrftagaction":<String[]_value>,
    "crosssitescriptingaction":<String[]_value>,
    "crosssitescriptingtransformunsafehtml":<String_value>,
    "crosssitescriptingcheckcompleteurls":<String_value>,
    "sqlinjectionaction":<String[]_value>,
    "sqlinjectiontransformspecialchars":<String_value>,
    "sqlinjectiononlycheckfieldswithsqlchars":<String_value>,
    "sqlinjectiontype":<String_value>,
    "sqlinjectionchecksqlwildchars":<String_value>,
    "fieldformataction":<String[]_value>,
    "defaultfieldformattype":<String_value>,
    "defaultfieldformatminlength":<Double_value>,
    "defaultfieldformatmaxlength":<Double_value>,
    "bufferoverflowaction":<String[]_value>,
    "bufferoverflowmaxurllength":<Double_value>,
    "bufferoverflowmaxheaderlength":<Double_value>,
    "bufferoverflowmaxcookielength":<Double_value>,
    "creditcardaction":<String[]_value>,
    "creditcard":<String[]_value>,
    "creditcardmaxallowed":<Double_value>,
    "creditcardxout":<String_value>,
    "dosecurecreditcardlogging":<String_value>,
    "streaming":<String_value>,
    "trace":<String_value>,
    "requestcontenttype":<String_value>,
    "responsecontenttype":<String_value>,
    "xmldosaction":<String[]_value>,
    "xmlformataction":<String[]_value>,
    "xmlsqlinjectionaction":<String[]_value>,
    "xmlsqlinjectiononlycheckfieldswithsqlchars":<String_value>,
    "xmlsqlinjectiontype":<String_value>,
    "xmlsqlinjectionchecksqlwildchars":<String_value>,
    "xmlsqlinjectionparsecomments":<String_value>,
    "xmlxssaction":<String[]_value>,
    "xmlwsiaction":<String[]_value>,
    "xmlattachmentaction":<String[]_value>,
    "xmlvalidationaction":<String[]_value>,
    "xmlerrorobject":<String_value>,
    "customsettings":<String_value>,
    "signatures":<String_value>,
    "xmlsoapfaultaction":<String[]_value>,
    "usehtmlerrorobject":<String_value>,
    "errorurl":<String_value>,
    "htmlerrorobject":<String_value>,
    "logeverypolicyhit":<String_value>,
    "stripcomments":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "striphtmlcomments":<String_value>,
    "stripxmlcomments":<String_value>,
    "exemptclosureurlsfromsecuritychecks":<String_value>,
    "defaultcharset":<String_value>,
    "postbodylimit":<Double_value>,
    "fileuploadmaxnum":<Double_value>,
    "canonicalizehtmlresponse":<String_value>,
    "enableformtagging":<String_value>,
    "sessionlessfieldconsistency":<String_value>,
    "sessionlessurlclosure":<String_value>,
    "semicolonfieldseparator":<String_value>,
    "excludefileuploadfromchecks":<String_value>,
    "sqlinjectionparsecomments":<String_value>,
    "invalidpercenthandling":<String_value>,
    "type":<String[]_value>,
    "checkrequestheaders":<String_value>,
    "optimizepartialreqs":<String_value>,
    "urldecoderequestcookies":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"appfwprofile":{
    "name":<String_value>,
    "starturlaction":<String[]_value>,
    "contenttypeaction":<String[]_value>,
    "inspectcontenttypes":<String[]_value>,
    "starturlclosure":<String_value>,
    "denyurlaction":<String[]_value>,
    "refererheadercheck":<String_value>,
    "cookieconsistencyaction":<String[]_value>,
    "cookietransforms":<String_value>,
    "cookieencryption":<String_value>,
    "cookieproxying":<String_value>,
    "addcookieflags":<String_value>,
    "fieldconsistencyaction":<String[]_value>,
    "csrftagaction":<String[]_value>,
    "crosssitescriptingaction":<String[]_value>,
    "crosssitescriptingtransformunsafehtml":<String_value>,
    "crosssitescriptingcheckcompleteurls":<String_value>,
    "sqlinjectionaction":<String[]_value>,
    "sqlinjectiontransformspecialchars":<String_value>,
    "sqlinjectiononlycheckfieldswithsqlchars":<String_value>,
    "sqlinjectiontype":<String_value>,
    "sqlinjectionchecksqlwildchars":<String_value>,
    "fieldformataction":<String[]_value>,
    "defaultfieldformattype":<String_value>,
    "defaultfieldformatminlength":<Double_value>,
    "defaultfieldformatmaxlength":<Double_value>,
    "bufferoverflowaction":<String[]_value>,
    "bufferoverflowmaxurllength":<Double_value>,
    "bufferoverflowmaxheaderlength":<Double_value>,
    "bufferoverflowmaxcookielength":<Double_value>,
    "creditcardaction":<String[]_value>,
    "creditcard":<String[]_value>,
    "creditcardmaxallowed":<Double_value>,
    "creditcardxout":<String_value>,
    "dosecurecreditcardlogging":<String_value>,
    "streaming":<String_value>,
    "trace":<String_value>,
    "requestcontenttype":<String_value>,
    "responsecontenttype":<String_value>,
    "xmldosaction":<String[]_value>,
    "xmlformataction":<String[]_value>,
    "xmlsqlinjectionaction":<String[]_value>,
    "xmlsqlinjectiononlycheckfieldswithsqlchars":<String_value>,
    "xmlsqlinjectiontype":<String_value>,
    "xmlsqlinjectionchecksqlwildchars":<String_value>,
    "xmlsqlinjectionparsecomments":<String_value>,
    "xmlxssaction":<String[]_value>,
    "xmlwsiaction":<String[]_value>,
    "xmlattachmentaction":<String[]_value>,
    "xmlvalidationaction":<String[]_value>,
    "xmlerrorobject":<String_value>,
    "customsettings":<String_value>,
    "signatures":<String_value>,
    "xmlsoapfaultaction":<String[]_value>,
    "usehtmlerrorobject":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "errorurl":<String_value>,
    "htmlerrorobject":<String_value>,
    "logeverypolicyhit":<String_value>,
    "stripcomments":<String_value>,
    "striphtmlcomments":<String_value>,
    "stripxmlcomments":<String_value>,
    "exemptclosureurlsfromsecuritychecks":<String_value>,
    "defaultcharset":<String_value>,
    "postbodylimit":<Double_value>,
    "fileuploadmaxnum":<Double_value>,
    "canonicalizehtmlresponse":<String_value>,
    "enableformtagging":<String_value>,
    "sessionlessfieldconsistency":<String_value>,
    "sessionlessurlclosure":<String_value>,
    "semicolonfieldseparator":<String_value>,
    "excludefileuploadfromchecks":<String_value>,
    "sqlinjectionparsecomments":<String_value>,
    "invalidpercenthandling":<String_value>,
    "type":<String[]_value>,
    "checkrequestheaders":<String_value>,
    "optimizepartialreqs":<String_value>,
    "urldecoderequestcookies":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwprofile":{
    "name":<String_value>,
    "starturlaction":true,
    "contenttypeaction":true,
    "inspectcontenttypes":true,
    "starturlclosure":true,
    "denyurlaction":true,
    "refererheadercheck":true,
    "cookieconsistencyaction":true,
    "cookietransforms":true,
    "cookieencryption":true,
    "cookieproxying":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

"addcookieflags":true,
"fieldconsistencyaction":true,
"csrftagaction":true,
"crosssitescriptingaction":true,
"crosssitescriptingtransformunsafehtml":true,
"crosssitescriptingcheckcompleteurls":true,
"sqlinjectionaction":true,
"sqlinjectiontransformspecialchars":true,
"sqlinjectiononlycheckfieldswithsqlchars":true,
"sqlinjectiontype":true,
"sqlinjectionchecksqlwildchars":true,
"fieldformataction":true,
"defaultfieldformattype":true,
"defaultfieldformatminlength":true,
"defaultfieldformatmaxlength":true,
"bufferoverflowaction":true,
"bufferoverflowmaxurllength":true,
"bufferoverflowmaxheaderlength":true,
"bufferoverflowmaxcookielength":true,
"creditcardaction":true,
"creditcard":true,
"creditcardmaxallowed":true,
"creditcardxout":true,
"dosecurecreditcardlogging":true,
"streaming":true,
"trace":true,
"requestcontenttype":true,
"responsecontenttype":true,
"xmldosaction":true,
"xmlformataction":true,
"xmlsqlinjectionaction":true,
"xmlsqlinjectiononlycheckfieldswithsqlchars":true,
"xmlsqlinjectiontype":true,
"xmlsqlinjectionchecksqlwildchars":true,
"xmlsqlinjectionparsecomments":true,
"xmlxssaction":true,
"xmlwsiaction":true,
"xmlattachmentaction":true,
"xmlvalidationaction":true,
"xmlerrorobject":true,
"customsettings":true,
"signatures":true,
"xmlsoapfaultaction":true,
"usehtmlerrorobject":true,
"errorurl":true,
"htmlerrorobject":true,
"logeverypolicyhit":true,
"stripcomments":true,
"striphtmlcomments":true,
"stripxmlcomments":true,
"exemptclosureurlsfromsecuritychecks":true,
"defaultcharset":true,
"postbodylimit":true,
"fileuploadmaxnum":true,
"canonicalizehtmlresponse":true,
"enableformtagging":true,
"sessionlessfieldconsistency":true,
"sessionlessurlclosure":true,
"semicolonfieldseparator":true,
"excludefileuploadfromchecks":true,
"sqlinjectionparsecomments":true,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "invalidpercenthandling":true,
        "type":true,
        "checkrequestheaders":true,
        "optimizepartialreqs":true,
        "urldecoderequestcookies":true,
        "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

restore

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile?**action=restore**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwprofile":{
        "archivename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile": [ {
    "name":<String_value>,
    "type":<String[]_value>,
    "defaults":<String_value>,
    "usehtmlerrorobject":<String_value>,
    "errorurl":<String_value>,
    "htmlerrorobject":<String_value>,
    "logeverypolicyhit":<String_value>,
    "stripcomments":<String_value>,
    "striphtmlcomments":<String_value>,
    "stripxmlcomments":<String_value>,
    "defaultcharset":<String_value>,
    "postbodylimit":<Double_value>,
    "fileuploadmaxnum":<Double_value>,
    "canonicalizehtmlresponse":<String_value>,
    "enableformtagging":<String_value>,
    "sessionlessfieldconsistency":<String_value>,
    "sessionlessurlclosure":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
"semicolonfieldseparator":<String_value>,
"excludefileuploadfromchecks":<String_value>,
"sqlinjectionparsecomments":<String_value>,
"checkrequestheaders":<String_value>,
"optimizepartialreqs":<String_value>,
"urldecoderequestcookies":<String_value>,
"comment":<String_value>,
"starturlaction":<String[]_value>,
"contenttypeaction":<String[]_value>,
"inspectcontenttypes":<String[]_value>,
"starturlclosure":<String_value>,
"denyurlaction":<String[]_value>,
"refererheadercheck":<String_value>,
"csrftagaction":<String[]_value>,
"csrftag":<String_value>,
"crosssitescriptingaction":<String[]_value>,
"crosssitescriptingtransformunsafehtml":<String_value>,
"crosssitescriptingcheckcompleteurls":<String_value>,
"exemptclosureurlsfromsecuritychecks":<String_value>,
"sqlinjectionaction":<String[]_value>,
"sqlinjectiontransformspecialchars":<String_value>,
"sqlinjectiononlycheckfieldswithsqlchars":<String_value>,
"sqlinjectiontype":<String_value>,
"sqlinjectionchecksqlwildchars":<String_value>,
"invalidpercenthandling":<String_value>,
"fieldconsistencyaction":<String[]_value>,
"cookieconsistencyaction":<String[]_value>,
"cookietransforms":<String_value>,
"cookieencryption":<String_value>,
"cookieproxying":<String_value>,
"addcookieflags":<String_value>,
"bufferoverflowaction":<String[]_value>,
"bufferoverflowmaxurllength":<Double_value>,
"bufferoverflowmaxheaderlength":<Double_value>,
"bufferoverflowmaxcookielength":<Double_value>,
"fieldformataction":<String[]_value>,
"defaultfieldformattype":<String_value>,
"defaultfieldformatminlength":<Double_value>,
"defaultfieldformatmaxlength":<Double_value>,
"creditcardaction":<String[]_value>,
"creditcard":<String[]_value>,
"creditcardmaxallowed":<Double_value>,
"creditcardxout":<String_value>,
"dosecurecreditcardlogging":<String_value>,
"streaming":<String_value>,
"trace":<String_value>,
"requestcontenttype":<String_value>,
"responsecontenttype":<String_value>,
"xmlerrorobject":<String_value>,
"signatures":<String_value>,
"xmlformataction":<String[]_value>,
"xmldosaction":<String[]_value>,
"xmlsqlinjectionaction":<String[]_value>,
"xmlsqlinjectiononlycheckfieldswithsqlchars":<String_value>,
"xmlsqlinjectiontype":<String_value>,
"xmlsqlinjectionchecksqlwildchars":<String_value>,
"xmlsqlinjectionparsecomments":<String_value>,
"xmlxssaction":<String[]_value>,
"xmlwsiaction":<String[]_value>,
"xmlattachmentaction":<String[]_value>,
"xmlvalidationaction":<String[]_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "xmlsoapfaultaction":<String[]_value>,
    "state":<String_value>,
    "builtin":<Boolean_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile": [ {
    "name":<String_value>,
    "type":<String[]_value>,
    "defaults":<String_value>,
    "usehtmlerrorobject":<String_value>,
    "errorurl":<String_value>,
    "htmlerrorobject":<String_value>,
    "logeverypolicyhit":<String_value>,
    "stripcomments":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

"striphtmlcomments":<String_value>,
"stripxmlcomments":<String_value>,
"defaultcharset":<String_value>,
"postbodylimit":<Double_value>,
"fileuploadmaxnum":<Double_value>,
"canonicalizehtmlresponse":<String_value>,
"enableformtagging":<String_value>,
"sessionlessfieldconsistency":<String_value>,
"sessionlessurlclosure":<String_value>,
"semicolonfieldseparator":<String_value>,
"excludefileuploadfromchecks":<String_value>,
"sqlinjectionparsecomments":<String_value>,
"checkrequestheaders":<String_value>,
"optimizepartialreqs":<String_value>,
"urldecoderequestcookies":<String_value>,
"comment":<String_value>,
"starturlaction":<String[]_value>,
"contenttypeaction":<String[]_value>,
"inspectcontenttypes":<String[]_value>,
"starturlclosure":<String_value>,
"denyurlaction":<String[]_value>,
"refererheadercheck":<String_value>,
"csrftagaction":<String[]_value>,
"csrftag":<String_value>,
"crosssitescriptingaction":<String[]_value>,
"crosssitescriptingtransformunsafehtml":<String_value>,
"crosssitescriptingcheckcompleteurls":<String_value>,
"exemptclosureurlsfromsecuritychecks":<String_value>,
"sqlinjectionaction":<String[]_value>,
"sqlinjectiontransformspecialchars":<String_value>,
"sqlinjectiononlycheckfieldswithsqlchars":<String_value>,
"sqlinjectiontype":<String_value>,
"sqlinjectionchecksqlwildchars":<String_value>,
"invalidpercenthandling":<String_value>,
"fieldconsistencyaction":<String[]_value>,
"cookieconsistencyaction":<String[]_value>,
"cookietransforms":<String_value>,
"cookieencryption":<String_value>,
"cookieproxying":<String_value>,
"addcookieflags":<String_value>,
"bufferoverflowaction":<String[]_value>,
"bufferoverflowmaxurllength":<Double_value>,
"bufferoverflowmaxheaderlength":<Double_value>,
"bufferoverflowmaxcookielength":<Double_value>,
"fieldformataction":<String[]_value>,
"defaultfieldformattype":<String_value>,
"defaultfieldformatminlength":<Double_value>,
"defaultfieldformatmaxlength":<Double_value>,
"creditcardaction":<String[]_value>,
"creditcard":<String[]_value>,
"creditcardmaxallowed":<Double_value>,
"creditcardxout":<String_value>,
"dosecurecreditcardlogging":<String_value>,
"streaming":<String_value>,
"trace":<String_value>,
"requestcontenttype":<String_value>,
"responsecontenttype":<String_value>,
"xmlerrorobject":<String_value>,
"signatures":<String_value>,
"xmlformataction":<String[]_value>,
"xmldosaction":<String[]_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "xmlsqlinjectionaction":<String[]_value>,
    "xmlsqlinjectiononlycheckfieldswithsqlchars":<String_value>,
    "xmlsqlinjectiontype":<String_value>,
    "xmlsqlinjectionchecksqlwildchars":<String_value>,
    "xmlsqlinjectionparsecomments":<String_value>,
    "xmlxssaction":<String[]_value>,
    "xmlwsiaction":<String[]_value>,
    "xmlattachmentaction":<String[]_value>,
    "xmlvalidationaction":<String[]_value>,
    "xmlsoapfaultaction":<String[]_value>,
    "state":<String_value>,
    "builtin":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_binding

Jul 07, 2017

Binding object which returns the resources bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the application firewall profile. Minimum length = 1
appfwprofile_excluderescontenttype_binding	<appfwprofile_excluderescontenttype_binding[]>	Read-only	excluderescontenttype that can be bound to appfwprofile.
appfwprofile_xmlattachmenturl_binding	<appfwprofile_xmlattachmenturl_binding[]>	Read-only	xmlattachmenturl that can be bound to appfwprofile.
appfwprofile_crosssitescripting_binding	<appfwprofile_crosssitescripting_binding[]>	Read-only	crosssitescripting that can be bound to appfwprofile.
appfwprofile_xmlwsiurl_binding	<appfwprofile_xmlwsiurl_binding[]>	Read-only	xmlwsiurl that can be bound to appfwprofile.
appfwprofile_xmldosurl_binding	<appfwprofile_xmldosurl_binding[]>	Read-only	xmldosurl that can be bound to appfwprofile.
appfwprofile_fieldconsistency_binding	<appfwprofile_fieldconsistency_binding[]>	Read-only	fieldconsistency that can be bound to appfwprofile.
appfwprofile_safeobject_binding	<appfwprofile_safeobject_binding[]>	Read-only	safeobject that can be bound to appfwprofile.
appfwprofile_sqlinjection_binding	<appfwprofile_sqlinjection_binding[]>	Read-only	sqlinjection that can be bound to appfwprofile.
appfwprofile_denyurl_binding	<appfwprofile_denyurl_binding[]>	Read-only	denyurl that can be bound to appfwprofile.
appfwprofile_trustedlearningclients_binding	<appfwprofile_trustedlearningclients_binding[]>	Read-only	trustedlearningclients that can be bound to appfwprofile.
appfwprofile_csrftag_binding	<appfwprofile_csrftag_binding[]>	Read-only	csrftag that can be bound to appfwprofile.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
appfwprofile_cookieconsistency_binding	<appfwprofile_cookieconsistency_binding[]>	Read-only	cookieconsistency that can be bound to appfwprofile.
appfwprofile_starturl_binding	<appfwprofile_starturl_binding[]>	Read-only	starturl that can be bound to appfwprofile.
appfwprofile_xmlsqlinjection_binding	<appfwprofile_xmlsqlinjection_binding[]>	Read-only	xmlsqlinjection that can be bound to appfwprofile.
appfwprofile_creditcardnumber_binding	<appfwprofile_creditcardnumber_binding[]>	Read-only	creditcardnumber that can be bound to appfwprofile.
appfwprofile_xmlxss_binding	<appfwprofile_xmlxss_binding[]>	Read-only	xmlxss that can be bound to appfwprofile.
appfwprofile_contenttype_binding	<appfwprofile_contenttype_binding[]>	Read-only	contenttype that can be bound to appfwprofile.
appfwprofile_fieldformat_binding	<appfwprofile_fieldformat_binding[]>	Read-only	fieldformat that can be bound to appfwprofile.
appfwprofile_xmlvalidationurl_binding	<appfwprofile_xmlvalidationurl_binding[]>	Read-only	xmlvalidationurl that can be bound to appfwprofile.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "appfwprofile_binding": [ {
    "name":<String_value>,
    "appfwprofile_csrftag_binding":<appfwprofile_csrftag_binding[]_value>,
    "appfwprofile_starturl_binding":<appfwprofile_starturl_binding[]_value>,
    "appfwprofile_xmlvalidationurl_binding":<appfwprofile_xmlvalidationurl_binding[]_value>,
    "appfwprofile_xmlattachmenturl_binding":<appfwprofile_xmlattachmenturl_binding[]_value>,
    "appfwprofile_safeobject_binding":<appfwprofile_safeobject_binding[]_value>,
    "appfwprofile_cookieconsistency_binding":<appfwprofile_cookieconsistency_binding[]_value>,
    "appfwprofile_fieldconsistency_binding":<appfwprofile_fieldconsistency_binding[]_value>,
    "appfwprofile_xmlwsiurl_binding":<appfwprofile_xmlwsiurl_binding[]_value>,
    "appfwprofile_xmlsqlinjection_binding":<appfwprofile_xmlsqlinjection_binding[]_value>,
    "appfwprofile_sqlinjection_binding":<appfwprofile_sqlinjection_binding[]_value>,
    "appfwprofile_trustedlearningclients_binding":<appfwprofile_trustedlearningclients_binding[]_value>,
    "appfwprofile_crosssitescripting_binding":<appfwprofile_crosssitescripting_binding[]_value>,
    "appfwprofile_excluderescontenttype_binding":<appfwprofile_excluderescontenttype_binding[]_value>,
    "appfwprofile_contenttype_binding":<appfwprofile_contenttype_binding[]_value>,
    "appfwprofile_denyurl_binding":<appfwprofile_denyurl_binding[]_value>,
    "appfwprofile_xmlxss_binding":<appfwprofile_xmlxss_binding[]_value>,
    "appfwprofile_fieldformat_binding":<appfwprofile_fieldformat_binding[]_value>,
    "appfwprofile_creditcardnumber_binding":<appfwprofile_creditcardnumber_binding[]_value>,
    "appfwprofile_xmldosurl_binding":<appfwprofile_xmldosurl_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwsettings

Jul 07, 2017

Configuration for AS settings resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
defaultprofile	<String>	Read-write	Profile to use when a connection does not match any policy. Default setting is APPFW_BYPASS, which sends unmatched connections back to the NetScaler appliance without attempting to filter them further. Default value: APPFW_BYPASS Minimum length = 1
undefaction	<String>	Read-write	Profile to use when an application firewall policy evaluates to undefined (UNDEF). An UNDEF event indicates an internal error condition. The APPFW_BLOCK built-in profile is the default setting. You can specify a different built-in or user-created profile as the UNDEF profile. Default value: APPFW_BLOCK Minimum length = 1
sessiontimeout	<Double>	Read-write	Timeout, in seconds, after which a user session is terminated. Before continuing to use the protected web site, the user must establish a new session by opening a designated start URL. Default value: 900 Minimum value = 1 Maximum value = 65535
learnratelimit	<Double>	Read-write	Maximum number of connections per second that the application firewall learning engine examines to generate new relaxations for learning-enabled security checks. The application firewall drops any connections above this limit from the list of connections used by the learning engine. Default value: 400 Minimum value = 1 Maximum value = 1000
sessionlifetime	<Double>	Read-write	Maximum amount of time (in seconds) that the application firewall allows a user session to remain active, regardless of user activity. After this time, the user session is terminated. Before continuing to use the protected web site, the user must establish a new session by opening a designated start URL. Default value: 0 Minimum value = 0 Maximum value = 2147483647
			Name of the session cookie that the application firewall uses to track user sessions.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			numbers, and the hyphen (-) and underscore (_) symbols.
sessioncookiename	<String>	Read-write	The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cookie name" or 'my cookie name'). Minimum length = 1
clientiploggingheader	<String>	Read-write	Name of an HTTP header that contains the IP address that the client used to connect to the protected web site or service.
importsizelimit	<Double>	Read-write	Cumulative total maximum number of bytes in web forms imported to a protected web site. If a user attempts to upload files with a total byte count higher than the specified limit, the application firewall blocks the request. Default value: 134217728 Minimum value = 1 Maximum value = 134217728
signatureautoupdate	<String>	Read-write	Flag used to enable/disable auto update signatures. Default value: OFF Possible values = ON, OFF
signatureurl	<String>	Read-write	URL to download the mapping file from server. Default value: https://s3.amazonaws.com/NSAppFwSignatures/SignaturesMapping.xml
cookiepostencryptprefix	<String>	Read-write	String that is prepended to all encrypted cookie values. Minimum length = 1
logmalformedreq	<String>	Read-write	Log requests that are so malformed that application firewall parsing doesn't occur. Default value: ON Possible values = ON, OFF
geolocationlogging	<String>	Read-write	Enable Geo-Location Logging in CEF format logs. Default value: OFF Possible values = ON, OFF
ceflogging	<String>	Read-write	Enable CEF format logs. Default value: OFF Possible values = ON, OFF
entitydecoding	<String>	Read-write	Transform multibyte (double- or half-width) characters to single width characters. Default value: OFF Possible values = ON, OFF
useconfigurablesecretkey	<String>	Read-write	Use configurable secret key in AppFw operations. Default value: OFF Possible values = ON, OFF

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwsettings

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwsettings":{
    "defaultprofile":<String_value>,
    "undefaction":<String_value>,
    "sessiontimeout":<Double_value>,
    "learnratelimit":<Double_value>,
    "sessionlifetime":<Double_value>,
    "sessioncookiename":<String_value>,
    "clientiploggingheader":<String_value>,
    "importsizelimit":<Double_value>,
    "signatureautoupdate":<String_value>,
    "signatureurl":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cookiepostencryptprefix":<String_value>,
    "logmalformedreq":<String_value>,
    "geolocationlogging":<String_value>,
    "ceflogging":<String_value>,
    "entitydecoding":<String_value>,
    "useconfigurablesecretkey":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwsettings?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwsettings":{
    "defaultprofile":true,
    "undefaction":true,
    "sessiontimeout":true,
    "learnratelimit":true,
    "sessionlifetime":true,
    "sessioncookiename":true,
    "clientiploggingheader":true,
    "importsizelimit":true,
    "signatureautoupdate":true,
    "signatureurl":true,
    "cookiepostencryptprefix":true,
    "logmalformedreq":true,
    "geolocationlogging":true,
    "ceflogging":true,
    "entitydecoding":true,
    "useconfigurablesecretkey":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwsettings

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwsettings": [ {
      "defaultprofile":<String_value>,
      "undefaction":<String_value>,
      "sessiontimeout":<Double_value>,
      "learnratelimit":<Double_value>,
      "sessionlifetime":<Double_value>,
      "sessioncookiename":<String_value>,
      "clientiploggingheader":<String_value>,
      "importsizelimit":<Double_value>,
      "signatureautoupdate":<String_value>,
      "signatureurl":<String_value>,
      "cookiepostencryptprefix":<String_value>,
      "logmalformedreq":<String_value>,
      "geolocationlogging":<String_value>,
      "ceflogging":<String_value>,
      "entitydecoding":<String_value>,
      "useconfigurablesecretkey":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_contenttype_binding

Jul 07, 2017

Binding object showing the contenttype that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
contenttype	<String>	Read-write	A regular expression that designates a content-type on the content-types list.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_contenttype_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_contenttype_binding":{
    "name":<String_value>,
    "comment":<String_value>,
    "state":<String_value>,
    "contenttype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_contenttype_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_contenttype_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_contenttype_binding": [ {
    "state":<String_value>,
    "name":<String_value>,
    "contenttype":<String_value>,
    "comment":<String_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_contenttype_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"appfwprofile_contenttype_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_cookieconsistency_binding

May 20, 2016

Binding object showing the cookieconsistency that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
isregex	<String>	Read-write	Is the cookie name a regular expression?. Possible values = REGEX, NOTREGEX
cookieconsistency	<String>	Read-write	The name of the cookie to be checked.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_cookieconsistency_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_cookieconsistency_binding":{
    "name":<String_value>,
    "cookieconsistency":<String_value>,
    "isregex":<String_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_cookieconsistency_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_cookieconsistency_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_cookieconsistency_binding": [ {
    "state":<String_value>,
    "name":<String_value>,
    "isregex":<String_value>,
    "cookieconsistency":<String_value>,
    "comment":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_cookieconsistency_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"appfwprofile_cookieconsistency_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_creditcardnumber_binding

May 20, 2016

Binding object showing the creditcardnumber that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
creditcardnumberurl	<String>	Read-write	The url for which the list of credit card numbers are needed to be bypassed from inspection.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
creditcardnumber	<String>	Read-write	The object expression that is to be excluded from safe commerce check.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_creditcardnumber_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_creditcardnumber_binding":{
    "name":<String_value>,
    "comment":<String_value>,
    "state":<String_value>,
    "creditcardnumber":<String_value>,
    "creditcardnumberurl":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_creditcardnumber_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_creditcardnumber_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_creditcardnumber_binding": [ {
    "creditcardnumberurl":<String_value>,
    "state":<String_value>,
    "name":<String_value>,
    "creditcardnumber":<String_value>,
    "comment":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_creditcardnumber_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"appfwprofile_creditcardnumber_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_crosssitescripting_binding

May 20, 2016

Binding object showing the crosssitescripting that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
crosssitescripting	<String>	Read-write	The web form field name.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
isregex_xss	<String>	Read-write	Is the web form field name a regular expression?. Possible values = REGEX, NOTREGEX
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
formactionurl_xss	<String>	Read-write	The web form action URL.
as_value_expr_xss	<String>	Read-write	The web form value expression.
as_scan_location_xss	<String>	Read-write	Location of cross-site scripting exception - form field, header or cookie. Possible values = FORMFIELD, HEADER, COOKIE
as_value_type_xss	<String>	Read-write	The web form value type. Possible values = Tag, Attribute, Pattern
isvalueregex_xss	<String>	Read-write	Is the web form field value a regular expression?. Possible values = REGEX, NOTREGEX
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_crosssitescripting_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_crosssitescripting_binding":{
    "name":<String_value>,
    "crosssitescripting":<String_value>,
    "formactionurl_xss":<String_value>,
    "isregex_xss":<String_value>,
    "as_scan_location_xss":<String_value>,
    "as_value_type_xss":<String_value>,
    "as_value_expr_xss":<String_value>,
    "isvalueregex_xss":<String_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_crosssitescripting_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_crosssitescripting_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_crosssitescripting_binding": [ {
    "crosssitescripting":<String_value>,
    "name":<String_value>,
    "isregex_xss":<String_value>,
    "state":<String_value>,
    "comment":<String_value>,
    "formactionurl_xss":<String_value>,
    "as_value_expr_xss":<String_value>,
    "as_scan_location_xss":<String_value>,
    "as_value_type_xss":<String_value>,
    "isvalueregex_xss":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_crosssitescripting_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_crosssitescripting_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_csrftag_binding

Jul 07, 2017

Binding object showing the csrftag that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
csrftag	<String>	Read-write	The web form originating URL.
csrfformactionurl	<String>	Read-write	The web form action URL.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_csrftag_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_csrftag_binding":{
    "name":<String_value>,
    "csrftag":<String_value>,
    "csrfformactionurl":<String_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_csrftag_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_csrftag_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_csrftag_binding": [ {
    "state":<String_value>,
    "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "csrftag":<String_value>,
    "csrfformactionurl":<String_value>,
    "comment":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_csrftag_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_csrftag_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_denyurl_binding

Jul 07, 2017

Binding object showing the denyurl that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
denyurl	<String>	Read-write	A regular expression that designates a URL on the Deny URL list.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_denyurl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_denyurl_binding":{
    "name":<String_value>,
    "denyurl":<String_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_denyurl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_denyurl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_denyurl_binding": [ {
      "state":<String_value>,
      "denyurl":<String_value>,
      "name":<String_value>,
      "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_denyurl_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"appfwprofile_denyurl_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_excluderescontenttype_binding

May 20, 2016

Binding object showing the excluderescontenttype that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
excluderescontenttype	<String>	Read-write	A regular expression that represents the content type of the response that are to be excluded from inspection.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_excluderescontenttype_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_excluderescontenttype_binding":{
    "name":<String_value>,
    "comment":<String_value>,
    "state":<String_value>,
    "excluderescontenttype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_excluderescontenttype_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_excluderescontenttype_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_excluderescontenttype_binding": [ {
      "excluderescontenttype":<String_value>,
      "name":<String_value>,
      "state":<String_value>,
      "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_excluderescontenttype_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"appfwprofile_excluderescontenttype_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_fieldconsistency_binding

May 20, 2016

Binding object showing the fieldconsistency that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
fieldconsistency	<String>	Read-write	The web form field name.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
isregex_ffc	<String>	Read-write	Is the web form field name a regular expression?. Possible values = REGEX, NOTREGEX
formactionurl_ffc	<String>	Read-write	The web form action URL.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_fieldconsistency_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_fieldconsistency_binding":{
    "name":<String_value>,
    "fieldconsistency":<String_value>,
    "formactionurl_ffc":<String_value>,
    "isregex_ffc":<String_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_fieldconsistency_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_fieldconsistency_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_fieldconsistency_binding": [ {
    "fieldconsistency":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "state":<String_value>,
    "name":<String_value>,
    "isregex_ffc":<String_value>,
    "formactionurl_ffc":<String_value>,
    "comment":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_fieldconsistency_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_fieldconsistency_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_fieldformat_binding

May 20, 2016

Binding object showing the fieldformat that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
fieldformatmaxlength	<Double>	Read-write	The maximum allowed length for data in this form field.
isregex_ff	<String>	Read-write	Is the form field name a regular expression?. Possible values = REGEX, NOTREGEX
fieldtype	<String>	Read-write	The field type you are assigning to this form field.
formactionurl_ff	<String>	Read-write	Action URL of the form field to which a field format will be assigned.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
fieldformatminlength	<Double>	Read-write	The minimum allowed length for data in this form field.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
fieldformat	<String>	Read-write	Name of the form field to which a field format will be assigned.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_fieldformat_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_fieldformat_binding":{
    "name":<String_value>,
    "fieldformat":<String_value>,
    "formactionurl_ff":<String_value>,
    "fieldtype":<String_value>,
    "fieldformatminlength":<Double_value>,
    "fieldformatmaxlength":<Double_value>,
    "isregex_ff":<String_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_fieldformat_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_fieldformat_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_fieldformat_binding": [ {
    "state":<String_value>,
    "fieldformatmaxlength":<Double_value>,
    "isregex_ff":<String_value>,
    "fieldtype":<String_value>,
    "formactionurl_ff":<String_value>,
    "name":<String_value>,
    "fieldformatminlength":<Double_value>,
    "comment":<String_value>,
    "fieldformat":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_fieldformat_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_fieldformat_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_safeobject_binding

May 20, 2016

Binding object showing the safeobject that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
maxmatchlength	<Double>	Read-write	Maximum match length for a Safe Object expression.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
expression	<String>	Read-write	A regular expression that defines the Safe Object.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
safeobject	<String>	Read-write	Name of the Safe Object.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
action	<String[]>	Read-write	Safe Object action types. (BLOCK | LOG | STATS | NONE). Possible values = none, block, log, remove, stats, xout
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_safeobject_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_safeobject_binding":{
    "name":<String_value>,
    "safeobject":<String_value>,
    "expression":<String_value>,
    "maxmatchlength":<Double_value>,
    "action":<String[]_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_safeobject_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_safeobject_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "appfwprofile_safeobject_binding": [ {
    "maxmatchlength":<Double_value>,
    "state":<String_value>,
    "expression":<String_value>,
    "name":<String_value>,
    "safeobject":<String_value>,
    "comment":<String_value>,
    "action":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_safeobject_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_safeobject_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_sqlinjection_binding

May 20, 2016

Binding object showing the sqlinjection that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
as_value_expr_sql	<String>	Read-write	The web form value expression.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
formactionurl_sql	<String>	Read-write	The web form action URL.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
isregex_sql	<String>	Read-write	Is the web form field name a regular expression?. Possible values = REGEX, NOTREGEX
isvalueregex_sql	<String>	Read-write	Is the web form field value a regular expression?. Possible values = REGEX, NOTREGEX
as_scan_location_sql	<String>	Read-write	Location of SQL injection exception - form field, header or cookie. Possible values = FORMFIELD, HEADER, COOKIE
sqlinjection	<String>	Read-write	The web form field name.
as_value_type_sql	<String>	Read-write	The web form value type. Possible values = Keyword, SpecialString, Wildchar
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_sqlinjection_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_sqlinjection_binding":{
    "name":<String_value>,
    "sqlinjection":<String_value>,
    "formactionurl_sql":<String_value>,
    "isregex_sql":<String_value>,
    "as_scan_location_sql":<String_value>,
    "as_value_type_sql":<String_value>,
    "as_value_expr_sql":<String_value>,
    "isvalueregex_sql":<String_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_sqlinjection_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_sqlinjection_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_sqlinjection_binding": [ {
    "as_value_expr_sql":<String_value>,
    "state":<String_value>,
    "formactionurl_sql":<String_value>,
    "name":<String_value>,
    "isregex_sql":<String_value>,
    "isvalueregex_sql":<String_value>,
    "as_scan_location_sql":<String_value>,
    "sqlinjection":<String_value>,
    "as_value_type_sql":<String_value>,
    "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_sqlinjection_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_sqlinjection_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_starturl_binding

May 20, 2016

Binding object showing the starturl that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
starturl	<String>	Read-write	A regular expression that designates a URL on the Start URL list.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_starturl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_starturl_binding":{
    "name":<String_value>,
    "starturl":<String_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_starturl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_starturl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_starturl_binding": [ {
      "state":<String_value>,
      "name":<String_value>,
      "starturl":<String_value>,
      "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_starturl_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"appfwprofile_starturl_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_trustedlearningclients_binding

Jul 07, 2017

Binding object showing the trustedlearningclients that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
trustedlearningclients	<String>	Read-write	Specify trusted host/network IP.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_trustedlearningclients_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_trustedlearningclients_binding":{
    "name":<String_value>,
    "trustedlearningclients":<String_value>,
    "comment":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_trustedlearningclients_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_trustedlearningclients_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_trustedlearningclients_binding": [ {
    "state":<String_value>,
    "trustedlearningclients":<String_value>,
    "name":<String_value>,
    "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_trustedlearningclients_binding/name_value<String>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"appfwprofile_trustedlearningclients_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_xmlattachmenturl_binding

May 20, 2016

Binding object showing the xmlattachmenturl that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
xmlattachmenturl	<String>	Read-write	XML attachment URL regular expression length.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
xmlmaxattachmentsize	<Double>	Read-write	Specify maximum attachment size.
xmlmaxattachmentsizecheck	<String>	Read-write	State if XML Max attachment size Check is ON or OFF. Protects against XML requests with large attachment data. Possible values = ON, OFF
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
xmlattachmentcontenttypecheck	<String>	Read-write	State if XML attachment content-type check is ON or OFF. Protects against XML requests with illegal attachments. Possible values = ON, OFF
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
xmlattachmentcontenttype	<String>	Read-write	Specify content-type regular expression.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_xmlattachmenturl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_xmlattachmenturl_binding":{
    "name":<String_value>,
    "comment":<String_value>,
    "state":<String_value>,
    "xmlattachmenturl":<String_value>,
    "xmlmaxattachmentsizecheck":<String_value>,
    "xmlmaxattachmentsize":<Double_value>,
    "xmlattachmentcontenttypecheck":<String_value>,
    "xmlattachmentcontenttype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmlattachmenturl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_xmlattachmenturl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_xmlattachmenturl_binding": [ {
      "xmlattachmenturl":<String_value>,
      "name":<String_value>,
      "xmlmaxattachmentsize":<Double_value>,
      "xmlmaxattachmentsizecheck":<String_value>,
      "state":<String_value>,
      "xmlattachmentcontenttypecheck":<String_value>,
      "comment":<String_value>,
      "xmlattachmentcontenttype":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlattachmenturl_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_xmlattachmenturl_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_xmldosurl_binding

May 20, 2016

Binding object showing the xmldosurl that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
xmlmaxelementdepthcheck	<String>	Read-write	State if XML Max element depth check is ON or OFF. Possible values = ON, OFF
xmlmaxfilesize	<Double>	Read-write	Specify the maximum size of XML messages. Protects against overflow attacks.
xmlmaxnamespaceurilength	<Double>	Read-write	Specify the longest URI of any XML namespace. Protects against overflow attacks.
xmldosurl	<String>	Read-write	XML DoS URL regular expression length.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
xmlsoaparraycheck	<String>	Read-write	State if XML SOAP Array check is ON or OFF. Possible values = ON, OFF
xmlmaxelementnamelengthcheck	<String>	Read-write	State if XML Max element name length check is ON or OFF. Possible values = ON, OFF
xmlmaxelementscheck	<String>	Read-write	State if XML Max elements check is ON or OFF. Possible values = ON, OFF
xmlmaxentityexpansions	<Double>	Read-write	Specify maximum allowed number of entity expansions. Protects aganist Entity Expansion Attack.
xmlmaxattributes	<Double>	Read-write	Specify maximum number of attributes per XML element. Protects against overflow attacks.
xmlmaxfilesizecheck	<String>	Read-write	State if XML Max file size check is ON or OFF. Possible values = ON, OFF
xmlmaxchardatalength	<Double>	Read-write	Specify the maximum size of CDATA. Protects against overflow attacks and large quantities of unparsed data within XML messages.
xmlmaxnamespacescheck	<String>	Read-write	State if XML Max namespaces check is ON or OFF. Possible values = ON, OFF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

xmlmaxnamespaces	<Double>	Read-write	Specify maximum number of active namespaces. Protects against overflow attacks.
Name	**Data Type**	**Permissions**	**Description**
xmlmaxattributenamelengthcheck	<String>	Read-write	State if XML Max attribute name length check is ON or OFF. Possible values = ON, OFF
xmlblockdtd	<String>	Read-write	State if XML DTD is ON or OFF. Protects against recursive Document Type Declaration (DTD) entity expansion attacks. Also, SOAP messages cannot have DTDs in messages. . Possible values = ON, OFF
xmlmaxattributevaluelength	<Double>	Read-write	Specify the longest value of any XML attribute. Protects against overflow attacks.
xmlmaxelementdepth	<Double>	Read-write	Maximum nesting (depth) of XML elements. This check protects against documents that have excessive hierarchy depths.
xmlmaxelementnamelength	<Double>	Read-write	Specify the longest name of any element (including the expanded namespace) to protect against overflow attacks.
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
xmlblockpi	<String>	Read-write	State if XML Block PI is ON or OFF. Protects resources from denial of service attacks as SOAP messages cannot have processing instructions (PI) in messages. Possible values = ON, OFF
xmlmaxelementchildrencheck	<String>	Read-write	State if XML Max element children check is ON or OFF. Possible values = ON, OFF
xmlmaxelements	<Double>	Read-write	Specify the maximum number of XML elements allowed. Protects against overflow attacks.
xmlmaxentityexpansionscheck	<String>	Read-write	State if XML Max Entity Expansions Check is ON or OFF. Possible values = ON, OFF
xmlmaxnamespaceurilengthcheck	<String>	Read-write	State if XML Max namespace URI length check is ON or OFF. Possible values = ON, OFF
xmlmaxentityexpansiondepthcheck	<String>	Read-write	State if XML Max Entity Expansions Depth Check is ON or OFF. Possible values = ON, OFF
xmlmaxattributevaluelengthcheck	<String>	Read-write	State if XML Max atribute value length is ON or OFF. Possible values = ON, OFF
xmlmaxsoaparraysize	<Double>	Read-write	XML Max Total SOAP Array Size. Protects against SOAP Array Abuse attack.
xmlmaxentityexpansiondepth	<Double>	Read-write	Specify maximum entity expansion depth. Protects aganist Entity Expansion Attack.
xmlmaxnodescheck	<String>	Read-write	State if XML Max nodes check is ON or OFF. Possible values = ON, OFF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
xmlmaxattributenamelength	<Double>	Read-write	Specify the longest name of any XML attribute. Protects against overflow attacks.
xmlmaxchardatalengthcheck	<String>	Read-write	State if XML Max CDATA length check is ON or OFF. Possible values = ON, OFF
xmlminfilesizecheck	<String>	Read-write	State if XML Min file size check is ON or OFF. Possible values = ON, OFF
xmlmaxelementchildren	<Double>	Read-write	Specify the maximum number of children allowed per XML element. Protects against overflow attacks.
xmlminfilesize	<Double>	Read-write	Enforces minimum message size.
xmlmaxnodes	<Double>	Read-write	Specify the maximum number of XML nodes. Protects against overflow attacks.
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
xmlmaxattributescheck	<String>	Read-write	State if XML Max attributes check is ON or OFF. Possible values = ON, OFF
xmlmaxsoaparrayrank	<Double>	Read-write	XML Max Individual SOAP Array Rank. This is the dimension of the SOAP array.
xmlblockexternalentities	<String>	Read-write	State if XML Block External Entities Check is ON or OFF. Protects against XML External Entity (XXE) attacks that force applications to parse untrusted external entities (sources) in XML documents. Possible values = ON, OFF
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_xmldosurl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_xmldosurl_binding":{
    "name":<String_value>,
    "comment":<String_value>,
    "state":<String_value>,
    "xmldosurl":<String_value>,
    "xmlmaxelementdepthcheck":<String_value>,
    "xmlmaxelementdepth":<Double_value>,
    "xmlmaxelementnamelengthcheck":<String_value>,
    "xmlmaxelementnamelength":<Double_value>,
    "xmlmaxelementscheck":<String_value>,
    "xmlmaxelements":<Double_value>,
    "xmlmaxelementchildrencheck":<String_value>,
    "xmlmaxelementchildren":<Double_value>,
    "xmlmaxattributescheck":<String_value>,
    "xmlmaxattributes":<Double_value>,
    "xmlmaxattributenamelengthcheck":<String_value>,
    "xmlmaxattributenamelength":<Double_value>,
    "xmlmaxattributevaluelengthcheck":<String_value>,
    "xmlmaxattributevaluelength":<Double_value>,
    "xmlmaxchardatalengthcheck":<String_value>,
    "xmlmaxchardatalength":<Double_value>,
    "xmlmaxfilesizecheck":<String_value>,
    "xmlmaxfilesize":<Double_value>,
    "xmlminfilesizecheck":<String_value>,
    "xmlminfilesize":<Double_value>,
    "xmlblockpi":<String_value>,
    "xmlblockdtd":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "xmlblockexternalentities":<String_value>,
    "xmlmaxentityexpansionscheck":<String_value>,
    "xmlmaxentityexpansions":<Double_value>,
    "xmlmaxentityexpansiondepthcheck":<String_value>,
    "xmlmaxentityexpansiondepth":<Double_value>,
    "xmlmaxnamespacescheck":<String_value>,
    "xmlmaxnamespaces":<Double_value>,
    "xmlmaxnamespaceurilengthcheck":<String_value>,
    "xmlmaxnamespaceurilength":<Double_value>,
    "xmlsoaparraycheck":<String_value>,
    "xmlmaxsoaparraysize":<Double_value>,
    "xmlmaxsoaparrayrank":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmldosurl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_xmldosurl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_xmldosurl_binding": [ {
    "xmlmaxelementdepthcheck":<String_value>,
    "xmlmaxfilesize":<Double_value>,
    "xmlmaxnamespaceurilength":<Double_value>,
    "xmldosurl":<String_value>,
    "state":<String_value>,
    "xmlsoaparraycheck":<String_value>,
    "xmlmaxelementnamelengthcheck":<String_value>,
    "xmlmaxelementscheck":<String_value>,
    "xmlmaxentityexpansions":<Double_value>,
    "xmlmaxattributes":<Double_value>,
    "xmlmaxfilesizecheck":<String_value>,
    "xmlmaxchardatalength":<Double_value>,
    "xmlmaxnamespacescheck":<String_value>,
    "xmlmaxnamespaces":<Double_value>,
    "xmlmaxattributenamelengthcheck":<String_value>,
    "xmlblockdtd":<String_value>,
    "xmlmaxattributevaluelength":<Double_value>,
    "xmlmaxelementdepth":<Double_value>,
    "xmlmaxelementnamelength":<Double_value>,
    "name":<String_value>,
    "xmlblockpi":<String_value>,
    "xmlmaxelementchildrencheck":<String_value>,
    "xmlmaxelements":<Double_value>,
    "xmlmaxentityexpansionscheck":<String_value>,
    "xmlmaxnamespaceurilengthcheck":<String_value>,
    "xmlmaxentityexpansiondepthcheck":<String_value>,
    "xmlmaxattributevaluelengthcheck":<String_value>,
    "xmlmaxsoaparraysize":<Double_value>,
    "xmlmaxentityexpansiondepth":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "xmlmaxnodescheck":<String_value>,
    "xmlmaxattributenamelength":<Double_value>,
    "xmlmaxchardatalengthcheck":<String_value>,
    "xmlminfilesizecheck":<String_value>,
    "xmlmaxelementchildren":<Double_value>,
    "xmlminfilesize":<Double_value>,
    "xmlmaxnodes":<Double_value>,
    "comment":<String_value>,
    "xmlmaxattributescheck":<String_value>,
    "xmlmaxsoaparrayrank":<Double_value>,
    "xmlblockexternalentities":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmldosurl_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_xmldosurl_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_xmlsqlinjection_binding

May 20, 2016

Binding object showing the xmlsqlinjection that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
as_scan_location_xmlsql	<String>	Read-write	Location of SQL injection exception - XML Element or Attribute. Possible values = ELEMENT, ATTRIBUTE
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
xmlsqlinjection	<String>	Read-write	Exempt the specified URL from the XML SQL injection check. An XML SQL injection exemption (relaxation) consists of the following items: * URL. URL to exempt, as a string or a PCRE-format regular expression. * ISREGEX flag. REGEX if URL is a regular expression, NOTREGEX if URL is a fixed string. * Location. ELEMENT if the injection is located in an XML element, ATTRIBUTE if located in an XML attribute.
isregex_xmlsql	<String>	Read-write	Is the XML SQL Injection exempted field name a regular expression?. Possible values = REGEX, NOTREGEX
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_xmlsqlinjection_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_xmlsqlinjection_binding":{
    "name":<String_value>,
    "comment":<String_value>,
    "state":<String_value>,
    "xmlsqlinjection":<String_value>,
    "isregex_xmlsql":<String_value>,
    "as_scan_location_xmlsql":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmlsqlinjection_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_xmlsqlinjection_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "appfwprofile_xmlsqlinjection_binding": [ {
    "as_scan_location_xmlsql":<String_value>,
    "state":<String_value>,
    "name":<String_value>,
    "xmlsqlinjection":<String_value>,
    "isregex_xmlsql":<String_value>,
    "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlsqlinjection_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_xmlsqlinjection_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_xmlvalidationurl_binding

May 20, 2016

Binding object showing the xmlvalidationurl that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
xmlwsdl	<String>	Read-write	WSDL object for soap request validation.
xmlendpointcheck	<String>	Read-write	Modifies the behaviour of the Request URL validation w.r.t. the Service URL. If set to ABSOLUTE, the entire request URL is validated with the entire URL mentioned in Service of the associated WSDL. eg: Service URL: http://example.org/ExampleService, Request URL: http//example.com/ExampleService would FAIL the validation. If set to RELAIVE, only the non-hostname part of the request URL is validated against the non-hostname part of the Service URL. eg: Service URL: http://example.org/ExampleService, Request URL: http//example.com/ExampleService would PASS the validation. Possible values = ABSOLUTE, RELATIVE
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
xmlvalidateresponse	<String>	Read-write	Validate response message. Possible values = ON, OFF
xmlvalidationurl	<String>	Read-write	XML Validation URL regular expression.
xmlresponseschema	<String>	Read-write	XML Schema object for response validation.
xmlvalidatesoapenvelope	<String>	Read-write	Validate SOAP Evelope only. Possible values = ON, OFF
xmlrequestschema	<String>	Read-write	XML Schema object for request validation .
xmladditionalsoapheaders	<String>	Read-write	Allow addtional soap headers. Possible values = ON, OFF
			Any comments about the purpose of profile, or other useful information

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
Name	<Double>	Read-Write	Description

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_xmlvalidationurl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_xmlvalidationurl_binding":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "comment":<String_value>,
    "state":<String_value>,
    "xmlvalidationurl":<String_value>,
    "xmlrequestschema":<String_value>,
    "xmlresponseschema":<String_value>,
    "xmlwsdl":<String_value>,
    "xmladditionalsoapheaders":<String_value>,
    "xmlendpointcheck":<String_value>,
    "xmlvalidatesoapenvelope":<String_value>,
    "xmlvalidateresponse":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmlvalidationurl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_xmlvalidationurl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_xmlvalidationurl_binding": [ {
    "state":<String_value>,
    "xmlwsdl":<String_value>,
    "xmlendpointcheck":<String_value>,
    "name":<String_value>,
    "xmlvalidateresponse":<String_value>,
    "xmlvalidationurl":<String_value>,
    "xmlresponseschema":<String_value>,
    "xmlvalidatesoapenvelope":<String_value>,
    "xmlrequestschema":<String_value>,
    "xmladditionalsoapheaders":<String_value>,
    "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlvalidationurl_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"appfwprofile_xmlvalidationurl_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_xmlwsiurl_binding

May 20, 2016

Binding object showing the xmlwsiurl that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
xmlwsichecks	<String>	Read-write	Specify a comma separated list of relevant WS-I rule IDs. (R1140, R1141).
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
xmlwsiurl	<String>	Read-write	XML WS-I URL regular expression length.
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_xmlwsiurl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_xmlwsiurl_binding":{
    "name":<String_value>,
    "comment":<String_value>,
    "state":<String_value>,
    "xmlwsiurl":<String_value>,
    "xmlwsichecks":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmlwsiurl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_xmlwsiurl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile_xmlwsiurl_binding": [ {
    "xmlwsichecks":<String_value>,
    "name":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "xmlwsiurl":<String_value>,
    "state":<String_value>,
    "comment":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlwsiurl_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_xmlwsiurl_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile_xmlxss_binding

May 20, 2016

Binding object showing the xmlxss that can be bound to appfwprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-write	Enabled. Possible values = ENABLED, DISABLED
name	<String>	Read-write	Name of the profile to which to bind an exemption or rule. Minimum length = 1
as_scan_location_xmlxss	<String>	Read-write	Location of XSS injection exception - XML Element or Attribute. Possible values = ELEMENT, ATTRIBUTE
isregex_xmlxss	<String>	Read-write	Is the XML XSS exempted field name a regular expression?. Possible values = REGEX, NOTREGEX
comment	<String>	Read-write	Any comments about the purpose of profile, or other useful information about the profile.
xmlxss	<String>	Read-write	Exempt the specified URL from the XML cross-site scripting (XSS) check. An XML cross-site scripting exemption (relaxation) consists of the following items: * URL. URL to exempt, as a string or a PCRE-format regular expression. * ISREGEX flag. REGEX if URL is a regular expression, NOTREGEX if URL is a fixed string. * Location. ELEMENT if the attachment is located in an XML element, ATTRIBUTE if located in an XML attribute.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/appfwprofile_xmlxss_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"appfwprofile_xmlxss_binding":{
    "name":<String_value>,
    "comment":<String_value>,
    "state":<String_value>,
    "xmlxss":<String_value>,
    "isregex_xmlxss":<String_value>,
    "as_scan_location_xmlxss":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appfwprofile_xmlxss_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwprofile_xmlxss_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "appfwprofile_xmlxss_binding": [ {
    "state":<String_value>,
    "name":<String_value>,
    "as_scan_location_xmlxss":<String_value>,
    "isregex_xmlxss":<String_value>,
    "comment":<String_value>,
    "xmlxss":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwprofile_xmlxss_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appfwprofile_xmlxss_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwsignatures

Jul 07, 2017

Configuration for application firewall signatures XML configuration resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the signature object. Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL (protocol, host, path, and file name) for the location at which to store the imported signatures object. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
xslt	<String>	Read-write	XSLT file source. Maximum length = 2047
comment	<String>	Read-write	Any comments to preserve information about the signatures object. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrite any existing signatures object of the same name.
merge	<Boolean>	Read-write	Merges the existing Signature with new signature rules.
sha1	<String>	Read-write	File path for sha1 file to validate signature file. Minimum length = 1 Maximum length = 2047
mergedefault	<Boolean>	Read-write	Merges signature file with default signature file.
response	<String>	Read-only	.

Operations (click to see Properties)

DELETE | IMPORT | CHANGE | GET (ALL) | GET

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwsignatures/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwsignatures?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"appfwsignatures":{
    "src":<String_value>,
    "name":<String_value>,
    "xslt":<String_value>,
    "comment":<String_value>,
    "overwrite":<Boolean_value>,
    "merge":<Boolean_value>,
    "sha1":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

change

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwsignatures?**action=update**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwsignatures":{
    "name":<String_value>,
    "mergedefault":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwsignatures

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwsignatures": [ {
      "name":<String_value>,
      "response":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwsignatures/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwsignatures/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwsignatures/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwsignatures": [ {
      "name":<String_value>,
      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwtransactionrecords

May 20, 2016

Configuration for Application firewall transaction record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
httptransactionid	<Double>	Read-only	The http transaction identifier.
packetengineid	<Integer>	Read-only	The packet engine identifier.
appfwsessionid	<String>	Read-only	The session identifier set by the Application Firewall to track the user session.
profilename	<String>	Read-only	Application Firewall profile name.
url	<String>	Read-only	Request URL.
clientip	<String>	Read-only	The IP address of client.
destip	<String>	Read-only	The IP address of destination.
starttime	<String>	Read-only	Conveys time at which request processing started.
endtime	<String>	Read-only	Conveys time at which request processing end.
requestcontentlength	<Integer>	Read-only	The content length of request.
requestyields	<Double>	Read-only	The number of times yielded during request processing to send heart beat packets.
requestmaxprocessingtime	<Double>	Read-only	The maximum processing time across yields during request processing.
responsecontentlength	<Integer>	Read-only	The content length of response.
responseyields	<Double>	Read-only	The number of times yielded during response processing to send heart

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
responsemaxprocessingtime	<Double>	Read-only	The maximum processing time across yields during response processing.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwtransactionrecords

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwtransactionrecords?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwtransactionrecords?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwtransactionrecords resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwtransactionrecords?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwtransactionrecords?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwtransactionrecords resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwtransactionrecords": [ {
    "httptransactionid":<Double_value>,
    "packetengineid":<Integer_value>,
    "appfwsessionid":<String_value>,
    "profilename":<String_value>,
    "url":<String_value>,
    "clientip":<String_value>,
    "destip":<String_value>,
    "starttime":<String_value>,
    "endtime":<String_value>,
    "requestcontentlength":<Integer_value>,
    "requestyields":<Double_value>,
    "requestmaxprocessingtime":<Double_value>,
    "responsecontentlength":<Integer_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "responseyields":<Double_value>,
    "responsemaxprocessingtime":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwtransactionrecords?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwtransactionrecords": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwxmlcontenttype

May 20, 2016

Configuration for XML Content type resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
xmlcontenttypevalue	<String>	Read-write	Content type to be classified as XML. Minimum length = 1
isregex	<String>	Read-write	Is field name a regular expression?. Default value: NOTREGEX Possible values = REGEX, NOTREGEX
builtin	<String[]>	Read-only	Flag to determine if xmlcontenttype is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwxmlcontenttype":{
    "xmlcontenttypevalue":<String_value>,
    "isregex":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype/xmlcontenttypevalue_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appfwxmlcontenttype resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appfwxmlcontenttype resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwxmlcontenttype": [ {
      "xmlcontenttypevalue":<String_value>,
      "isregex":<String_value>,
      "builtin":<String[]_value>
}]}
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype/xmlcontenttypevalue_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype/xmlcontenttypevalue_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype/xmlcontenttypevalue_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwxmlcontenttype": [ {
      "xmlcontenttypevalue":<String_value>,
      "isregex":<String_value>,
      "builtin":<String[]_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlcontenttype?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "appfwxmlcontenttype": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwxmlerrorpage

May 20, 2016

Configuration for xml error page resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Indicates name of the imported xml error page to be removed. Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL (protocol, host, path, and name) for the location at which to store the imported XML error object. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
comment	<String>	Read-write	Any comments to preserve information about the XML error object. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrite any existing XML error object of the same name.
response	<String>	Read-only	.

Operations (click to see Properties)

DELETE | IMPORT | CHANGE | GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlerrorpage/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlerrorpage?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwxmlerrorpage":{
    "src":<String_value>,
    "name":<String_value>,
    "comment":<String_value>,
    "overwrite":<Boolean_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

change

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlerrorpage?**action=update**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwxmlerrorpage":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlerrorpage

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "appfwxmlerrorpage": [ {
    "name":<String_value>,
    "response":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlerrorpage/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwxmlerrorpage/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appfwxmlerrorpage/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwxmlerrorpage": [ {
    "name":<String_value>,
    "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwxmlschema

May 20, 2016

Configuration for XML schema resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the XML Schema object to remove. Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL (protocol, host, path, and file name) for the location at which to store the imported XML Schema. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
comment	<String>	Read-write	Any comments to preserve information about the XML Schema object. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrite any existing XML Schema object of the same name.
response	<String>	Read-only	.

Operations (click to see Properties)

DELETE | IMPORT | GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlschema/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlschema?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appfwxmlschema":{
    "src":<String_value>,
    "name":<String_value>,
    "comment":<String_value>,
    "overwrite":<Boolean_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlschema

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwxmlschema": [ {
      "name":<String_value>,
      "response":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appfwxmlschema/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appfwxmlschema/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/appfwxmlschema/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwxmlschema": [ {
      "name":<String_value>,
      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Appqoe

May 20, 2016

Application Level Quality of Experience configuration

Name	Description
appqoeaction	Configuration for AppQoS action
appqoecustomresp	Configuration for AppQoE custom response page
appqoeparameter	Configuration for QOS parameter
appqoepolicy	Configuration for AppQoS policy
appqoepolicy_binding	Binding object showing the resources that can be bound to appqoepolicy
appqoepolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to appqoepolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appqoeaction

May 20, 2016

Configuration for AppQoS action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the AppQoE action. Must begin with a letter, number, or the underscore symbol (_). Other characters allowed, after the first character, are the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), and colon (:) characters. This is a mandatory argument.
priority	<String>	Read-write	Priority for queuing the request. If server resources are not available for a request that matches the configured rule, this option specifies a priority for queuing the request until the server resources are available again. If priority is not configured then Lowest priority will be used to queue the request. Possible values = HIGH, MEDIUM, LOW, LOWEST
respondwith	<String>	Read-write	Responder action to be taken when the threshold is reached. Available settings function as follows: ACS - Serve content from an alternative content service Threshold : maxConn or delay NS - Serve from the NetScaler appliance (built-in response) Threshold : maxConn or delay. Possible values = ACS, NS
customfile	<String>	Read-write	name of the HTML page object to use as the response. Minimum length = 1
altcontentsvcname	<String>	Read-write	Name of the alternative content service to be used in the ACS. Minimum length = 1 Maximum length = 127
altcontentpath	<String>	Read-write	Path to the alternative content service to be used in the ACS. Minimum length = 4 Maximum length = 127
polqdepth	<Double>	Read-write	Policy queue depth threshold value. When the policy queue size (number of requests queued for the policy binding this action is attached to) increases to the specified polqDepth value, subsequent requests are dropped to the lowest priority level. Minimum value = 0 Maximum value = 4294967294
			Queue depth threshold value per priorirty level. If the queue size (number of requests in the queue of that particular priorirty) on the virtual server to which this

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			dropped to the lowest priority level. Minimum value = 0 Maximum value = 4294967294
maxconn	<Double>	Read-write	Maximum number of concurrent connections that can be open for requests that matches with rule. Minimum value = 1 Maximum value = 4294967294
delay	<Double>	Read-write	Delay threshold, in microseconds, for requests that match the policy's rule. If the delay statistics gathered for the matching request exceed the specified delay, configured action triggered for that request, if there is no action then requests are dropped to the lowest priority level. Minimum value = 1 Maximum value = 599999999
dostrigexpression	<String>	Read-write	Optional expression to add second level check to trigger DoS actions. Specifically used for Analytics based DoS response generation.
dosaction	<String>	Read-write	DoS Action to take when vserver will be considered under DoS attack and corresponding rule matches. Mandatory if AppQoE actions are to be used for DoS attack prevention. Possible values = SimpleResponse, HICResponse
hits	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoeaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appqoeaction":{
    "name":<String_value>,
    "priority":<String_value>,
    "respondwith":<String_value>,
    "customfile":<String_value>,
    "altcontentsvcname":<String_value>,
    "altcontentpath":<String_value>,
    "polqdepth":<Double_value>,
    "priqdepth":<Double_value>,
    "maxconn":<Double_value>,
    "delay":<Double_value>,
    "dostrigexpression":<String_value>,
    "dosaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoeaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoeaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appqoeaction":{
    "name":<String_value>,
    "priority":<String_value>,
    "altcontentsvcname":<String_value>,
    "altcontentpath":<String_value>,
    "polqdepth":<Double_value>,
    "priqdepth":<Double_value>,
    "maxconn":<Double_value>,
    "delay":<Double_value>,
    "dostrigexpression":<String_value>,
    "dosaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoeaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"appqoeaction":{
    "name":<String_value>,
    "priority":true,
    "altcontentsvcname":true,
    "altcontentpath":true,
    "polqdepth":true,
    "priqdepth":true,
    "maxconn":true,
    "delay":true,
    "dosaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoeaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appqoeaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appqoeaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appqoeaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appqoeaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appqoeaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appqoeaction resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoeaction": [ {
      "name":<String_value>,
      "hits":<Double_value>,
      "priority":<String_value>,
      "respondwith":<String_value>,
      "polqdepth":<Double_value>,
      "priqdepth":<Double_value>,
      "altcontentsvcname":<String_value>,
      "altcontentpath":<String_value>,
      "maxconn":<Double_value>,
      "delay":<Double_value>,
      "customfile":<String_value>,
      "dostrigexpression":<String_value>,
      "dosaction":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoeaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appqoeaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appqoeaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoeaction": [ {
    "name":<String_value>,
    "hits":<Double_value>,
    "priority":<String_value>,
    "respondwith":<String_value>,
    "polqdepth":<Double_value>,
    "priqdepth":<Double_value>,
    "altcontentsvcname":<String_value>,
    "altcontentpath":<String_value>,
    "maxconn":<Double_value>,
    "delay":<Double_value>,
    "customfile":<String_value>,
    "dostrigexpression":<String_value>,
    "dosaction":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoeaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "appqoeaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appqoecustomresp

May 20, 2016

Configuration for AppQoE custom response page resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
src	<String>	Read-write	. Minimum length = 1 Maximum length = 2047
name	<String>	Read-write	Indicates name of the custom response HTML page to import/update. Minimum length = 1 Maximum length = 31
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

IMPORT | DELETE | CHANGE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoecustomresp?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appqoecustomresp":{
    "src":<String_value>,
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoecustomresp/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

change

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoecustomresp?**action=update**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appqoecustomresp":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoecustomresp

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appqoecustomresp?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appqoecustomresp?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appqoecustomresp resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appqoecustomresp?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/appqoecustomresp?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appqoecustomresp resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoecustomresp": [ {
      "name":<String_value>,
      "src":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoecustomresp?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoecustomresp": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appqoeparameter

Jul 07, 2017

Configuration for QOS parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sessionlife	<Double>	Read-write	Time, in seconds, between the first time and the next time the AppQoE alternative content window is displayed. The alternative content window is displayed only once during a session for the same browser accessing a configured URL, so this parameter determines the length of a session. Default value: 300 Minimum value = 1 Maximum value = 4294967294
avgwaitingclient	<Double>	Read-write	average number of client connections, that can sit in service waiting queue. Default value: 1000000 Minimum value = 0 Maximum value = 4294967294
maxaltrespbandwidth	<Double>	Read-write	maximum bandwidth which will determine whether to send alternate content response. Default value: 100 Minimum value = 1 Maximum value = 4294967294
dosattackthresh	<Double>	Read-write	average number of client connection that can queue up on vserver level without triggering DoS mitigation module. Default value: 2000 Minimum value = 0 Maximum value = 4294967294

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoeparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appqoeparameter":{
    "sessionlife":<Double_value>,
    "avgwaitingclient":<Double_value>,
    "maxaltrespbandwidth":<Double_value>,
    "dosattackthresh":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoeparameter?**action=unset**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appqoeparameter":{
    "sessionlife":true,
    "avgwaitingclient":true,
    "maxaltrespbandwidth":true,
    "dosattackthresh":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoeparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoeparameter": [ {
    "sessionlife":<Double_value>,
    "avgwaitingclient":<Double_value>,
    "maxaltrespbandwidth":<Double_value>,
    "dosattackthresh":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appqoepolicy

Jul 07, 2017

Configuration for AppQoS policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	. Minimum length = 1
rule	<String>	Read-write	Expression or name of a named expression, against which the request is evaluated. The policy is applied if the rule evaluates to true.
action	<String>	Read-write	Configured AppQoE action to trigger. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appqoepolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appqoepolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of appqoepolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appqoepolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoepolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "hits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoepolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "hits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoepolicy": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appqoepolicy_binding

May 20, 2016

Binding object which returns the resources bound to appqoepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	. Minimum length = 1
appqoepolicy_lbvserver_binding	<appqoepolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to appqoepolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "appqoepolicy_binding": [ {
    "name":<String_value>,
    "appqoepolicy_lbvserver_binding":<appqoepolicy_lbvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appqoepolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to appqoepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	. Minimum length = 1
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
bindpriority	<Double>	Read-only	Specifies the binding of the policy. use only in display.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of appqoepolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the appqoepolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "appqoepolicy_lbvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "bindpriority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/appqoepolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"appqoepolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Audit

May 20, 2016

Audit configuration.

Name	Description
auditmessageaction	Configuration for message action
auditmessages	Configuration for audit message
auditnslogaction	Configuration for ns log action
auditnslogparams	Configuration for ns log parameters
auditnslogpolicy	Configuration for ns log policy
auditnslogpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to auditnslogpolicy
auditnslogpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to auditnslogpolicy
auditnslogpolicy_appfwglobal_binding	Binding object showing the appfwglobal that can be bound to auditnslogpolicy
auditnslogpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to auditnslogpolicy
auditnslogpolicy_binding	Binding object showing the resources that can be bound to auditnslogpolicy
auditnslogpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to auditnslogpolicy
auditnslogpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to auditnslogpolicy
auditnslogpolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to auditnslogpolicy
auditnslogpolicy_tmglobal_binding	Binding object showing the tmglobal that can be bound to auditnslogpolicy
auditnslogpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to auditnslogpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_vpnvserver_binding	
Name	**Description**
auditsyslogaction	Configuration for system log action
auditsyslogparams	Configuration for system log parameters
auditsyslogpolicy	Configuration for system log policy
auditsyslogpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to auditsyslogpolicy
auditsyslogpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to auditsyslogpolicy
auditsyslogpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to auditsyslogpolicy
auditsyslogpolicy_binding	Binding object showing the resources that can be bound to auditsyslogpolicy
auditsyslogpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to auditsyslogpolicy
auditsyslogpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to auditsyslogpolicy
auditsyslogpolicy_rnatglobal_binding	Binding object showing the rnatglobal that can be bound to auditsyslogpolicy
auditsyslogpolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to auditsyslogpolicy
auditsyslogpolicy_tmglobal_binding	Binding object showing the tmglobal that can be bound to auditsyslogpolicy
auditsyslogpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to auditsyslogpolicy
auditsyslogpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to auditsyslogpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditmessageaction

May 20, 2016

Configuration for message action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the audit message action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the message action is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my message action? or ?my message action?). Minimum length = 1
loglevel	<String>	Read-write	Audit log level, which specifies the severity level of the log message being generated.. The following loglevels are valid: * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. Possible values = EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG
stringbuilderexpr	<String>	Read-write	Default-syntax expression that defines the format and content of the log message.
logtonewnslog	<String>	Read-write	Send the message to the new nslog. Possible values = YES, NO
bypasssafetycheck	<String>	Read-write	Bypass the safety check and allow unsafe expressions. Default value: NO Possible values = YES, NO
loglevel1	<String>	Read-only	. Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG, NONE
hits	<Double>	Read-only	The number of times the action has been taken.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
referencecount	<Double>	Read-only	The number of references to the action.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"auditmessageaction":{
    "name":<String_value>,
    "loglevel":<String_value>,
    "stringbuilderexpr":<String_value>,
    "logtonewnslog":<String_value>,
    "bypasssafetycheck":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditmessageaction":{
    "name":<String_value>,
    "loglevel":<String_value>,
    "stringbuilderexpr":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "logtonewnslog":<String_value>,
    "bypasssafetycheck":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditmessageaction":{
    "name":<String_value>,
    "logtonewnslog":true,
    "bypasssafetycheck":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction?**filter=property-name1:property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

val1,property-name2:property-val2

Use this query-parameter to get the filtered set of auditmessageaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditmessageaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditmessageaction": [ {
    "name":<String_value>,
    "loglevel":<String_value>,
    "loglevel1":<String_value>,
    "stringbuilderexpr":<String_value>,
    "logtonewnslog":<String_value>,
    "bypasssafetycheck":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "referencecount":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditmessageaction": [ {
      "name":<String_value>,
      "loglevel":<String_value>,
      "loglevel1":<String_value>,
      "stringbuilderexpr":<String_value>,
      "logtonewnslog":<String_value>,
      "bypasssafetycheck":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "referencecount":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditmessageaction?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "auditmessageaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditmessages

Jul 07, 2017

Configuration for audit message resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
loglevel	<String[]>	Read-write	Audit log level filter, which specifies the types of events to display. The following loglevels are valid: * ALL - All events. * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG
numofmesgs	<Double>	Read-write	Number of log messages to be displayed. Default value: 20 Minimum value = 1 Maximum value = 256
value	<String>	Read-only	The Audit message.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/auditmessages

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/auditmessages?**args=loglevel:<String[]_value>,numofmesgs:<Double_value>**

Use this query-parameter to get auditmessages resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditmessages?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/auditmessages?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of auditmessages resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/auditmessages?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/auditmessages?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditmessages resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditmessages": [ {
loglevel:<String[]_value>,numofmesgs:<Double_value>      "value":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditmessages?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "auditmessages": [{ "__count": "#no"}] }

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogaction

Jul 07, 2017

Configuration for ns log action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the nslog action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the nslog action is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my nslog action? or ?my nslog action). Minimum length = 1
serverip	<String>	Read-write	IP address of the nslog server. Minimum length = 1
serverdomainname	<String>	Read-write	Auditserver name as a FQDN. Mutually exclusive with serverIP. Minimum length = 1 Maximum length = 255
domainresolveretry	<Integer>	Read-write	Time, in seconds, for which the NetScaler appliance waits before sending another DNS query to resolve the host name of the audit server if the last query failed. Default value: 5 Minimum value = 5 Maximum value = 20939
serverport	<Integer>	Read-write	Port on which the nslog server accepts connections. Minimum value = 1
loglevel	<String[]>	Read-write	Audit log level, which specifies the types of events to log. Available settings function as follows: * ALL - All events. * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. * NONE - No events. Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
dateformat	<String>	Read-write	Format of dates in the logs. Supported formats are: * MMDDYYYY - U.S. style month/date/year format. * DDMMYYYY - European style date/month/year format. * YYYYMMDD - ISO style year/month/date format. Possible values = MMDDYYYY, DDMMYYYY, YYYYMMDD
logfacility	<String>	Read-write	Facility value, as defined in RFC 3164, assigned to the log message. Log facility values are numbers 0 to 7 (LOCAL0 through LOCAL7). Each number indicates where a specific message originated from, such as the NetScaler appliance itself, the VPN, or external. Possible values = LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7
tcp	<String>	Read-write	Log TCP messages. Possible values = NONE, ALL
acl	<String>	Read-write	Log access control list (ACL) messages. Possible values = ENABLED, DISABLED
timezone	<String>	Read-write	Time zone used for date and timestamps in the logs. Available settings function as follows: * GMT_TIME. Coordinated Universal Time. * LOCAL_TIME. The server?s timezone setting. Possible values = GMT_TIME, LOCAL_TIME
userdefinedauditlog	<String>	Read-write	Log user-configurable log messages to nslog. Setting this parameter to NO causes auditing to ignore all user-configured message actions. Setting this parameter to YES causes auditing to log user-configured message actions that meet the other logging criteria. Possible values = YES, NO
appflowexport	<String>	Read-write	Export log messages to AppFlow collectors. Appflow collectors are entities to which log messages can be sent so that some action can be performed on them. Possible values = ENABLED, DISABLED
lsn	<String>	Read-write	Log the LSN messages. Possible values = ENABLED, DISABLED
alg	<String>	Read-write	Log the ALG messages. Possible values = ENABLED, DISABLED
subscriberlog	<String>	Read-write	Log subscriber session event information. Possible values = ENABLED, DISABLED
domainresolvenow	<Boolean>	Read-write	Immediately send a DNS query to resolve the server's domain name.
ip	<String>	Read-only	The resolved IP address of the auditserver.
			Indicates that a variable is a built-in (SYSTEM INTERNAL) type.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
__count	Type<boolean>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditnslogaction":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "serverip":<String_value>,
    "serverdomainname":<String_value>,
    "domainresolveretry":<Integer_value>,
    "serverport":<Integer_value>,
    "loglevel":<String[]_value>,
    "dateformat":<String_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"auditnslogaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "serverdomainname":<String_value>,
    "domainresolveretry":<Integer_value>,
    "domainresolvenow":<Boolean_value>,
    "serverport":<Integer_value>,
    "loglevel":<String[]_value>,
    "dateformat":<String_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditnslogaction":{
    "name":<String_value>,
    "serverport":true,
    "loglevel":true,
    "dateformat":true,
    "logfacility":true,
    "tcp":true,
    "acl":true,
    "timezone":true,
    "userdefinedauditlog":true,
    "appflowexport":true,
    "lsn":true,
    "alg":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "subscriberlog":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of auditnslogaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "serverdomainname":<String_value>,
    "domainresolveretry":<Integer_value>,
    "ip":<String_value>,
    "serverport":<Integer_value>,
    "loglevel":<String[]_value>,
    "dateformat":<String_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "builtin":<String[]_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "serverdomainname":<String_value>,
    "domainresolveretry":<Integer_value>,
    "ip":<String_value>,
    "serverport":<Integer_value>,
    "loglevel":<String[]_value>,
    "dateformat":<String_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "builtin":<String[]_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "auditnslogaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogparams

May 20, 2016

Configuration for ns log parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
serverip	<String>	Read-write	IP address of the nslog server. Minimum length = 1
serverport	<Integer>	Read-write	Port on which the nslog server accepts connections. Minimum value = 1
dateformat	<String>	Read-write	Format of dates in the logs. Supported formats are: * MMDDYYYY - U.S. style month/date/year format. * DDMMYYYY - European style date/month/year format. * YYYYMMDD - ISO style year/month/date format. Possible values = MMDDYYYY, DDMMYYYY, YYYYMMDD
loglevel	<String[]>	Read-write	Types of information to be logged. Available settings function as follows: * ALL - All events. * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. * NONE - No events. Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG, NONE
logfacility	<String>	Read-write	Facility value, as defined in RFC 3164, assigned to the log message. Log facility values are numbers 0 to 7 (LOCAL0 through LOCAL7). Each number indicates where a specific message originated from, such as the NetScaler appliance itself, the VPN, or external. Possible values = LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7
tcp	<String>	Read-write	Configure auditing to log TCP messages. Possible values = NONE, ALL
			Configure auditing to log access control list (ACL) messages.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
timezone	<String>	Read-write	Time zone used for date and timestamps in the logs. Supported settings are: * GMT_TIME - Coordinated Universal Time. * LOCAL_TIME - Use the server?s timezone setting. Possible values = GMT_TIME, LOCAL_TIME
userdefinedauditlog	<String>	Read-write	Log user-configurable log messages to nslog. Setting this parameter to NO causes auditing to ignore all user-configured message actions. Setting this parameter to YES causes auditing to log user-configured message actions that meet the other logging criteria. Possible values = YES, NO
appflowexport	<String>	Read-write	Export log messages to AppFlow collectors. Appflow collectors are entities to which log messages can be sent so that some action can be performed on them. Possible values = ENABLED, DISABLED
lsn	<String>	Read-write	Log the LSN messages. Possible values = ENABLED, DISABLED
alg	<String>	Read-write	Log the ALG messages. Possible values = ENABLED, DISABLED
subscriberlog	<String>	Read-write	Log subscriber session event information. Possible values = ENABLED, DISABLED
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogparams

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditnslogparams":{
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "dateformat":<String_value>,
    "loglevel":<String[]_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogparams?**action=unset**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditnslogparams":{
    "serverip":true,
    "serverport":true,
    "loglevel":true,
    "dateformat":true,
    "logfacility":true,
    "tcp":true,
    "acl":true,
    "timezone":true,
    "userdefinedauditlog":true,
    "appflowexport":true,
    "lsn":true,
    "alg":true,
    "subscriberlog":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogparams

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogparams": [ {
    "name":<String_value>,
    "serverip":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "serverport":<Integer_value>,
    "dateformat":<String_value>,
    "loglevel":<String[]_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "builtin":<String[]_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy

May 20, 2016

Configuration for ns log policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the policy. Must begin with a letter, number, or the underscore character (_), and must consist only of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the nslog policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my nslog policy? or ?my nslog policy?). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that defines the messages to be logged to the nslog server. Minimum length = 1
action	<String>	Read-write	Nslog server action that is performed when this policy matches. NOTE: An nslog server action must be associated with an nslog audit policy. Minimum length = 1
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditnslogpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditnslogpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of auditnslogpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy/name_value<String>?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "auditnslogpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_aaagroup_binding

May 20, 2016

Binding object showing the aaagroup that can be bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_aaagroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_aaagroup_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy_aaagroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy_aaagroup_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"auditnslogpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_aaauser_binding

May 20, 2016

Binding object showing the aaauser that can be bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_aaauser_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_aaauser_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy_aaauser_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy_aaauser_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditnslogpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_appfwglobal_binding

May 20, 2016

Binding object showing the appfwglobal that can be bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_appfwglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_appfwglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_appfwglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_appfwglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy_appfwglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy_appfwglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_appfwglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditnslogpolicy_appfwglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy_authenticationvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditnslogpolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_binding

May 20, 2016

Binding object which returns the resources bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
auditnslogpolicy_tmglobal_binding	<auditnslogpolicy_tmglobal_binding[]>	Read-only	tmglobal that can be bound to auditnslogpolicy.
auditnslogpolicy_appfwglobal_binding	<auditnslogpolicy_appfwglobal_binding[]>	Read-only	appfwglobal that can be bound to auditnslogpolicy.
auditnslogpolicy_lbvserver_binding	<auditnslogpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to auditnslogpolicy.
auditnslogpolicy_vpnglobal_binding	<auditnslogpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to auditnslogpolicy.
auditnslogpolicy_vpnvserver_binding	<auditnslogpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to auditnslogpolicy.
auditnslogpolicy_csvserver_binding	<auditnslogpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to auditnslogpolicy.
auditnslogpolicy_aaauser_binding	<auditnslogpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to auditnslogpolicy.
auditnslogpolicy_systemglobal_binding	<auditnslogpolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to auditnslogpolicy.
auditnslogpolicy_authenticationvserver_binding	<auditnslogpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to auditnslogpolicy.
auditnslogpolicy_aaagroup_binding	<auditnslogpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to auditnslogpolicy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "auditnslogpolicy_binding": [ {
    "name":<String_value>,
    "auditnslogpolicy_systemglobal_binding":<auditnslogpolicy_systemglobal_binding[]_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "auditnslogpolicy_aaagroup_binding":<auditnslogpolicy_aaagroup_binding[]_value>,
    "auditnslogpolicy_lbvserver_binding":<auditnslogpolicy_lbvserver_binding[]_value>,
    "auditnslogpolicy_csvserver_binding":<auditnslogpolicy_csvserver_binding[]_value>,
    "auditnslogpolicy_vpnglobal_binding":<auditnslogpolicy_vpnglobal_binding[]_value>,
    "auditnslogpolicy_aaauser_binding":<auditnslogpolicy_aaauser_binding[]_value>,
    "auditnslogpolicy_authenticationvserver_binding":<auditnslogpolicy_authenticationvserver_binding[]_value>,
    "auditnslogpolicy_vpnvserver_binding":<auditnslogpolicy_vpnvserver_binding[]_value>,
    "auditnslogpolicy_appfwglobal_binding":<auditnslogpolicy_appfwglobal_binding[]_value>,
    "auditnslogpolicy_tmglobal_binding":<auditnslogpolicy_tmglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy_csvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditnslogpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_lbvserver_binding

Jul 07, 2017

Binding object showing the lbvserver that can be bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy_lbvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditnslogpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_systemglobal_binding

May 20, 2016

Binding object showing the systemglobal that can be bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_systemglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_systemglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy_systemglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy_systemglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditnslogpolicy_systemglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_tmglobal_binding

May 20, 2016

Binding object showing the tmglobal that can be bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_tmglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_tmglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_tmglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_tmglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy_tmglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy_tmglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_tmglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditnslogpolicy_tmglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_vpnglobal_binding

May 20, 2016

Binding object showing the vpnglobal that can be bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy_vpnglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"auditnslogpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditnslogpolicy_vpnvserver_binding

Jul 07, 2017

Binding object showing the vpnvserver that can be bound to auditnslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditnslogpolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditnslogpolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditnslogpolicy_vpnvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditnslogpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditnslogpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogaction

May 20, 2016

Configuration for system log action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the syslog action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the syslog action is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my syslog action? or ?my syslog action). Minimum length = 1
serverip	<String>	Read-write	IP address of the syslog server. Minimum length = 1
serverdomainname	<String>	Read-write	SYSLOG server name as a FQDN. Mutually exclusive with serverIP/lbVserverName. Minimum length = 1 Maximum length = 255
domainresolveretry	<Integer>	Read-write	Time, in seconds, for which the NetScaler appliance waits before sending another DNS query to resolve the host name of the syslog server if the last query failed. Default value: 5 Minimum value = 5 Maximum value = 20939
lbvservername	<String>	Read-write	Name of the LB vserver. Mutually exclusive with syslog serverIP/serverName. Minimum length = 1 Maximum length = 127
serverport	<Integer>	Read-write	Port on which the syslog server accepts connections. Minimum value = 1
			Audit log level, which specifies the types of events to log. Available values function as follows: * ALL - All events. * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			* NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. * NONE - No events. Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG, NONE
dateformat	<String>	Read-write	Format of dates in the logs. Supported formats are: * MMDDYYYY. -U.S. style month/date/year format. * DDMMYYYY - European style date/month/year format. * YYYYMMDD - ISO style year/month/date format. Possible values = MMDDYYYY, DDMMYYYY, YYYYMMDD
logfacility	<String>	Read-write	Facility value, as defined in RFC 3164, assigned to the log message. Log facility values are numbers 0 to 7 (LOCAL0 through LOCAL7). Each number indicates where a specific message originated from, such as the NetScaler appliance itself, the VPN, or external. Possible values = LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7
tcp	<String>	Read-write	Log TCP messages. Possible values = NONE, ALL
acl	<String>	Read-write	Log access control list (ACL) messages. Possible values = ENABLED, DISABLED
timezone	<String>	Read-write	Time zone used for date and timestamps in the logs. Supported settings are: * GMT_TIME. Coordinated Universal time. * LOCAL_TIME. Use the server?s timezone setting. Possible values = GMT_TIME, LOCAL_TIME
userdefinedauditlog	<String>	Read-write	Log user-configurable log messages to syslog. Setting this parameter to NO causes auditing to ignore all user-configured message actions. Setting this parameter to YES causes auditing to log user-configured message actions that meet the other logging criteria. Possible values = YES, NO
appflowexport	<String>	Read-write	Export log messages to AppFlow collectors. Appflow collectors are entities to which log messages can be sent so that some action can be performed on them. Possible values = ENABLED, DISABLED
lsn	<String>	Read-write	Log lsn info. Possible values = ENABLED, DISABLED
alg	<String>	Read-write	Log alg info. Possible values = ENABLED, DISABLED
subscriberlog	<String>	Read-write	Log subscriber session event information. Possible values = ENABLED, DISABLED
			Transport type used to send auditlogs to syslog server. Default type is UDP.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
tcpprofilename	<String>	Read-write	Name of the TCP profile whose settings are to be applied to the audit server info to tune the TCP connection parameters. Minimum length = 1 Maximum length = 127
maxlogdatasizetohold	<Double>	Read-write	Max size of log data that can be held in NSB chain of server info. Default value: 500 Minimum value = 50 Maximum value = 25600
dns	<String>	Read-write	Log DNS related syslog messages. Possible values = ENABLED, DISABLED
netprofile	<String>	Read-write	Name of the network profile. The SNIP configured in the network profile will be used as source IP while sending log messages. Minimum length = 1 Maximum length = 127
domainresolvenow	<Boolean>	Read-write	Immediately send a DNS query to resolve the server's domain name.
ip	<String>	Read-only	The resolved IP address of the syslog server.
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditsyslogaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "serverdomainname":<String_value>,
    "domainresolveretry":<Integer_value>,
    "lbvservername":<String_value>,
    "serverport":<Integer_value>,
    "loglevel":<String[]_value>,
    "dateformat":<String_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>,
    "transport":<String_value>,
    "tcpprofilename":<String_value>,
    "maxlogdatasizetohold":<Double_value>,
    "dns":<String_value>,
    "netprofile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

update

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditsyslogaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "serverdomainname":<String_value>,
    "lbvservername":<String_value>,
    "domainresolveretry":<Integer_value>,
    "domainresolvenow":<Boolean_value>,
    "serverport":<Integer_value>,
    "loglevel":<String[]_value>,
    "dateformat":<String_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>,
    "tcpprofilename":<String_value>,
    "maxlogdatasizetohold":<Double_value>,
    "dns":<String_value>,
    "netprofile":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditsyslogaction":{
    "name":<String_value>,
    "serverport":true,
    "loglevel":true,
    "dateformat":true,
    "logfacility":true,
    "tcp":true,
    "acl":true,
    "timezone":true,
    "userdefinedauditlog":true,
    "appflowexport":true,
    "lsn":true,
    "alg":true,
    "subscriberlog":true,
    "tcpprofilename":true,
    "maxlogdatasizetohold":true,
    "dns":true,
    "netprofile":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of auditsyslogaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "serverdomainname":<String_value>,
    "ip":<String_value>,
    "lbvservername":<String_value>,
    "domainresolveretry":<Integer_value>,
    "serverport":<Integer_value>,
    "loglevel":<String[]_value>,
    "dateformat":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "builtin":<String[]_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>,
    "transport":<String_value>,
    "tcpprofilename":<String_value>,
    "maxlogdatasizetohold":<Double_value>,
    "dns":<String_value>,
    "netprofile":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "auditsyslogaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "serverdomainname":<String_value>,
    "ip":<String_value>,
    "lbvservername":<String_value>,
    "domainresolveretry":<Integer_value>,
    "serverport":<Integer_value>,
    "loglevel":<String[]_value>,
    "dateformat":<String_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "builtin":<String[]_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>,
    "transport":<String_value>,
    "tcpprofilename":<String_value>,
    "maxlogdatasizetohold":<Double_value>,
    "dns":<String_value>,
    "netprofile":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogparams

May 20, 2016

Configuration for system log parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
serverip	<String>	Read-write	IP address of the syslog server. Minimum length = 1
serverport	<Integer>	Read-write	Port on which the syslog server accepts connections. Minimum value = 1
dateformat	<String>	Read-write	Format of dates in the logs. Supported formats are: * MMDDYYYY - U.S. style month/date/year format. * DDMMYYYY. European style -date/month/year format. * YYYYMMDD - ISO style year/month/date format. Possible values = MMDDYYYY, DDMMYYYY, YYYYMMDD
loglevel	<String[]>	Read-write	Types of information to be logged. Available settings function as follows: * ALL - All events. * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. * NONE - No events. Possible values = ALL, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG, NONE
logfacility	<String>	Read-write	Facility value, as defined in RFC 3164, assigned to the log message. Log facility values are numbers 0 to 7 (LOCAL0 through LOCAL7). Each number indicates where a specific message originated from, such as the NetScaler appliance itself, the VPN, or external. Possible values = LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7
tcp	<String>	Read-write	Log TCP messages. Possible values = NONE, ALL
			Log access control list (ACL) messages.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
timezone	<String>	Read-write	Time zone used for date and timestamps in the logs. Available settings function as follows: * GMT_TIME - Coordinated Universal Time. * LOCAL_TIME Use the server?s timezone setting. Possible values = GMT_TIME, LOCAL_TIME
userdefinedauditlog	<String>	Read-write	Log user-configurable log messages to syslog. Setting this parameter to NO causes audit to ignore all user-configured message actions. Setting this parameter to YES causes audit to log user-configured message actions that meet the other logging criteria. Possible values = YES, NO
appflowexport	<String>	Read-write	Export log messages to AppFlow collectors. Appflow collectors are entities to which log messages can be sent so that some action can be performed on them. Possible values = ENABLED, DISABLED
lsn	<String>	Read-write	Log the LSN messages. Possible values = ENABLED, DISABLED
alg	<String>	Read-write	Log the ALG messages. Possible values = ENABLED, DISABLED
subscriberlog	<String>	Read-write	Log subscriber session event information. Possible values = ENABLED, DISABLED
dns	<String>	Read-write	Log DNS related syslog messages. Possible values = ENABLED, DISABLED
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogparams

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditsyslogparams":{
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "dateformat":<String_value>,
    "loglevel":<String[]_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>,
    "dns":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogparams?**action=unset**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditsyslogparams":{
    "serverip":true,
    "serverport":true,
    "loglevel":true,
    "dateformat":true,
    "logfacility":true,
    "tcp":true,
    "acl":true,
    "timezone":true,
    "userdefinedauditlog":true,
    "appflowexport":true,
    "lsn":true,
    "alg":true,
    "subscriberlog":true,
    "dns":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogparams

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "auditsyslogparams": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "dateformat":<String_value>,
    "loglevel":<String[]_value>,
    "logfacility":<String_value>,
    "tcp":<String_value>,
    "acl":<String_value>,
    "timezone":<String_value>,
    "userdefinedauditlog":<String_value>,
    "appflowexport":<String_value>,
    "builtin":<String[]_value>,
    "lsn":<String_value>,
    "alg":<String_value>,
    "subscriberlog":<String_value>,
    "dns":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy

May 20, 2016

Configuration for system log policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the policy. Must begin with a letter, number, or the underscore character (_), and must consist only of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the syslog policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my syslog policy? or ?my syslog policy). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that defines the messages to be logged to the syslog server. Minimum length = 1
action	<String>	Read-write	Syslog server action to perform when this policy matches traffic. NOTE: A syslog server action must be associated with a syslog audit policy. Minimum length = 1
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditsyslogpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"auditsyslogpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of auditsyslogpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy/name_value<String>?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "auditsyslogpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_aaagroup_binding

May 20, 2016

Binding object showing the aaagroup that can be bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_aaagroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_aaagroup_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_aaagroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy_aaagroup_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditsyslogpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_aaauser_binding

May 20, 2016

Binding object showing the aaauser that can be bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_aaauser_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_aaauser_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_aaauser_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy_aaauser_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"auditsyslogpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy_authenticationvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"auditsyslogpolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
auditsyslogpolicy_csvserver_binding	<auditsyslogpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to auditsyslogpolicy.
auditsyslogpolicy_tmglobal_binding	<auditsyslogpolicy_tmglobal_binding[]>	Read-only	tmglobal that can be bound to auditsyslogpolicy.
auditsyslogpolicy_lbvserver_binding	<auditsyslogpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to auditsyslogpolicy.
auditsyslogpolicy_aaauser_binding	<auditsyslogpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to auditsyslogpolicy.
auditsyslogpolicy_vpnglobal_binding	<auditsyslogpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to auditsyslogpolicy.
auditsyslogpolicy_rnatglobal_binding	<auditsyslogpolicy_rnatglobal_binding[]>	Read-only	rnatglobal that can be bound to auditsyslogpolicy.
auditsyslogpolicy_vpnvserver_binding	<auditsyslogpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to auditsyslogpolicy.
auditsyslogpolicy_systemglobal_binding	<auditsyslogpolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to auditsyslogpolicy.
auditsyslogpolicy_authenticationvserver_binding	<auditsyslogpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to auditsyslogpolicy.
auditsyslogpolicy_aaagroup_binding	<auditsyslogpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to auditsyslogpolicy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "auditsyslogpolicy_binding": [ {
    "name":<String_value>,
    "auditsyslogpolicy_aaagroup_binding":<auditsyslogpolicy_aaagroup_binding[]_value>,
    "auditsyslogpolicy_tmglobal_binding":<auditsyslogpolicy_tmglobal_binding[]_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "auditsyslogpolicy_rnatglobal_binding":<auditsyslogpolicy_rnatglobal_binding[]_value>,
    "auditsyslogpolicy_vpnvserver_binding":<auditsyslogpolicy_vpnvserver_binding[]_value>,
    "auditsyslogpolicy_aaauser_binding":<auditsyslogpolicy_aaauser_binding[]_value>,
    "auditsyslogpolicy_vpnglobal_binding":<auditsyslogpolicy_vpnglobal_binding[]_value>,
    "auditsyslogpolicy_authenticationvserver_binding":<auditsyslogpolicy_authenticationvserver_binding[]_value>,
    "auditsyslogpolicy_lbvserver_binding":<auditsyslogpolicy_lbvserver_binding[]_value>,
    "auditsyslogpolicy_csvserver_binding":<auditsyslogpolicy_csvserver_binding[]_value>,
    "auditsyslogpolicy_systemglobal_binding":<auditsyslogpolicy_systemglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy_csvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"auditsyslogpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy_lbvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditsyslogpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_rnatglobal_binding

May 20, 2016

Binding object showing the rnatglobal that can be bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_rnatglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_rnatglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_rnatglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_rnatglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_rnatglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy_rnatglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_rnatglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditsyslogpolicy_rnatglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_systemglobal_binding

May 20, 2016

Binding object showing the systemglobal that can be bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_systemglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_systemglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_systemglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy_systemglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditsyslogpolicy_systemglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_tmglobal_binding

May 20, 2016

Binding object showing the tmglobal that can be bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_tmglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_tmglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_tmglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_tmglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_tmglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy_tmglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_tmglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditsyslogpolicy_tmglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_vpnglobal_binding

May 20, 2016

Binding object showing the vpnglobal that can be bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy_vpnglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditsyslogpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

auditsyslogpolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to auditsyslogpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of auditsyslogpolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the auditsyslogpolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "auditsyslogpolicy_vpnvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/auditsyslogpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"auditsyslogpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Authentication

May 20, 2016

Authentication configuration.

Name	Description
authenticationauthnprofile	Configuration for Authentication profile
authenticationcertaction	Configuration for CERT action
authenticationcertpolicy	Configuration for CERT policy
authenticationcertpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationcertpolicy
authenticationcertpolicy_binding	Binding object showing the resources that can be bound to authenticationcertpolicy
authenticationcertpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationcertpolicy
authenticationcertpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationcertpolicy
authenticationdfaaction	Configuration for Dfa authentication action
authenticationdfapolicy	Configuration for Dfa authentication policy
authenticationdfapolicy_binding	Binding object showing the resources that can be bound to authenticationdfapolicy
authenticationdfapolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationdfapolicy
authenticationldapaction	Configuration for LDAP action
authenticationldappolicy	Configuration for LDAP policy
authenticationldappolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationldappolicy
authenticationldappolicy_binding	Binding object showing the resources that can be bound to authenticationldappolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
authenticationldappolicy_systemglobal_binding	authenticationldappolicy
authenticationldappolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationldappolicy
authenticationldappolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationldappolicy
authenticationlocalpolicy	Configuration for LOCAL policy
authenticationlocalpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationlocalpolicy
authenticationlocalpolicy_binding	Binding object showing the resources that can be bound to authenticationlocalpolicy
authenticationlocalpolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to authenticationlocalpolicy
authenticationlocalpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationlocalpolicy
authenticationlocalpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationlocalpolicy
authenticationloginschema	
authenticationloginschemapolicy	
authenticationloginschemapolicy_authenticationvserver_binding	
authenticationloginschemapolicy_binding	
authenticationloginschemapolicy_vpnvserver_binding	
authenticationnegotiateaction	Configuration for Negotiate action
authenticationnegotiatepolicy	Configuration for Negotiate Policy
authenticationnegotiatepolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationnegotiatepolicy
authenticationnegotiatepolicy_binding	Binding object showing the resources that can be bound to authenticationnegotiatepolicy
authenticationoauthaction	Configuration for OAuth authentication action

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
authenticationpolicy	Configuration for Authentication Policy
authenticationpolicy_authenticationpolicylabel_binding	Binding object showing the authenticationpolicylabel that can be bound to authenticationpolicy
authenticationpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationpolicy
authenticationpolicy_binding	Binding object showing the resources that can be bound to authenticationpolicy
authenticationpolicylabel	Configuration for authentication policy label
authenticationpolicylabel_authenticationpolicy_binding	Binding object showing the authenticationpolicy that can be bound to authenticationpolicylabel
authenticationpolicylabel_binding	Binding object showing the resources that can be bound to authenticationpolicylabel
authenticationradiusaction	Configuration for RADIUS action
authenticationradiuspolicy	Configuration for RADIUS policy
authenticationradiuspolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationradiuspolicy
authenticationradiuspolicy_binding	Binding object showing the resources that can be bound to authenticationradiuspolicy
authenticationradiuspolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to authenticationradiuspolicy
authenticationradiuspolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationradiuspolicy
authenticationradiuspolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationradiuspolicy
authenticationsamlaction	Configuration for AAA Saml action
authenticationsamlidppolicy	Configuration for AAA Saml IdentityProvider (IdP) policy
authenticationsamlidppolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationsamlidppolicy
authenticationsamlidppolicy_binding	Binding object showing the resources that can be bound to authenticationsamlidppolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
authenticationsamlidppolicy_vpnvserver_binding	authenticationsamlidppolicy
authenticationsamlidpprofile	Configuration for AAA Saml IdentityProvider (IdP) profile
authenticationsamlpolicy	Configuration for AAA Saml policy
authenticationsamlpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationsamlpolicy
authenticationsamlpolicy_binding	Binding object showing the resources that can be bound to authenticationsamlpolicy
authenticationsamlpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationsamlpolicy
authenticationsamlpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationsamlpolicy
authenticationtacacsaction	Configuration for TACACS action
authenticationtacacspolicy	Configuration for TACACS policy
authenticationtacacspolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationtacacspolicy
authenticationtacacspolicy_binding	Binding object showing the resources that can be bound to authenticationtacacspolicy
authenticationtacacspolicy_systemglobal_binding	Binding object showing the systemglobal that can be bound to authenticationtacacspolicy
authenticationtacacspolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationtacacspolicy
authenticationtacacspolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationtacacspolicy
authenticationvserver	Configuration for authentication virtual server
authenticationvserver_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to authenticationvserver
authenticationvserver_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to authenticationvserver
authenticationvserver_authenticationcertpolicy_binding	Binding object showing the authenticationcertpolicy that can be bound to authenticationvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationldappolicy_binding	bound to authenticationvserver
Name	**Description**
authenticationvserver_authenticationlocalpolicy_binding	Binding object showing the authenticationlocalpolicy that can be bound to authenticationvserver
authenticationvserver_authenticationloginschemapolicy_binding	Binding object showing the authenticationloginschemapolicy that can be bound to authenticationvserver
authenticationvserver_authenticationnegotiatepolicy_binding	Binding object showing the authenticationnegotiatepolicy that can be bound to authenticationvserver
authenticationvserver_authenticationpolicy_binding	Binding object showing the authenticationpolicy that can be bound to authenticationvserver
authenticationvserver_authenticationradiuspolicy_binding	Binding object showing the authenticationradiuspolicy that can be bound to authenticationvserver
authenticationvserver_authenticationsamlidppolicy_binding	Binding object showing the authenticationsamlidppolicy that can be bound to authenticationvserver
authenticationvserver_authenticationsamlpolicy_binding	Binding object showing the authenticationsamlpolicy that can be bound to authenticationvserver
authenticationvserver_authenticationtacacspolicy_binding	Binding object showing the authenticationtacacspolicy that can be bound to authenticationvserver
authenticationvserver_authenticationwebauthpolicy_binding	Binding object showing the authenticationwebauthpolicy that can be bound to authenticationvserver
authenticationvserver_binding	Binding object showing the resources that can be bound to authenticationvserver
authenticationvserver_cspolicy_binding	Binding object showing the cspolicy that can be bound to authenticationvserver
authenticationvserver_tmsessionpolicy_binding	Binding object showing the tmsessionpolicy that can be bound to authenticationvserver
authenticationvserver_vpnportaltheme_binding	Binding object showing the vpnportaltheme that can be bound to authenticationvserver
authenticationwebauthaction	Configuration for Web authentication action
authenticationwebauthpolicy	Configuration for Web authentication policy
authenticationwebauthpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to authenticationwebauthpolicy
authenticationwebauthpolicy_binding	Binding object showing the resources that can be bound to authenticationwebauthpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationwebauthpolicy_systemglobal_binding	
Name	**Description**
authenticationwebauthpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to authenticationwebauthpolicy
authenticationwebauthpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to authenticationwebauthpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationcertaction

May 20, 2016

Configuration for CERT action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the client cert authentication server profile (action). Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after certifcate action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1
twofactor	<String>	Read-write	Enables or disables two-factor authentication. Two factor authentication is client cert authentication followed by password authentication. Default value: OFF Possible values = ON, OFF
usernamefield	<String>	Read-write	Client-cert field from which the username is extracted. Must be set to either ""Subject"" and ""Issuer"" (include both sets of double quotation marks). Format: <field>:<subfield>. Minimum length = 1
groupnamefield	<String>	Read-write	Client-cert field from which the group is extracted. Must be set to either ""Subject"" and ""Issuer"" (include both sets of double quotation marks). Format: <field>:<subfield>. Minimum length = 1
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationcertaction":{
    "name":<String_value>,
    "twofactor":<String_value>,
    "usernamefield":<String_value>,
    "groupnamefield":<String_value>,
    "defaultauthenticationgroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationcertaction":{
    "name":<String_value>,
    "twofactor":<String_value>,
    "usernamefield":<String_value>,
    "groupnamefield":<String_value>,
    "defaultauthenticationgroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction?**action=unset**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationcertaction":{
    "name":<String_value>,
    "twofactor":true,
    "usernamefield":true,
    "groupnamefield":true,
    "defaultauthenticationgroup":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationcertaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationcertaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationcertaction": [ {
    "name":<String_value>,
    "twofactor":<String_value>,
    "usernamefield":<String_value>,
    "groupnamefield":<String_value>,
    "defaultauthenticationgroup":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationcertaction": [ {
    "name":<String_value>,
    "twofactor":<String_value>,
    "usernamefield":<String_value>,
    "groupnamefield":<String_value>,
    "defaultauthenticationgroup":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationcertaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationcertpolicy

May 20, 2016

Configuration for CERT policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the client certificate authentication policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after cert authentication policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to determine whether to attempt to authenticate the user with the authentication server. Minimum length = 1
reqaction	<String>	Read-write	Name of the client cert authentication action to be performed if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationcertpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationcertpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationcertpolicy":{
    "name":<String_value>,
    "rule":true,
    "reqaction":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationcertpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationcertpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationcertpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationcertpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "reqaction":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationcertpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationcertpolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to authenticationcertpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the client cert authentication policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationcertpolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationcertpolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationcertpolicy_authenticationvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationcertpolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationcertpolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to authenticationcertpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the client cert authentication policy. Minimum length = 1
authenticationcertpolicy_vpnglobal_binding	<authenticationcertpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationcertpolicy.
authenticationcertpolicy_vpnvserver_binding	<authenticationcertpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationcertpolicy.
authenticationcertpolicy_authenticationvserver_binding	<authenticationcertpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationcertpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationcertpolicy_binding": [ {
      "name":<String_value>,
      "authenticationcertpolicy_authenticationvserver_binding":<authenticationcertpolicy_authenticationvserver_binding[]_value>,
      "authenticationcertpolicy_vpnvserver_binding":<authenticationcertpolicy_vpnvserver_binding[]_value>,
      "authenticationcertpolicy_vpnglobal_binding":<authenticationcertpolicy_vpnglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationcertpolicy_vpnglobal_binding

May 20, 2016

Binding object showing the vpnglobal that can be bound to authenticationcertpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the client cert authentication policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationcertpolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationcertpolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationcertpolicy_vpnglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationcertpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationcertpolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to authenticationcertpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the client cert authentication policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationcertpolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationcertpolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationcertpolicy_vpnvserver_binding": [ {
      "name":<String_value>,
      "boundto":<String_value>,
      "priority":<Double_value>,
      "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationcertpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationcertpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationdfaaction

May 20, 2016

Configuration for Dfa authentication action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the DFA action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the DFA action is added. Minimum length = 1
clientid	<String>	Read-write	If configured, this string is sent to the DFA server as the X-Citrix-Exchange header value.
serverurl	<String>	Read-write	DFA Server URL.
passphrase	<String>	Read-write	Key shared between the DFA server and the NetScaler appliance. Required to allow the NetScaler appliance to communicate with the DFA server. Minimum length = 1
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
success	<Double>	Read-only	.
failure	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationdfaaction":{
    "name":<String_value>,
    "clientid":<String_value>,
    "serverurl":<String_value>,
    "passphrase":<String_value>,
    "defaultauthenticationgroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationdfaaction":{
    "name":<String_value>,
    "clientid":<String_value>,
    "serverurl":<String_value>,
    "passphrase":<String_value>,
    "defaultauthenticationgroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"authenticationdfaaction":{
    "name":<String_value>,
    "clientid":true,
    "serverurl":true,
    "defaultauthenticationgroup":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationdfaaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationdfaaction resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationdfaaction": [ {
      "name":<String_value>,
      "passphrase":<String_value>,
      "clientid":<String_value>,
      "serverurl":<String_value>,
      "success":<Double_value>,
      "failure":<Double_value>,
      "defaultauthenticationgroup":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationdfaaction": [ {
      "name":<String_value>,
      "passphrase":<String_value>,
      "clientid":<String_value>,
      "serverurl":<String_value>,
      "success":<Double_value>,
      "failure":<Double_value>,
      "defaultauthenticationgroup":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfaaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationdfaaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationdfapolicy

Jul 07, 2017

Configuration for Dfa authentication policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the DFA policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after DFA policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to determine whether to attempt to authenticate the user with the Web server. Minimum length = 1
action	<String>	Read-write	Name of the DFA action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationdfapolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationdfapolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationdfapolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationdfapolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationdfapolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationdfapolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "authenticationdfapolicy": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationradiuspolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to authenticationradiuspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the RADIUS authentication policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationradiuspolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationradiuspolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationradiuspolicy_authenticationvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationradiuspolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationdfapolicy_binding

May 20, 2016

Binding object which returns the resources bound to authenticationdfapolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
authenticationdfapolicy_vpnvserver_binding	<authenticationdfapolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationdfapolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationdfapolicy_binding": [ {
    "name":<String_value>,
    "authenticationdfapolicy_vpnvserver_binding":<authenticationdfapolicy_vpnvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationdfapolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to authenticationdfapolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationdfapolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationdfapolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationdfapolicy_vpnvserver_binding": [ {
      "name":<String_value>,
      "boundto":<String_value>,
      "priority":<Double_value>,
      "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationdfapolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationdfapolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationldapaction

Jul 07, 2017

Configuration for LDAP action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the new LDAP action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the LDAP action is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1
serverip	<String>	Read-write	IP address assigned to the LDAP server. Minimum length = 1
servername	<String>	Read-write	LDAP server name as a FQDN. Mutually exclusive with LDAP IP address. Minimum length = 1
serverport	<Integer>	Read-write	Port on which the LDAP server accepts connections. Default value: 389 Minimum value = 1
authtimeout	<Double>	Read-write	Number of seconds the NetScaler appliance waits for a response from the RADIUS server. Default value: 3 Minimum value = 1
ldapbase	<String>	Read-write	Base (node) from which to start LDAP searches. If the LDAP server is running locally, the default value of base is dc=netscaler, dc=com.
ldapbinddn	<String>	Read-write	Full distinguished name (DN) that is used to bind to the LDAP server. Default: cn=Manager,dc=netscaler,dc=com.
ldapbinddnpassword	<String>	Read-write	Password used to bind to the LDAP server. Minimum length = 1
ldaploginname	<String>	Read-write	LDAP login name attribute. The NetScaler appliance uses the LDAP login name to query external LDAP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
searchfilter	<String>	Read-write	String to be combined with the default LDAP user search string to form the search value. For example, if the search filter "vpnallowed=true" is combined with the LDAP login name "samaccount" and the user-supplied username is "bob", the result is the LDAP search string ""(&(vpnallowed=true)(samaccount=bob)"" (Be sure to enclose the search string in two sets of double quotation marks; both sets are needed.). Minimum length = 1
groupattrname	<String>	Read-write	LDAP group attribute name. Used for group extraction on the LDAP server.
subattributename	<String>	Read-write	LDAP group sub-attribute name. Used for group extraction from the LDAP server.
sectype	<String>	Read-write	Type of security used for communications between the NetScaler appliance and the LDAP server. For the PLAINTEXT setting, no encryption is required. Default value: PLAINTEXT Possible values = PLAINTEXT, TLS, SSL
svrtype	<String>	Read-write	The type of LDAP server. Default value: AAA_LDAP_SERVER_TYPE_DEFAULT Possible values = AD, NDS
ssonameattribute	<String>	Read-write	LDAP single signon (SSO) attribute. The NetScaler appliance uses the SSO name attribute to query external LDAP servers or Active Directories for an alternate username.
authentication	<String>	Read-write	Perform LDAP authentication. If authentication is disabled, any LDAP authentication attempt returns authentication success if the user is found. CAUTION! Authentication should be disabled only for authorization group extraction or where other (non-LDAP) authentication methods are in use and either bound to a primary list or flagged as secondary. Default value: ENABLED Possible values = ENABLED, DISABLED
requireuser	<String>	Read-write	Require a successful user search for authentication. Default value: YES Possible values = YES, NO
passwdchange	<String>	Read-write	Allow password change requests. Default value: DISABLED Possible values = ENABLED, DISABLED
nestedgroupextraction	<String>	Read-write	Allow nested group extraction, in which the NetScaler appliance queries external LDAP servers to determine whether a group is part of another group. Default value: OFF Possible values = ON, OFF
			If nested group extraction is ON, specifies the number of levels up to which group extraction is performed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
followreferrals	<String>	Read-write	Minimum value = 2 Setting this option to ON enables following LDAP referrals received from the LDAP server. Default value: OFF Possible values = ON, OFF
maxldapreferrals	<Double>	Read-write	Specifies the maximum number of nested referrals to follow. Default value: 1 Minimum value = 1
referraldnslookup	<String>	Read-write	Specifies the DNS Record lookup Type for the referrals. Default value: A-REC Possible values = A-REC, SRV-REC, MSSRV-REC
mssrvrecordlocation	<String>	Read-write	MSSRV Specific parameter. Used to locate the DNS node to which the SRV record pertains in the domainname. The domainname is appended to it to form the srv record. Example : For "dc._msdcs", the srv record formed is _ldap._tcp.dc._msdcs.<domainname>.
validateservercert	<String>	Read-write	When to validate LDAP server certs. Default value: NO Possible values = YES, NO
ldaphostname	<String>	Read-write	Hostname for the LDAP server. If -validateServerCert is ON then this must be the host name on the certificate from the LDAP server. A hostname mismatch will cause a connection failure.
groupnameidentifier	<String>	Read-write	Name that uniquely identifies a group in LDAP or Active Directory.
groupsearchattribute	<String>	Read-write	LDAP group search attribute. Used to determine to which groups a group belongs.
groupsearchsubattribute	<String>	Read-write	LDAP group search subattribute. Used to determine to which groups a group belongs.
groupsearchfilter	<String>	Read-write	String to be combined with the default LDAP group search string to form the search value. For example, the group search filter ""vpnallowed=true"" when combined with the group identifier ""samaccount"" and the group name ""g1"" yields the LDAP search string ""(&(vpnallowed=true)(samaccount=g1)"". (Be sure to enclose the search string in two sets of double quotation marks; both sets are needed.).
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
attribute1	<String>	Read-write	Expression that would be evaluated to extract attribute1 from the ldap response.
attribute2	<String>	Read-write	Expression that would be evaluated to extract attribute2 from the ldap response.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute3	<String>	Read-write	response.
attribute4	<String>	Read-write	Expression that would be evaluated to extract attribute4 from the ldap response.
attribute5	<String>	Read-write	Expression that would be evaluated to extract attribute5 from the ldap response.
attribute6	<String>	Read-write	Expression that would be evaluated to extract attribute6 from the ldap response.
attribute7	<String>	Read-write	Expression that would be evaluated to extract attribute7 from the ldap response.
attribute8	<String>	Read-write	Expression that would be evaluated to extract attribute8 from the ldap response.
attribute9	<String>	Read-write	Expression that would be evaluated to extract attribute9 from the ldap response.
attribute10	<String>	Read-write	Expression that would be evaluated to extract attribute10 from the ldap response.
attribute11	<String>	Read-write	Expression that would be evaluated to extract attribute11 from the ldap response.
attribute12	<String>	Read-write	Expression that would be evaluated to extract attribute12 from the ldap response.
attribute13	<String>	Read-write	Expression that would be evaluated to extract attribute13 from the ldap response.
attribute14	<String>	Read-write	Expression that would be evaluated to extract attribute14 from the ldap response.
attribute15	<String>	Read-write	Expression that would be evaluated to extract attribute15 from the ldap response.
attribute16	<String>	Read-write	Expression that would be evaluated to extract attribute16 from the ldap response.
success	<Double>	Read-only	.
failure	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationldapaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "servername":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "ldapbase":<String_value>,
    "ldapbinddn":<String_value>,
    "ldapbinddnpassword":<String_value>,
    "ldaploginname":<String_value>,
    "searchfilter":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "groupattrname":<String_value>,
    "subattributename":<String_value>,
    "sectype":<String_value>,
    "svrtype":<String_value>,
    "ssonameattribute":<String_value>,
    "authentication":<String_value>,
    "requireuser":<String_value>,
    "passwdchange":<String_value>,
    "nestedgroupextraction":<String_value>,
    "maxnestinglevel":<Double_value>,
    "followreferrals":<String_value>,
    "maxldapreferrals":<Double_value>,
    "referraldnslookup":<String_value>,
    "mssrvrecordlocation":<String_value>,
    "validateservercert":<String_value>,
    "ldaphostname":<String_value>,
    "groupnameidentifier":<String_value>,
    "groupsearchattribute":<String_value>,
    "groupsearchsubattribute":<String_value>,
    "groupsearchfilter":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationldapaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "servername":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "ldapbase":<String_value>,
    "ldapbinddn":<String_value>,
    "ldapbinddnpassword":<String_value>,
    "ldaploginname":<String_value>,
    "searchfilter":<String_value>,
    "groupattrname":<String_value>,
    "subattributename":<String_value>,
    "sectype":<String_value>,
    "svrtype":<String_value>,
    "ssonameattribute":<String_value>,
    "authentication":<String_value>,
    "requireuser":<String_value>,
    "passwdchange":<String_value>,
    "validateservercert":<String_value>,
    "ldaphostname":<String_value>,
    "nestedgroupextraction":<String_value>,
    "maxnestinglevel":<Double_value>,
    "groupnameidentifier":<String_value>,
    "groupsearchattribute":<String_value>,
    "groupsearchsubattribute":<String_value>,
    "groupsearchfilter":<String_value>,
    "followreferrals":<String_value>,
    "maxldapreferrals":<Double_value>,
    "referraldnslookup":<String_value>,
    "mssrvrecordlocation":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationldapaction":{
    "name":<String_value>,
    "serverport":true,
    "authtimeout":true,
    "ldapbase":true,
    "ldapbinddn":true,
    "ldapbinddnpassword":true,
    "ldaploginname":true,
    "searchfilter":true,
    "groupattrname":true,
    "subattributename":true,
    "sectype":true,
    "svrtype":true,
    "ssonameattribute":true,
    "authentication":true,
    "requireuser":true,
    "passwdchange":true,
    "validateservercert":true,
    "ldaphostname":true,
    "nestedgroupextraction":true,
    "maxnestinglevel":true,
    "groupnameidentifier":true,
    "groupsearchattribute":true,
    "groupsearchsubattribute":true,
    "groupsearchfilter":true,
    "followreferrals":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "maxldapreferrals":true,
        "referraldnslookup":true,
        "mssrvrecordlocation":true,
        "defaultauthenticationgroup":true,
        "attribute1":true,
        "attribute2":true,
        "attribute3":true,
        "attribute4":true,
        "attribute5":true,
        "attribute6":true,
        "attribute7":true,
        "attribute8":true,
        "attribute9":true,
        "attribute10":true,
        "attribute11":true,
        "attribute12":true,
        "attribute13":true,
        "attribute14":true,
        "attribute15":true,
        "attribute16":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationldapaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationldapaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationldapaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "servername":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "ldapbinddn":<String_value>,
    "ldapbinddnpassword":<String_value>,
    "ldaploginname":<String_value>,
    "ldapbase":<String_value>,
    "searchfilter":<String_value>,
    "groupattrname":<String_value>,
    "subattributename":<String_value>,
    "sectype":<String_value>,
    "svrtype":<String_value>,
    "ssonameattribute":<String_value>,
    "authentication":<String_value>,
    "requireuser":<String_value>,
    "success":<Double_value>,
    "failure":<Double_value>,
    "nestedgroupextraction":<String_value>,
    "maxnestinglevel":<Double_value>,
    "followreferrals":<String_value>,
    "maxldapreferrals":<Double_value>,
    "referraldnslookup":<String_value>,
    "mssrvrecordlocation":<String_value>,
    "validateservercert":<String_value>,
    "ldaphostname":<String_value>,
    "groupnameidentifier":<String_value>,
    "groupsearchattribute":<String_value>,
    "groupsearchsubattribute":<String_value>,
    "groupsearchfilter":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "passwdchange":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}}}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationldapaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "servername":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "ldapbinddn":<String_value>,
    "ldapbinddnpassword":<String_value>,
    "ldaploginname":<String_value>,
    "ldapbase":<String_value>,
    "searchfilter":<String_value>,
    "groupattrname":<String_value>,
    "subattributename":<String_value>,
    "sectype":<String_value>,
    "svrtype":<String_value>,
    "ssonameattribute":<String_value>,
    "authentication":<String_value>,
    "requireuser":<String_value>,
    "success":<Double_value>,
    "failure":<Double_value>,
    "nestedgroupextraction":<String_value>,
    "maxnestinglevel":<Double_value>,
    "followreferrals":<String_value>,
    "maxldapreferrals":<Double_value>,
    "referraldnslookup":<String_value>,
    "mssrvrecordlocation":<String_value>,
    "validateservercert":<String_value>,
    "ldaphostname":<String_value>,
    "groupnameidentifier":<String_value>,
    "groupsearchattribute":<String_value>,
    "groupsearchsubattribute":<String_value>,
    "groupsearchfilter":<String_value>,
    "passwdchange":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldapaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "authenticationldapaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationldappolicy

May 20, 2016

Configuration for LDAP policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the LDAP policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after LDAP policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to determine whether to attempt to authenticate the user with the LDAP server. Minimum length = 1
reqaction	<String>	Read-write	Name of the LDAP action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationldappolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationldappolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationldappolicy":{
    "name":<String_value>,
    "rule":true,
    "reqaction":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationldappolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationldappolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationldappolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationldappolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "reqaction":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationldappolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationldappolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to authenticationldappolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LDAP policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationldappolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationldappolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationldappolicy_authenticationvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationldappolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationldappolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to authenticationldappolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the LDAP policy. Minimum length = 1
authenticationldappolicy_vpnvserver_binding	<authenticationldappolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationldappolicy.
authenticationldappolicy_authenticationvserver_binding	<authenticationldappolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationldappolicy.
authenticationldappolicy_systemglobal_binding	<authenticationldappolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationldappolicy.
authenticationldappolicy_vpnglobal_binding	<authenticationldappolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationldappolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationldappolicy_binding": [ {
    "name":<String_value>,
    "authenticationldappolicy_authenticationvserver_binding":<authenticationldappolicy_authenticationvserver_binding[]_value>,
    "authenticationldappolicy_systemglobal_binding":<authenticationldappolicy_systemglobal_binding[]_value>,
    "authenticationldappolicy_vpnvserver_binding":<authenticationldappolicy_vpnvserver_binding[]_value>,
    "authenticationldappolicy_vpnglobal_binding":<authenticationldappolicy_vpnglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationldappolicy_systemglobal_binding

May 20, 2016

Binding object showing the systemglobal that can be bound to authenticationldappolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LDAP policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationldappolicy_systemglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_systemglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationldappolicy_systemglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationldappolicy_systemglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationldappolicy_systemglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationldappolicy_vpnglobal_binding

May 20, 2016

Binding object showing the vpnglobal that can be bound to authenticationldappolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LDAP policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationldappolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationldappolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationldappolicy_vpnglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationldappolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationldappolicy_vpnvserver_binding

Jul 07, 2017

Binding object showing the vpnvserver that can be bound to authenticationldappolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LDAP policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationldappolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationldappolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationldappolicy_vpnvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationldappolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationldappolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationpolicy

May 20, 2016

Configuration for Authentication Policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the advance AUTHENTICATION policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after AUTHENTICATION policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to determine whether to attempt to authenticate the user with the AUTHENTICATION server.
action	<String>	Read-write	Name of the authentication action to be performed if the policy matches.
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.
comment	<String>	Read-write	Any comments to preserve information about this policy.
logaction	<String>	Read-write	Name of messagelog action to use when a request matches this policy.
newname	<String>	Read-write	New name for the authentication policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
description	<String>	Read-only	Description of the policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
policysubtype	<String>	Read-only	. Possible values = LOCAL, RADIUS, LDAP, TACACS, CERT, NEGOTIATE, SAML, PROFILE, DFA, SAMLIDP, WEBAUTH, OAUTH, NO_AUTHN
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"authenticationpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationpolicy":{
    "name":<String_value>,
    "undefaction":true,
    "comment":true,
    "logaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationpolicy":{
    "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "undefaction":<String_value>,
      "comment":<String_value>,
      "logaction":<String_value>,
      "hits":<Double_value>,
      "description":<String_value>,
      "policysubtype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "hits":<Double_value>,
    "description":<String_value>,
    "policysubtype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicy": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationlocalpolicy

May 20, 2016

Configuration for LOCAL policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the local authentication policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after local policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy').
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to perform the authentication.
reqaction	<String>	Read-only	The name of the RADIUS action the policy uses.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationlocalpolicy":{
    "name":<String_value>,
    "rule":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationlocalpolicy":{
    "name":<String_value>,
    "rule":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationlocalpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationlocalpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationlocalpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "reqaction":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationlocalpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationlocalpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationlocalpolicy_binding

May 20, 2016

Binding object which returns the resources bound to authenticationlocalpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the local authentication policy.
authenticationlocalpolicy_vpnglobal_binding	<authenticationlocalpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationlocalpolicy.
authenticationlocalpolicy_vpnvserver_binding	<authenticationlocalpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationlocalpolicy.
authenticationlocalpolicy_systemglobal_binding	<authenticationlocalpolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationlocalpolicy.
authenticationlocalpolicy_authenticationvserver_binding	<authenticationlocalpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationlocalpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationlocalpolicy_binding": [ {
    "name":<String_value>,
    "authenticationlocalpolicy_vpnvserver_binding":<authenticationlocalpolicy_vpnvserver_binding[]_value>,
    "authenticationlocalpolicy_systemglobal_binding":<authenticationlocalpolicy_systemglobal_binding[]_value>,
    "authenticationlocalpolicy_vpnglobal_binding":<authenticationlocalpolicy_vpnglobal_binding[]_value>,
    "authenticationlocalpolicy_authenticationvserver_binding":<authenticationlocalpolicy_authenticationvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationlocalpolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to authenticationlocalpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the local authentication policy.
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationlocalpolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationlocalpolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationlocalpolicy_authenticationvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationlocalpolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationlocalpolicy_systemglobal_binding

May 20, 2016

Binding object showing the systemglobal that can be bound to authenticationlocalpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the local authentication policy.
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationlocalpolicy_systemglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_systemglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationlocalpolicy_systemglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationlocalpolicy_systemglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationlocalpolicy_systemglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationlocalpolicy_vpnglobal_binding

Jul 07, 2017

Binding object showing the vpnglobal that can be bound to authenticationlocalpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the local authentication policy.
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationlocalpolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationlocalpolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationlocalpolicy_vpnglobal_binding": [ {
      "name":<String_value>,
      "boundto":<String_value>,
      "priority":<Double_value>,
      "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationlocalpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationlocalpolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to authenticationlocalpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the local authentication policy.
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationlocalpolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationlocalpolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationlocalpolicy_vpnvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationlocalpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationlocalpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationloginschema

May 20, 2016

Configuration for 0 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the new login schema. Login schema defines the way login form is rendered. It provides a way to customize the fields that are shown to the user. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
authenticationschema	<String>	Read-write	Name of the file for reading authentication schema to be sent for Login Page UI. This file should contain xml definition of elements as per Citrix Forms Authentication Protocol to be able to render login form. If administrator does not want to prompt users for additional credentials but continue with previously obtained credentials, then "noschema" can be given as argument. Please note that this applies only to loginSchemas that are used with user-defined factors, and not the vserver factor. Minimum length = 1
userexpression	<String>	Read-write	Expression for username extraction during login. Minimum length = 1
passwdexpression	<String>	Read-write	Expression for password extraction during login. Minimum length = 1
usercredentialindex	<Double>	Read-write	The index at which user entered username should be stored in session. Minimum value = 1 Maximum value = 16
passwordcredentialindex	<Double>	Read-write	The index at which user entered password should be stored in session. Minimum value = 1 Maximum value = 16
authenticationstrength	<Double>	Read-write	Weight of the current authentication. Minimum value = 0 Maximum value = 65535

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
Name	**Data Type**	**Permissions**	**Description**
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationloginschema":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "authenticationschema":<String_value>,
    "userexpression":<String_value>,
    "passwdexpression":<String_value>,
    "usercredentialindex":<Double_value>,
    "passwordcredentialindex":<Double_value>,
    "authenticationstrength":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationloginschema":{
    "name":<String_value>,
    "authenticationschema":<String_value>,
    "userexpression":<String_value>,
    "passwdexpression":<String_value>,
    "usercredentialindex":<Double_value>,
    "passwordcredentialindex":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "authenticationstrength":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationloginschema":{
    "name":<String_value>,
    "userexpression":true,
    "passwdexpression":true,
    "usercredentialindex":true,
    "passwordcredentialindex":true,
    "authenticationstrength":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationloginschema resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationloginschema resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationloginschema": [ {
    "name":<String_value>,
    "authenticationschema":<String_value>,
    "userexpression":<String_value>,
    "passwdexpression":<String_value>,
    "usercredentialindex":<Double_value>,
    "passwordcredentialindex":<Double_value>,
    "authenticationstrength":<Double_value>,
    "builtin":<String[]_value>
}]}
```

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationloginschema": [ {
    "name":<String_value>,
    "authenticationschema":<String_value>,
    "userexpression":<String_value>,
    "passwdexpression":<String_value>,
    "usercredentialindex":<Double_value>,
    "passwordcredentialindex":<Double_value>,
    "authenticationstrength":<Double_value>,
    "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschema?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "authenticationloginschema": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationloginschemapolicy

May 20, 2016

Configuration for 0 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the LoginSchema policy. This is used for selecting parameters for user logon. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression which is evaluated to choose a profile for authentication. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks. Minimum length = 1
action	<String>	Read-write	Name of the profile to apply to requests or connections that match this policy. * NOOP - Do not take any specific action when this policy evaluates to true. This is useful to implicitly go to a different policy set. * RESET - Reset the client connection by closing it. The client program, such as a browser, will handle this and may inform the user. The client may then resend the request if desired. * DROP - Drop the request without sending a response to the user. Minimum length = 1
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.
comment	<String>	Read-write	Any comments to preserve information about this policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
newname	<String>	Read-write	New name for the LoginSchema policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my loginschemapolicy policy" or 'my loginschemapolicy policy'). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of Undef hits.
builtin	<String[]>	Read-only	Flag to determine if policy is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationloginschemapolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationloginschemapolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationloginschemapolicy":{
    "name":<String_value>,
    "undefaction":true,
    "comment":true,
    "logaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationloginschemapolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationloginschemapolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationloginschemapolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationloginschemapolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationloginschemapolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "undefaction":<String_value>,
      "comment":<String_value>,
      "logaction":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "authenticationloginschemapolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationloginschemapolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to authenticationloginschemapolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LoginSchema policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationloginschemapolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationloginschemapolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationloginschemapolicy_authenticationvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy_authenticationvserver_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationloginschemapolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationloginschemapolicy_binding

May 20, 2016

Binding object which returns the resources bound to authenticationloginschemapolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the LoginSchema policy for which to display detailed information. Minimum length = 1
authenticationloginschemapolicy_vpnvserver_binding	<authenticationloginschemapolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationloginschemapolicy.
authenticationloginschemapolicy_authenticationvserver_binding	<authenticationloginschemapolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationloginschemapolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

error

Response Payload:

```
{ "authenticationloginschemapolicy_binding": [ {
    "name":<String_value>,
    "authenticationloginschemapolicy_authenticationvserver_binding":<authenticationloginschemapolicy_authenticationvserver_binding[]_value>,
    "authenticationloginschemapolicy_vpnvserver_binding":<authenticationloginschemapolicy_vpnvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationloginschemapolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to authenticationloginschemapolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the LoginSchema policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationloginschemapolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationloginschemapolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationloginschemapolicy_vpnvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "labelname":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationloginschemapolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationloginschemapolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationnegotiateaction

May 20, 2016

Configuration for Negotiate action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the AD KDC server profile (negotiate action). Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after AD KDC server profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1
domain	<String>	Read-write	Domain name of the AD KDC server. Minimum length = 1
domainuser	<String>	Read-write	User name that the NetScaler appliance uses to join the AD KDC server domain. The NetScaler appliance uses the domain user name to check the health of the AD KDC server. Minimum length = 1
domainuserpasswd	<String>	Read-write	Password that the NetScaler appliance uses to join the AD KDC server domain. Minimum length = 1
ou	<String>	Read-write	Active Directory organizational units (OU) attribute. Minimum length = 1
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
keytab	<String>	Read-write	The path to the keytab file. Minimum length = 1
ntlmpath	<String>	Read-write	The path to the site that is enabled for NTLM authentication, including FQDN of the server. Minimum length = 1
kcdspn	<String>	Read-only	Host SPN extracted from keytab file.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
count	Boolean	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationnegotiateaction":{
    "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "domain":<String_value>,
    "domainuser":<String_value>,
    "domainuserpasswd":<String_value>,
    "ou":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "keytab":<String_value>,
    "ntlmpath":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationnegotiateaction":{
    "name":<String_value>,
    "domain":<String_value>,
    "domainuser":<String_value>,
    "domainuserpasswd":<String_value>,
    "ou":<String_value>,
    "defaultauthenticationgroup":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "keytab":<String_value>,
    "ntlmpath":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationnegotiateaction":{
    "name":<String_value>,
    "domain":true,
    "domainuser":true,
    "domainuserpasswd":true,
    "ou":true,
    "defaultauthenticationgroup":true,
    "ntlmpath":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationnegotiateaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationnegotiateaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationnegotiateaction": [ {
    "name":<String_value>,
    "domain":<String_value>,
    "domainuser":<String_value>,
    "domainuserpasswd":<String_value>,
    "ou":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "keytab":<String_value>,
    "kcdspn":<String_value>,
    "ntlmpath":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationnegotiateaction": [ {
    "name":<String_value>,
    "domain":<String_value>,
    "domainuser":<String_value>,
    "domainuserpasswd":<String_value>,
    "ou":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "keytab":<String_value>,
    "kcdspn":<String_value>,
    "ntlmpath":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiateaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationnegotiateaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationnegotiatepolicy

Jul 07, 2017

Configuration for Negotiate Policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the negotiate authentication policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after AD KCD (negotiate) policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to determine whether to attempt to authenticate the user with the AD KCD server. Minimum length = 1
reqaction	<String>	Read-write	Name of the negotiate action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationnegotiatepolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationnegotiatepolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationnegotiatepolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationnegotiatepolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationnegotiatepolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationnegotiatepolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "reqaction":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationnegotiatepolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationnegotiatepolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to authenticationnegotiatepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the negotiate policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationnegotiatepolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationnegotiatepolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationnegotiatepolicy_authenticationvserver_binding": [ {
      "boundto":<String_value>,
      "name":<String_value>,
      "priority":<Double_value>,
      "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationnegotiatepolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationnegotiatepolicy_binding

May 20, 2016

Binding object which returns the resources bound to authenticationnegotiatepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the negotiate policy. Minimum length = 1
authenticationnegotiatepolicy_authenticationvserver_binding	<authenticationnegotiatepolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationnegotiatepolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationnegotiatepolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "authenticationnegotiatepolicy_binding": [ {
    "name":<String_value>,
    "authenticationnegotiatepolicy_authenticationvserver_binding":<authenticationnegotiatepolicy_authenticationvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationoauthaction

May 20, 2016

Configuration for OAuth authentication action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the OAuth Authentication action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1
authorizationendpoint	<String>	Read-write	Authorization endpoint/url to which unauthenticated user will be redirected. Netscaler appliance redirects user to this endpoint by adding query parameters including clientid.
tokenendpoint	<String>	Read-write	URL to which OAuth token will be posted to verify its authenticity. User obtains this token from Authorization server upon successful authentication. Netscaler appliance will validate presented token by posting it to the URL configured.
idtokendecryptendpoint	<String>	Read-write	URL to which obtained idtoken will be posted to get a decrypted user identity. Encrypted idtoken will be obtained by posting OAuth token to token endpoint. In order to decrypt idtoken, Netscaler appliance posts request to the URL configured.
clientid	<String>	Read-write	Unique identity of the client/user who is getting authenticated. Authorization server infers client configuration using this ID.
clientsecret	<String>	Read-write	Secret string established by user and authorization server.
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
attribute1	<String>	Read-write	Expression that would be evaluated to extract attribute1 from the oauth response.
			Expression that would be evaluated to extract attribute2 from the oauth

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute3	<String>	Read-write	Expression that would be evaluated to extract attribute3 from the oauth response.
attribute4	<String>	Read-write	Expression that would be evaluated to extract attribute4 from the oauth response.
attribute5	<String>	Read-write	Expression that would be evaluated to extract attribute5 from the oauth response.
attribute6	<String>	Read-write	Expression that would be evaluated to extract attribute6 from the oauth response.
attribute7	<String>	Read-write	Expression that would be evaluated to extract attribute7 from the oauth response.
attribute8	<String>	Read-write	Expression that would be evaluated to extract attribute8 from the oauth response.
attribute9	<String>	Read-write	Expression that would be evaluated to extract attribute9 from the oauth response.
attribute10	<String>	Read-write	Expression that would be evaluated to extract attribute10 from the oauth response.
attribute11	<String>	Read-write	Expression that would be evaluated to extract attribute11 from the oauth response.
attribute12	<String>	Read-write	Expression that would be evaluated to extract attribute12 from the oauth response.
attribute13	<String>	Read-write	Expression that would be evaluated to extract attribute13 from the oauth response.
attribute14	<String>	Read-write	Expression that would be evaluated to extract attribute14 from the oauth response.
attribute15	<String>	Read-write	Expression that would be evaluated to extract attribute15 from the oauth response.
attribute16	<String>	Read-write	Expression that would be evaluated to extract attribute16 from the oauth response.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationoauthaction":{
    "name":<String_value>,
    "authorizationendpoint":<String_value>,
    "tokenendpoint":<String_value>,
    "idtokendecryptendpoint":<String_value>,
    "clientid":<String_value>,
    "clientsecret":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationoauthaction":{
    "name":<String_value>,
    "authorizationendpoint":<String_value>,
    "tokenendpoint":<String_value>,
    "idtokendecryptendpoint":<String_value>,
    "clientid":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "clientsecret":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationoauthaction":{
    "name":<String_value>,
    "idtokendecryptendpoint":true,
    "defaultauthenticationgroup":true,
    "attribute1":true,
    "attribute2":true,
    "attribute3":true,
    "attribute4":true,
    "attribute5":true,
    "attribute6":true,
    "attribute7":true,
    "attribute8":true,
    "attribute9":true,
    "attribute10":true,
    "attribute11":true,
    "attribute12":true,
    "attribute13":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "attribute14":true,
        "attribute15":true,
        "attribute16":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationoauthaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationoauthaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationoauthaction": [ {
    "name":<String_value>,
    "authorizationendpoint":<String_value>,
    "tokenendpoint":<String_value>,
    "idtokendecryptendpoint":<String_value>,
    "clientid":<String_value>,
    "clientsecret":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction/name_value<String>?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationoauthaction": [ {
    "name":<String_value>,
    "authorizationendpoint":<String_value>,
    "tokenendpoint":<String_value>,
    "idtokendecryptendpoint":<String_value>,
    "clientid":<String_value>,
    "clientsecret":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationoauthaction?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "authenticationoauthaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationpolicylabel

May 20, 2016

Configuration for authentication policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name for the new authentication policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy label" or 'authentication policy label').
comment	<String>	Read-write	Any comments to preserve information about this authentication policy label.
loginschema	<String>	Read-write	Login schema associated with authentication policy label. Login schema defines the UI rendering by providing customization option of the fields. If user intervention is not needed for a given factor such as group extraction, a loginSchema whose authentication schema is "noschema" should be used.
newname	<String>	Read-write	The new name of the auth policy label. Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
policyname	<String>	Read-only	Name of the authentication policy to bind to the policy label.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
flowtype	<Double>	Read-only	Flowtype of the bound authentication policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
__count	<Double>	Read-Only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationpolicylabel":{
    "labelname":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "comment":<String_value>,
    "loginschema":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationpolicylabel":{
    "labelname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationpolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationpolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "authenticationpolicylabel": [ {
      "labelname":<String_value>,
      "numpol":<Double_value>,
      "hits":<Double_value>,
      "policyname":<String_value>,
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "flowtype":<Double_value>,
      "description":<String_value>,
      "comment":<String_value>,
      "loginschema":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "authenticationpolicylabel": [ {
    "labelname":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "flowtype":<Double_value>,
    "description":<String_value>,
    "comment":<String_value>,
    "loginschema":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicylabel": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationpolicylabel_authenticationpolicy_binding

May 20, 2016

Binding object showing the authenticationpolicy that can be bound to authenticationpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
nextfactor	<String>	Read-write	On success invoke label.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the authentication policy to bind to the policy label.
labelname	<String>	Read-write	Name of the authentication policy label to which to bind the policy.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationpolicylabel_authenticationpolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "nextfactor":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of authenticationpolicylabel_authenticationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationpolicylabel_authenticationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicylabel_authenticationpolicy_binding": [ {
    "priority":<Double_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel_authenticationpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"authenticationpolicylabel_authenticationpolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationpolicylabel_binding

May 20, 2016

Binding object which returns the resources bound to authenticationpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the authorization policy label.
authenticationpolicylabel_authenticationpolicy_binding	<authenticationpolicylabel_authenticationpolicy_binding[]>	Read-only	authenticationpolicy that can be bound to authenticationpolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationpolicylabel_binding": [ {
    "labelname":<String_value>,
    "authenticationpolicylabel_authenticationpolicy_binding":<authenticationpolicylabel_authenticationpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationpolicy_authenticationpolicylabel_binding

May 20, 2016

Binding object showing the authenticationpolicylabel that can be bound to authenticationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the advance authentication policy. Minimum length = 1
priority	<Double>	Read-only	.
nextfactor	<String>	Read-only	On success invoke label.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy_authenticationpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy_authenticationpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationpolicy_authenticationpolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy_authenticationpolicylabel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationpolicy_authenticationpolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicy_authenticationpolicylabel_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy_authenticationpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationpolicy_authenticationpolicylabel_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationpolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to authenticationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the advance authentication policy. Minimum length = 1
priority	<Double>	Read-only	.
nextfactor	<String>	Read-only	On success invoke label.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationpolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationpolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicy_authenticationvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Double_value>
}]}
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationpolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationpolicy_binding

May 20, 2016

Binding object which returns the resources bound to authenticationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the advance authentication policy. Minimum length = 1
authenticationpolicy_authenticationpolicylabel_binding	<authenticationpolicy_authenticationpolicylabel_binding[]>	Read-only	authenticationpolicylabel that can be bound to authenticationpolicy.
authenticationpolicy_authenticationvserver_binding	<authenticationpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationpolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationpolicy_binding": [ {
    "name":<String_value>,
    "authenticationpolicy_authenticationvserver_binding":<authenticationpolicy_authenticationvserver_binding[]_value>,
    "authenticationpolicy_authenticationpolicylabel_binding":<authenticationpolicy_authenticationpolicylabel_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationradiusaction

May 20, 2016

Configuration for RADIUS action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the RADIUS action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the RADIUS action is added. Minimum length = 1
serverip	<String>	Read-write	IP address assigned to the RADIUS server. Minimum length = 1
servername	<String>	Read-write	RADIUS server name as a FQDN. Mutually exclusive with RADIUS IP address. Minimum length = 1
serverport	<Integer>	Read-write	Port number on which the RADIUS server listens for connections. Minimum value = 1
authtimeout	<Double>	Read-write	Number of seconds the NetScaler appliance waits for a response from the RADIUS server. Default value: 3 Minimum value = 1
radkey	<String>	Read-write	Key shared between the RADIUS server and the NetScaler appliance. Required to allow the NetScaler appliance to communicate with the RADIUS server. Minimum length = 1
radnasip	<String>	Read-write	If enabled, the NetScaler appliance IP address (NSIP) is sent to the RADIUS server as the Network Access Server IP (NASIP) address. The RADIUS protocol defines the meaning and use of the NASIP address. Possible values = ENABLED, DISABLED
radnasid	<String>	Read-write	If configured, this string is sent to the RADIUS server as the Network Access Server ID (NASID).
radvendorid	<Double>	Read-write	RADIUS vendor ID attribute, used for RADIUS group extraction. Minimum value = 1
radattributetype	<Double>	Read-write	RADIUS attribute type, used for RADIUS group extraction. Minimum value = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
radgroupsprefix	<String>	Read-write	This groups prefix precedes the group names within a RADIUS attribute for RADIUS group extraction.
radgroupseparator	<String>	Read-write	RADIUS group separator string The group separator delimits group names within a RADIUS attribute for RADIUS group extraction.
passencoding	<String>	Read-write	Encoding type for passwords in RADIUS packets that the NetScaler appliance sends to the RADIUS server. Default value: pap Possible values = pap, chap, mschapv1, mschapv2
ipvendorid	<Double>	Read-write	Vendor ID of the intranet IP attribute in the RADIUS response. NOTE: A value of 0 indicates that the attribute is not vendor encoded.
ipattributetype	<Double>	Read-write	Remote IP address attribute type in a RADIUS response. Minimum value = 1
accounting	<String>	Read-write	Whether the RADIUS server is currently accepting accounting messages. Possible values = ON, OFF
pwdvendorid	<Double>	Read-write	Vendor ID of the attribute, in the RADIUS response, used to extract the user password. Minimum value = 1
pwdattributetype	<Double>	Read-write	Vendor-specific password attribute type in a RADIUS response. Minimum value = 1
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
callingstationid	<String>	Read-write	Send Calling-Station-ID of the client to the RADIUS server. IP Address of the client is sent as its Calling-Station-ID. Default value: DISABLED Possible values = ENABLED, DISABLED
ipaddress	<String>	Read-only	IP address.
success	<Double>	Read-only	.
failure	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationradiusaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "servername":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "radkey":<String_value>,
    "radnasip":<String_value>,
    "radnasid":<String_value>,
    "radvendorid":<Double_value>,
    "radattributetype":<Double_value>,
    "radgroupsprefix":<String_value>,
    "radgroupseparator":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "passencoding":<String_value>,
    "ipvendorid":<Double_value>,
    "ipattributetype":<Double_value>,
    "accounting":<String_value>,
    "pwdvendorid":<Double_value>,
    "pwdattributetype":<Double_value>,
    "defaultauthenticationgroup":<String_value>,
    "callingstationid":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationradiusaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "servername":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "radkey":<String_value>,
    "radnasip":<String_value>,
    "radnasid":<String_value>,
    "radvendorid":<Double_value>,
    "radattributetype":<Double_value>,
    "radgroupsprefix":<String_value>,
    "radgroupseparator":<String_value>,
    "passencoding":<String_value>,
    "ipvendorid":<Double_value>,
    "ipattributetype":<Double_value>,
    "accounting":<String_value>,
    "pwdvendorid":<Double_value>,
    "pwdattributetype":<Double_value>,
    "defaultauthenticationgroup":<String_value>,
    "callingstationid":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationradiusaction":{
    "name":<String_value>,
    "serverport":true,
    "authtimeout":true,
    "radnasip":true,
    "radnasid":true,
    "radvendorid":true,
    "radattributetype":true,
    "radgroupsprefix":true,
    "radgroupseparator":true,
    "passencoding":true,
    "ipvendorid":true,
    "ipattributetype":true,
    "accounting":true,
    "pwdvendorid":true,
    "pwdattributetype":true,
    "defaultauthenticationgroup":true,
    "callingstationid":true
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationradiusaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationradiusaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationradiusaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "servername":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "radkey":<String_value>,
    "radnasip":<String_value>,
    "ipaddress":<String_value>,
    "radnasid":<String_value>,
    "radvendorid":<Double_value>,
    "radattributetype":<Double_value>,
    "radgroupsprefix":<String_value>,
    "radgroupseparator":<String_value>,
    "passencoding":<String_value>,
    "ipvendorid":<Double_value>,
    "ipattributetype":<Double_value>,
    "accounting":<String_value>,
    "success":<Double_value>,
    "failure":<Double_value>,
    "pwdvendorid":<Double_value>,
    "pwdattributetype":<Double_value>,
    "defaultauthenticationgroup":<String_value>,
    "callingstationid":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationradiusaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "servername":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "radkey":<String_value>,
    "radnasip":<String_value>,
    "ipaddress":<String_value>,
    "radnasid":<String_value>,
    "radvendorid":<Double_value>,
    "radattributetype":<Double_value>,
    "radgroupsprefix":<String_value>,
    "radgroupseparator":<String_value>,
    "passencoding":<String_value>,
    "ipvendorid":<Double_value>,
    "ipattributetype":<Double_value>,
    "accounting":<String_value>,
    "success":<Double_value>,
    "failure":<Double_value>,
    "pwdvendorid":<Double_value>,
    "pwdattributetype":<Double_value>,
    "defaultauthenticationgroup":<String_value>,
    "callingstationid":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiusaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "authenticationradiusaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationradiuspolicy

May 20, 2016

Configuration for RADIUS policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the RADIUS authentication policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after RADIUS policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to determine whether to attempt to authenticate the user with the RADIUS server.
reqaction	<String>	Read-write	Name of the RADIUS action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationradiuspolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationradiuspolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationradiuspolicy":{
    "name":<String_value>,
    "rule":true,
    "reqaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationradiuspolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationradiuspolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationradiuspolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationradiuspolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "reqaction":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationradiuspolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationradiuspolicy_binding

May 20, 2016

Binding object which returns the resources bound to authenticationradiuspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the RADIUS authentication policy. Minimum length = 1
authenticationradiuspolicy_vpnglobal_binding	<authenticationradiuspolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationradiuspolicy.
authenticationradiuspolicy_vpnvserver_binding	<authenticationradiuspolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationradiuspolicy.
authenticationradiuspolicy_systemglobal_binding	<authenticationradiuspolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationradiuspolicy.
authenticationradiuspolicy_authenticationvserver_binding	<authenticationradiuspolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationradiuspolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationradiuspolicy_binding": [ {
    "name":<String_value>,
    "authenticationradiuspolicy_authenticationvserver_binding":<authenticationradiuspolicy_authenticationvserver_binding[]_value>,
    "authenticationradiuspolicy_vpnglobal_binding":<authenticationradiuspolicy_vpnglobal_binding[]_value>,
    "authenticationradiuspolicy_systemglobal_binding":<authenticationradiuspolicy_systemglobal_binding[]_value>,
    "authenticationradiuspolicy_vpnvserver_binding":<authenticationradiuspolicy_vpnvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationradiuspolicy_systemglobal_binding

Jul 07, 2017

Binding object showing the systemglobal that can be bound to authenticationradiuspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the RADIUS authentication policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationradiuspolicy_systemglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_systemglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationradiuspolicy_systemglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationradiuspolicy_systemglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationradiuspolicy_systemglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationwebauthpolicy_systemglobal_binding

May 20, 2016

Binding object showing the systemglobal that can be bound to authenticationwebauthpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationwebauthpolicy_systemglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_systemglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationwebauthpolicy_systemglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationwebauthpolicy_systemglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_systemglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationwebauthpolicy_systemglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationradiuspolicy_vpnglobal_binding

May 20, 2016

Binding object showing the vpnglobal that can be bound to authenticationradiuspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the RADIUS authentication policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationradiuspolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationradiuspolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationradiuspolicy_vpnglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationradiuspolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationradiuspolicy_vpnvserver_binding

Jul 07, 2017

Binding object showing the vpnvserver that can be bound to authenticationradiuspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the RADIUS authentication policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationradiuspolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationradiuspolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationradiuspolicy_vpnvserver_binding": [ {
      "name":<String_value>,
      "boundto":<String_value>,
      "priority":<Double_value>,
      "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationradiuspolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationradiuspolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlaction

May 20, 2016

Configuration for AAA Saml action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the SAML server profile (action). Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after SAML profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1
samlidpcertname	<String>	Read-write	Name of the SAML server as given in that server's SSL certificate. Minimum length = 1
samlsigningcertname	<String>	Read-write	Name of the signing authority as given in the SAML server's SSL certificate. Minimum length = 1
samlredirecturl	<String>	Read-write	URL to which users are redirected for authentication. Minimum length = 1
samlacsindex	<Double>	Read-write	Index/ID of the metadata entry corresponding to this configuration. Default value: 255 Minimum value = 0 Maximum value = 255
samluserfield	<String>	Read-write	SAML user ID, as given in the SAML assertion. Minimum length = 1
samlrejectunsignedassertion	<String>	Read-write	Reject unsigned SAML assertions. Default value: ON Possible values = ON, OFF, STRICT
samlissuername	<String>	Read-write	The name to be used in requests sent from Netscaler to IdP to uniquely identify Netscaler. Minimum length = 1
			Option to enable second factor after SAML.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Possible values = ON, OFF
Name	**Data Type**	**Permissions**	**Description**
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
attribute1	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute1.
attribute2	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute2.
attribute3	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute3.
attribute4	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute4.
attribute5	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute5.
attribute6	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute6.
attribute7	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute7.
attribute8	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute8.
attribute9	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute9.
attribute10	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute10.
attribute11	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute11.
attribute12	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute12.
attribute13	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute13.
attribute14	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute14.
attribute15	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute15.
attribute16	<String>	Read-write	Name of the attribute in SAML Assertion whose value needs to be extracted and stored as attribute16.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: RSA-SHA1 Possible values = RSA-SHA1, RSA-SHA256
digestmethod	<String>	Read-write	Algorithm to be used to compute/verify digest for SAML transactions. Default value: SHA1 Possible values = SHA1, SHA256
requestedauthncontext	<String>	Read-write	This element specifies the authentication context requirements of authentication statements returned in the response. Default value: exact Possible values = exact, minimum, maximum, better
authnctxclassref	<String[]>	Read-write	This element specifies the authentication class types that are requested from IdP (IdentityProvider). InternetProtocol: This is applicable when a principal is authenticated through the use of a provided IP address. InternetProtocolPassword: This is applicable when a principal is authenticated through the use of a provided IP address, in addition to a username/password. Kerberos: This is applicable when the principal has authenticated using a password to a local authentication authority, in order to acquire a Kerberos ticket. MobileOneFactorUnregistered: This indicates authentication of the mobile device without requiring explicit end-user interaction. MobileTwoFactorUnregistered: This indicates two-factor based authentication during mobile customer registration process, such as secure device and user PIN. MobileOneFactorContract: Reflects mobile contract customer registration procedures and a single factor authentication. MobileTwoFactorContract: Reflects mobile contract customer registration procedures and a two-factor based authentication. Password: This class is applicable when a principal authenticates using password over unprotected http session. PasswordProtectedTransport: This class is applicable when a principal authenticates to an authentication authority through the presentation of a password over a protected session. PreviousSession: This class is applicable when a principal had authenticated to an authentication authority at some point in the past using any authentication context. X509: This indicates that the principal authenticated by means of a digital signature where the key was validated as part of an X.509 Public Key Infrastructure. PGP: This indicates that the principal authenticated by means of a digital signature where the key was validated as part of a PGP Public Key Infrastructure. SPKI: This indicates that the principal authenticated by means of a digital signature where the key was validated via an SPKI Infrastructure. XMLDSig: This indicates that the principal authenticated by means of a digital signature according to the processing rules specified in the XML Digital Signature specification. Smartcard: This indicates that the principal has authenticated using smartcard. SmartcardPKI: This class is applicable when a principal authenticates to an authentication authority through a two-factor authentication mechanism using a smartcard with enclosed private key and a PIN.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			certificate stored in software to authenticate to the authentication authority. Telephony: This class is used to indicate that the principal authenticated via the provision of a fixed-line telephone number, transported via a telephony protocol such as ADSL. NomadTelephony: Indicates that the principal is "roaming" and authenticates via the means of the line number, a user suffix, and a password element. PersonalTelephony: This class is used to indicate that the principal authenticated via the provision of a fixed-line telephone. AuthenticatedTelephony: Indicates that the principal authenticated via the means of the line number, a user suffix, and a password element. SecureRemotePassword: This class is applicable when the authentication was performed by means of Secure Remote Password. TLSClient: This class indicates that the principal authenticated by means of a client certificate, secured with the SSL/TLS transport. TimeSyncToken: This is applicable when a principal authenticates through a time synchronization token. Unspecified: This indicates that the authentication was performed by unspecified means. Windows: This indicates that Windows integrated authentication is utilized for authentication. Possible values = InternetProtocol, InternetProtocolPassword, Kerberos, MobileOneFactorUnregistered, MobileTwoFactorUnregistered, MobileOneFactorContract, MobileTwoFactorContract, Password, PasswordProtectedTransport, PreviousSession, X509, PGP, SPKI, XMLDSig, Smartcard, SmartcardPKI, SoftwarePKI, Telephony, NomadTelephony, PersonalTelephony, AuthenticatedTelephony, SecureRemotePassword, TLSClient, TimeSyncToken, Unspecified, Windows
samlbinding	<String>	Read-write	This element specifies the transport mechanism of saml messages. Default value: POST Possible values = REDIRECT, POST, ARTIFACT
attributeconsumingserviceindex	<Double>	Read-write	Index/ID of the attribute specification at Identity Provider (IdP). IdP will locate attributes requested by SP using this index and send those attributes in Assertion. Default value: 255 Minimum value = 0 Maximum value = 255
sendthumbprint	<String>	Read-write	Option to send thumbprint instead of x509 certificate in SAML request. Default value: OFF Possible values = ON, OFF
enforceusername	<String>	Read-write	Option to choose whether the username that is extracted from SAML assertion can be edited in login page while doing second factor. Default value: ON Possible values = ON, OFF
logouturl	<String>	Read-write	SingleLogout URL on IdP to which logoutRequest will be sent on Netscaler session cleanup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
artifactresolutionservicedn	<String>	Read-write	post artifact to get actual SAML token.
skewtime	<Double>	Read-write	This option specifies the allowed clock skew in number of minutes that Netscaler ServiceProvider allows on an incoming assertion. For example, if skewTime is 10, then assertion would be valid from (current time - 10) min to (current time + 10) min, ie 20min in all. Default value: 5
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlaction":{
    "name":<String_value>,
    "samlidpcertname":<String_value>,
    "samlsigningcertname":<String_value>,
    "samlredirecturl":<String_value>,
    "samlacsindex":<Double_value>,
    "samluserfield":<String_value>,
    "samlrejectunsignedassertion":<String_value>,
    "samlissuername":<String_value>,
    "samltwofactor":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "requestedauthncontext":<String_value>,
    "authnctxclassref":<String[]_value>,
    "samlbinding":<String_value>,
    "attributeconsumingserviceindex":<Double_value>,
    "sendthumbprint":<String_value>,
    "enforceusername":<String_value>,
    "logouturl":<String_value>,
    "artifactresolutionserviceurl":<String_value>,
    "skewtime":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlaction":{
    "name":<String_value>,
    "samlidpcertname":<String_value>,
    "samlsigningcertname":<String_value>,
    "samlredirecturl":<String_value>,
    "samlacsindex":<Double_value>,
    "samluserfield":<String_value>,
    "samlrejectunsignedassertion":<String_value>,
    "samlissuername":<String_value>,
    "samltwofactor":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "requestedauthncontext":<String_value>,
    "authnctxclassref":<String[]_value>,
    "samlbinding":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attributeconsumingserviceindex":<Double_value>,
    "sendthumbprint":<String_value>,
    "enforceusername":<String_value>,
    "logouturl":<String_value>,
    "artifactresolutionserviceurl":<String_value>,
    "skewtime":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlaction":{
    "name":<String_value>,
    "samlidpcertname":true,
    "samlsigningcertname":true,
    "samlredirecturl":true,
    "samlacsindex":true,
    "samluserfield":true,
    "samlrejectunsignedassertion":true,
    "samlissuername":true,
    "samltwofactor":true,
    "defaultauthenticationgroup":true,
    "attribute1":true,
    "attribute2":true,
    "attribute3":true,
    "attribute4":true,
    "attribute5":true,
    "attribute6":true,
    "attribute7":true,
    "attribute8":true,
    "attribute9":true,
    "attribute10":true,
    "attribute11":true,
    "attribute12":true,
    "attribute13":true,
    "attribute14":true,
    "attribute15":true,
    "attribute16":true,
    "signaturealg":true,
    "digestmethod":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "requestedauthncontext":true,
        "authnctxclassref":true,
        "samlbinding":true,
        "attributeconsumingserviceindex":true,
        "sendthumbprint":true,
        "enforceusername":true,
        "logouturl":true,
        "artifactresolutionserviceurl":true,
        "skewtime":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationsamlaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationsamlaction resources in chunks.

HTTP Method: GET

Request Headers:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlaction": [ {
    "name":<String_value>,
    "samlidpcertname":<String_value>,
    "samlsigningcertname":<String_value>,
    "samlredirecturl":<String_value>,
    "samlacsindex":<Double_value>,
    "samluserfield":<String_value>,
    "samlrejectunsignedassertion":<String_value>,
    "samlissuername":<String_value>,
    "samltwofactor":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "requestedauthncontext":<String_value>,
    "authnctxclassref":<String[]_value>,
    "samlbinding":<String_value>,
    "attributeconsumingserviceindex":<Double_value>,
    "sendthumbprint":<String_value>,
    "enforceusername":<String_value>,
    "logouturl":<String_value>,
    "artifactresolutionserviceurl":<String_value>,
    "skewtime":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlaction": [ {
    "name":<String_value>,
    "samlidpcertname":<String_value>,
    "samlsigningcertname":<String_value>,
    "samlredirecturl":<String_value>,
    "samlacsindex":<Double_value>,
    "samluserfield":<String_value>,
    "samlrejectunsignedassertion":<String_value>,
    "samlissuername":<String_value>,
    "samltwofactor":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "requestedauthncontext":<String_value>,
    "authnctxclassref":<String[]_value>,
    "samlbinding":<String_value>,
    "attributeconsumingserviceindex":<Double_value>,
    "sendthumbprint":<String_value>,
    "enforceusername":<String_value>,
    "logouturl":<String_value>,
    "artifactresolutionserviceurl":<String_value>,
    "skewtime":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlidppolicy

May 20, 2016

Configuration for AAA Saml IdentityProvider (IdP) policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the SAML Identity Provider (IdP) authentication policy. This is used for configuring Netscaler as SAML Identity Provider. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression which is evaluated to choose a profile for authentication. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks. Minimum length = 1
action	<String>	Read-write	Name of the profile to apply to requests or connections that match this policy. Minimum length = 1
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.
comment	<String>	Read-write	Any comments to preserve information about this policy.
logaction	<String>	Read-write	Name of messagelog action to use when a request matches this policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters.
newname	<String>	Read-write	The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my samlidppolicy policy" or 'my samlidppolicy policy'). Minimum length = 1
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlidppolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlidppolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlidppolicy":{
    "name":<String_value>,
    "undefaction":true,
    "comment":true,
    "logaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlidppolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationsamlidppolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy?**pagesize=#no&pageno=#no**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the authenticationsamlidppolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlidppolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "gotopriorityexpression":<String_value>,
    "hits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy/name_value<String>?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlidppolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "gotopriorityexpression":<String_value>,
    "hits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "authenticationsamlidppolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlidppolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to authenticationsamlidppolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the SAML IdentityProvider (IdP) policy for which to display detailed information. Minimum length = 1
authenticationsamlidppolicy_vpnvserver_binding	<authenticationsamlidppolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationsamlidppolicy.
authenticationsamlidppolicy_authenticationvserver_binding	<authenticationsamlidppolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationsamlidppolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationsamlidppolicy_binding": [ {
    "name":<String_value>,
    "authenticationsamlidppolicy_authenticationvserver_binding":<authenticationsamlidppolicy_authenticationvserver_binding[]_value>,
    "authenticationsamlidppolicy_vpnvserver_binding":<authenticationsamlidppolicy_vpnvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlidppolicy_authenticationvserver_binding

Jul 07, 2017

Binding object showing the authenticationvserver that can be bound to authenticationsamlidppolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SAML IdentityProvider (IdP) policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationsamlidppolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationsamlidppolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlidppolicy_authenticationvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationsamlidppolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlidppolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to authenticationsamlidppolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SAML IdentityProvider (IdP) policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationsamlidppolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationsamlidppolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlidppolicy_vpnvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidppolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationsamlidppolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlidpprofile

May 20, 2016

Configuration for AAA Saml IdentityProvider (IdP) profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the new saml single sign-on profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
samlspcertname	<String>	Read-write	Name of the SSL certificate of SAML Relying Party. This certificate is used to verify signature of the incoming AuthnRequest from a Relying Party or Service Provider. Minimum length = 1
samlidpcertname	<String>	Read-write	Name of the signing authority as given in the SAML server's SSL certificate. This certificate is used to sign the SAMLResposne that is sent to Relying Party or Service Provider after successful authentication. Minimum length = 1
assertionconsumerserviceurl	<String>	Read-write	URL to which the assertion is to be sent. Minimum length = 1
sendpassword	<String>	Read-write	Option to send password in assertion. Default value: OFF Possible values = ON, OFF
samlissuername	<String>	Read-write	The name to be used in requests sent from Netscaler to IdP to uniquely identify Netscaler. Minimum length = 1
rejectunsignedrequests	<String>	Read-write	Option to Reject unsigned SAML Requests. Default value: ON Possible values = ON, OFF
signaturealg	<String>	Read-write	Algorithm to be used to sign/verify SAML transactions. Default value: RSA-SHA1 Possible values = RSA-SHA1, RSA-SHA256

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
digestmethod	<String>	Read-write	Default value: SHA1 Possible values = SHA1, SHA256
audience	<String>	Read-write	Audience for which assertion sent by IdP is applicable. This is typically entity name or url that represents ServiceProvider.
nameidformat	<String>	Read-write	Format of Name Identifier sent in Assertion. Default value: transient Possible values = Unspecified, emailAddress, X509SubjectName, WindowsDomainQualifiedName, kerberos, entity, persistent, transient
nameidexpr	<String>	Read-write	Expression that will be evaluated to obtain NameIdentifier to be sent in assertion. Maximum length = 128
attribute1	<String>	Read-write	Name of attribute1 that needs to be sent in SAML Assertion.
attribute1expr	<String>	Read-write	Expression that will be evaluated to obtain attribute1's value to be sent in Assertion. Maximum length = 128
attribute1friendlyname	<String>	Read-write	User-Friendly Name of attribute1 that needs to be sent in SAML Assertion.
attribute1format	<String>	Read-write	Format of Attribute1 to be sent in Assertion. Possible values = URI, Basic
attribute2	<String>	Read-write	Name of attribute2 that needs to be sent in SAML Assertion.
attribute2expr	<String>	Read-write	Expression that will be evaluated to obtain attribute2's value to be sent in Assertion. Maximum length = 128
attribute2friendlyname	<String>	Read-write	User-Friendly Name of attribute2 that needs to be sent in SAML Assertion.
attribute2format	<String>	Read-write	Format of Attribute2 to be sent in Assertion. Possible values = URI, Basic
attribute3	<String>	Read-write	Name of attribute3 that needs to be sent in SAML Assertion.
attribute3expr	<String>	Read-write	Expression that will be evaluated to obtain attribute3's value to be sent in Assertion. Maximum length = 128
attribute3friendlyname	<String>	Read-write	User-Friendly Name of attribute3 that needs to be sent in SAML Assertion.
attribute3format	<String>	Read-write	Format of Attribute3 to be sent in Assertion. Possible values = URI, Basic

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute4expr	<String>	Read-write	Expression that will be evaluated to obtain attribute4's value to be sent in Assertion. Maximum length = 128
attribute4friendlyname	<String>	Read-write	User-Friendly Name of attribute4 that needs to be sent in SAML Assertion.
attribute4format	<String>	Read-write	Format of Attribute4 to be sent in Assertion. Possible values = URI, Basic
attribute5	<String>	Read-write	Name of attribute5 that needs to be sent in SAML Assertion.
attribute5expr	<String>	Read-write	Expression that will be evaluated to obtain attribute5's value to be sent in Assertion. Maximum length = 128
attribute5friendlyname	<String>	Read-write	User-Friendly Name of attribute5 that needs to be sent in SAML Assertion.
attribute5format	<String>	Read-write	Format of Attribute5 to be sent in Assertion. Possible values = URI, Basic
attribute6	<String>	Read-write	Name of attribute6 that needs to be sent in SAML Assertion.
attribute6expr	<String>	Read-write	Expression that will be evaluated to obtain attribute6's value to be sent in Assertion. Maximum length = 128
attribute6friendlyname	<String>	Read-write	User-Friendly Name of attribute6 that needs to be sent in SAML Assertion.
attribute6format	<String>	Read-write	Format of Attribute6 to be sent in Assertion. Possible values = URI, Basic
attribute7	<String>	Read-write	Name of attribute7 that needs to be sent in SAML Assertion.
attribute7expr	<String>	Read-write	Expression that will be evaluated to obtain attribute7's value to be sent in Assertion. Maximum length = 128
attribute7friendlyname	<String>	Read-write	User-Friendly Name of attribute7 that needs to be sent in SAML Assertion.
attribute7format	<String>	Read-write	Format of Attribute7 to be sent in Assertion. Possible values = URI, Basic
attribute8	<String>	Read-write	Name of attribute8 that needs to be sent in SAML Assertion.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute8expr	<String>	Read-write	Assertion. Maximum length = 128
attribute8friendlyname	<String>	Read-write	User-Friendly Name of attribute8 that needs to be sent in SAML Assertion.
attribute8format	<String>	Read-write	Format of Attribute8 to be sent in Assertion. Possible values = URI, Basic
attribute9	<String>	Read-write	Name of attribute9 that needs to be sent in SAML Assertion.
attribute9expr	<String>	Read-write	Expression that will be evaluated to obtain attribute9's value to be sent in Assertion. Maximum length = 128
attribute9friendlyname	<String>	Read-write	User-Friendly Name of attribute9 that needs to be sent in SAML Assertion.
attribute9format	<String>	Read-write	Format of Attribute9 to be sent in Assertion. Possible values = URI, Basic
attribute10	<String>	Read-write	Name of attribute10 that needs to be sent in SAML Assertion.
attribute10expr	<String>	Read-write	Expression that will be evaluated to obtain attribute10's value to be sent in Assertion. Maximum length = 128
attribute10friendlyname	<String>	Read-write	User-Friendly Name of attribute10 that needs to be sent in SAML Assertion.
attribute10format	<String>	Read-write	Format of Attribute10 to be sent in Assertion. Possible values = URI, Basic
attribute11	<String>	Read-write	Name of attribute11 that needs to be sent in SAML Assertion.
attribute11expr	<String>	Read-write	Expression that will be evaluated to obtain attribute11's value to be sent in Assertion. Maximum length = 128
attribute11friendlyname	<String>	Read-write	User-Friendly Name of attribute11 that needs to be sent in SAML Assertion.
attribute11format	<String>	Read-write	Format of Attribute11 to be sent in Assertion. Possible values = URI, Basic
attribute12	<String>	Read-write	Name of attribute12 that needs to be sent in SAML Assertion.
attribute12expr	<String>	Read-write	Expression that will be evaluated to obtain attribute12's value to be sent in Assertion. Maximum length = 128

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute12friendlyname	<String>	Read-write	Assertion.
attribute12format	<String>	Read-write	Format of Attribute12 to be sent in Assertion. Possible values = URI, Basic
attribute13	<String>	Read-write	Name of attribute13 that needs to be sent in SAML Assertion.
attribute13expr	<String>	Read-write	Expression that will be evaluated to obtain attribute13's value to be sent in Assertion. Maximum length = 128
attribute13friendlyname	<String>	Read-write	User-Friendly Name of attribute13 that needs to be sent in SAML Assertion.
attribute13format	<String>	Read-write	Format of Attribute13 to be sent in Assertion. Possible values = URI, Basic
attribute14	<String>	Read-write	Name of attribute14 that needs to be sent in SAML Assertion.
attribute14expr	<String>	Read-write	Expression that will be evaluated to obtain attribute14's value to be sent in Assertion. Maximum length = 128
attribute14friendlyname	<String>	Read-write	User-Friendly Name of attribute14 that needs to be sent in SAML Assertion.
attribute14format	<String>	Read-write	Format of Attribute14 to be sent in Assertion. Possible values = URI, Basic
attribute15	<String>	Read-write	Name of attribute15 that needs to be sent in SAML Assertion.
attribute15expr	<String>	Read-write	Expression that will be evaluated to obtain attribute15's value to be sent in Assertion. Maximum length = 128
attribute15friendlyname	<String>	Read-write	User-Friendly Name of attribute15 that needs to be sent in SAML Assertion.
attribute15format	<String>	Read-write	Format of Attribute15 to be sent in Assertion. Possible values = URI, Basic
attribute16	<String>	Read-write	Name of attribute16 that needs to be sent in SAML Assertion.
attribute16expr	<String>	Read-write	Expression that will be evaluated to obtain attribute16's value to be sent in Assertion. Maximum length = 128
attribute16friendlyname	<String>	Read-write	User-Friendly Name of attribute16 that needs to be sent in SAML Assertion.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute16format	<String>	Read-write	Possible values = URI, Basic
encryptassertion	<String>	Read-write	Option to encrypt assertion when Netscaler IDP sends one. Default value: OFF Possible values = ON, OFF
encryptionalgorithm	<String>	Read-write	Algorithm to be used to encrypt SAML assertion. Default value: AES256 Possible values = DES3, AES128, AES192, AES256
samlbinding	<String>	Read-write	This element specifies the transport mechanism of saml messages. Default value: POST Possible values = REDIRECT, POST, ARTIFACT
skewtime	<Double>	Read-write	This option specifies the number of minutes on either side of current time that the assertion would be valid. For example, if skewTime is 10, then assertion would be valid from (current time - 10) min to (current time + 10) min, ie 20min in all. Default value: 5
serviceproviderid	<String>	Read-write	Unique identifier of the Service Provider that sends SAML Request. Netscaler will ensure that the Issuer of the SAML Request matches this URI. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlidpprofile":{
    "name":<String_value>,
    "samlspcertname":<String_value>,
    "samlidpcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "rejectunsignedrequests":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute1expr":<String_value>,
    "attribute1friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2":<String_value>,
    "attribute2expr":<String_value>,
    "attribute2friendlyname":<String_value>,
    "attribute2format":<String_value>,
    "attribute3":<String_value>,
    "attribute3expr":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute3format":<String_value>,
    "attribute4":<String_value>,
    "attribute4expr":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute4format":<String_value>,
    "attribute5":<String_value>,
    "attribute5expr":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute5format":<String_value>,
    "attribute6":<String_value>,
    "attribute6expr":<String_value>,
    "attribute6friendlyname":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute6format":<String_value>,
    "attribute7":<String_value>,
    "attribute7expr":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute7format":<String_value>,
    "attribute8":<String_value>,
    "attribute8expr":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute8format":<String_value>,
    "attribute9":<String_value>,
    "attribute9expr":<String_value>,
    "attribute9friendlyname":<String_value>,
    "attribute9format":<String_value>,
    "attribute10":<String_value>,
    "attribute10expr":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute10format":<String_value>,
    "attribute11":<String_value>,
    "attribute11expr":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute11format":<String_value>,
    "attribute12":<String_value>,
    "attribute12expr":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute12format":<String_value>,
    "attribute13":<String_value>,
    "attribute13expr":<String_value>,
    "attribute13friendlyname":<String_value>,
    "attribute13format":<String_value>,
    "attribute14":<String_value>,
    "attribute14expr":<String_value>,
    "attribute14friendlyname":<String_value>,
    "attribute14format":<String_value>,
    "attribute15":<String_value>,
    "attribute15expr":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute15format":<String_value>,
    "attribute16":<String_value>,
    "attribute16expr":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute16format":<String_value>,
    "encryptassertion":<String_value>,
    "encryptionalgorithm":<String_value>,
    "samlbinding":<String_value>,
    "skewtime":<Double_value>,
    "serviceproviderid":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlidpprofile":{
    "name":<String_value>,
    "samlspcertname":<String_value>,
    "samlidpcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "rejectunsignedrequests":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute1expr":<String_value>,
    "attribute1friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2":<String_value>,
    "attribute2expr":<String_value>,
    "attribute2friendlyname":<String_value>,
    "attribute2format":<String_value>,
    "attribute3":<String_value>,
    "attribute3expr":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute3format":<String_value>,
    "attribute4":<String_value>,
    "attribute4expr":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute4format":<String_value>,
    "attribute5":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute5expr":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute5format":<String_value>,
    "attribute6":<String_value>,
    "attribute6expr":<String_value>,
    "attribute6friendlyname":<String_value>,
    "attribute6format":<String_value>,
    "attribute7":<String_value>,
    "attribute7expr":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute7format":<String_value>,
    "attribute8":<String_value>,
    "attribute8expr":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute8format":<String_value>,
    "attribute9":<String_value>,
    "attribute9expr":<String_value>,
    "attribute9friendlyname":<String_value>,
    "attribute9format":<String_value>,
    "attribute10":<String_value>,
    "attribute10expr":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute10format":<String_value>,
    "attribute11":<String_value>,
    "attribute11expr":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute11format":<String_value>,
    "attribute12":<String_value>,
    "attribute12expr":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute12format":<String_value>,
    "attribute13":<String_value>,
    "attribute13expr":<String_value>,
    "attribute13friendlyname":<String_value>,
    "attribute13format":<String_value>,
    "attribute14":<String_value>,
    "attribute14expr":<String_value>,
    "attribute14friendlyname":<String_value>,
    "attribute14format":<String_value>,
    "attribute15":<String_value>,
    "attribute15expr":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute15format":<String_value>,
    "attribute16":<String_value>,
    "attribute16expr":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute16format":<String_value>,
    "encryptassertion":<String_value>,
    "encryptionalgorithm":<String_value>,
    "samlbinding":<String_value>,
    "skewtime":<Double_value>,
    "serviceproviderid":<String_value>
}}
```


Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlidpprofile":{
    "name":<String_value>,
    "samlspcertname":true,
    "samlidpcertname":true,
    "assertionconsumerserviceurl":true,
    "sendpassword":true,
    "samlissuername":true,
    "rejectunsignedrequests":true,
    "signaturealg":true,
    "digestmethod":true,
    "audience":true,
    "nameidformat":true,
    "nameidexpr":true,
    "attribute1":true,
    "attribute1friendlyname":true,
    "attribute1format":true,
    "attribute2":true,
    "attribute2friendlyname":true,
    "attribute2format":true,
    "attribute3":true,
    "attribute3friendlyname":true,
    "attribute3format":true,
    "attribute4":true,
    "attribute4friendlyname":true,
    "attribute4format":true,
    "attribute5":true,
    "attribute5friendlyname":true,
    "attribute5format":true,
    "attribute6":true,
    "attribute6friendlyname":true,
    "attribute6format":true,
    "attribute7":true,
    "attribute7friendlyname":true,
    "attribute7format":true,
    "attribute8":true,
    "attribute8friendlyname":true,
    "attribute8format":true,
    "attribute9":true,
    "attribute9friendlyname":true,
    "attribute9format":true,
    "attribute10":true,
    "attribute10friendlyname":true,
    "attribute10format":true,
    "attribute11":true,
    "attribute11friendlyname":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "attribute11format":true,
        "attribute12":true,
        "attribute12friendlyname":true,
        "attribute12format":true,
        "attribute13":true,
        "attribute13friendlyname":true,
        "attribute13format":true,
        "attribute14":true,
        "attribute14friendlyname":true,
        "attribute14format":true,
        "attribute15":true,
        "attribute15friendlyname":true,
        "attribute15format":true,
        "attribute16":true,
        "attribute16friendlyname":true,
        "attribute16format":true,
        "encryptassertion":true,
        "encryptionalgorithm":true,
        "samlbinding":true,
        "skewtime":true,
        "serviceproviderid":true
    }}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationsamlidpprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationsamlidpprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlidpprofile": [ {
    "name":<String_value>,
    "samlspcertname":<String_value>,
    "samlidpcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "rejectunsignedrequests":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>,
    "attribute1friendlyname":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute2friendlyname":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute6friendlyname":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute9friendlyname":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute13friendlyname":<String_value>,
    "attribute14friendlyname":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2format":<String_value>,
    "attribute3format":<String_value>,
    "attribute4format":<String_value>,
    "attribute5format":<String_value>,
    "attribute6format":<String_value>,
    "attribute7format":<String_value>,
    "attribute8format":<String_value>,
    "attribute9format":<String_value>,
    "attribute10format":<String_value>,
    "attribute11format":<String_value>,
    "attribute12format":<String_value>,
    "attribute13format":<String_value>,
    "attribute14format":<String_value>,
    "attribute15format":<String_value>,
    "attribute16format":<String_value>,
    "attribute1expr":<String_value>,
    "attribute2expr":<String_value>,
    "attribute3expr":<String_value>,
    "attribute4expr":<String_value>,
    "attribute5expr":<String_value>,
    "attribute6expr":<String_value>,
    "attribute7expr":<String_value>,
    "attribute8expr":<String_value>,
    "attribute9expr":<String_value>,
    "attribute10expr":<String_value>,
    "attribute11expr":<String_value>,
    "attribute12expr":<String_value>,
    "attribute13expr":<String_value>,
    "attribute14expr":<String_value>,
    "attribute15expr":<String_value>,
    "attribute16expr":<String_value>,
    "encryptassertion":<String_value>,
    "encryptionalgorithm":<String_value>,
    "samlbinding":<String_value>,
    "skewtime":<Double_value>,
    "serviceproviderid":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlidpprofile": [ {
    "name":<String_value>,
    "samlspcertname":<String_value>,
    "samlidpcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "rejectunsignedrequests":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
"attribute7":<String_value>,
"attribute8":<String_value>,
"attribute9":<String_value>,
"attribute10":<String_value>,
"attribute11":<String_value>,
"attribute12":<String_value>,
"attribute13":<String_value>,
"attribute14":<String_value>,
"attribute15":<String_value>,
"attribute16":<String_value>,
"attribute1friendlyname":<String_value>,
"attribute2friendlyname":<String_value>,
"attribute3friendlyname":<String_value>,
"attribute4friendlyname":<String_value>,
"attribute5friendlyname":<String_value>,
"attribute6friendlyname":<String_value>,
"attribute7friendlyname":<String_value>,
"attribute8friendlyname":<String_value>,
"attribute9friendlyname":<String_value>,
"attribute10friendlyname":<String_value>,
"attribute11friendlyname":<String_value>,
"attribute12friendlyname":<String_value>,
"attribute13friendlyname":<String_value>,
"attribute14friendlyname":<String_value>,
"attribute15friendlyname":<String_value>,
"attribute16friendlyname":<String_value>,
"attribute1format":<String_value>,
"attribute2format":<String_value>,
"attribute3format":<String_value>,
"attribute4format":<String_value>,
"attribute5format":<String_value>,
"attribute6format":<String_value>,
"attribute7format":<String_value>,
"attribute8format":<String_value>,
"attribute9format":<String_value>,
"attribute10format":<String_value>,
"attribute11format":<String_value>,
"attribute12format":<String_value>,
"attribute13format":<String_value>,
"attribute14format":<String_value>,
"attribute15format":<String_value>,
"attribute16format":<String_value>,
"attribute1expr":<String_value>,
"attribute2expr":<String_value>,
"attribute3expr":<String_value>,
"attribute4expr":<String_value>,
"attribute5expr":<String_value>,
"attribute6expr":<String_value>,
"attribute7expr":<String_value>,
"attribute8expr":<String_value>,
"attribute9expr":<String_value>,
"attribute10expr":<String_value>,
"attribute11expr":<String_value>,
"attribute12expr":<String_value>,
"attribute13expr":<String_value>,
"attribute14expr":<String_value>,
"attribute15expr":<String_value>,
"attribute16expr":<String_value>,
"encryptassertion":<String_value>,
"encryptionalgorithm":<String_value>,
"samlbinding":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "skewtime":<Double_value>,
    "serviceproviderid":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlidpprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlidpprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlpolicy

Jul 07, 2017

Configuration for AAA Saml policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the SAML policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after SAML policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to determine whether to attempt to authenticate the user with the SAML server. Minimum length = 1
reqaction	<String>	Read-write	Name of the SAML authentication action to be performed if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationsamlpolicy":{
    "name":<String_value>,
    "rule":true,
    "reqaction":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationsamlpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationsamlpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "reqaction":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationtacacsaction

May 20, 2016

Configuration for TACACS action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the TACACS+ profile (action). Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after TACACS profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'y authentication action'). Minimum length = 1
serverip	<String>	Read-write	IP address assigned to the TACACS+ server. Minimum length = 1
serverport	<Integer>	Read-write	Port number on which the TACACS+ server listens for connections. Default value: 49 Minimum value = 1
authtimeout	<Double>	Read-write	Number of seconds the NetScaler appliance waits for a response from the TACACS+ server. Default value: 3 Minimum value = 1
tacacssecret	<String>	Read-write	Key shared between the TACACS+ server and the NetScaler appliance. Required for allowing the NetScaler appliance to communicate with the TACACS+ server. Minimum length = 1
authorization	<String>	Read-write	Use streaming authorization on the TACACS+ server. Possible values = ON, OFF
accounting	<String>	Read-write	Whether the TACACS+ server is currently accepting accounting messages. Possible values = ON, OFF
auditfailedcmds	<String>	Read-write	The state of the TACACS+ server that will receive accounting messages. Possible values = ON, OFF
			This is the default group that is chosen when the authentication succeeds in

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
failure	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"authenticationtacacsaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "tacacssecret":<String_value>,
    "authorization":<String_value>,
    "accounting":<String_value>,
    "auditfailedcmds":<String_value>,
    "defaultauthenticationgroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"authenticationtacacsaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "tacacssecret":<String_value>,
    "authorization":<String_value>,
    "accounting":<String_value>,
    "auditfailedcmds":<String_value>,
    "defaultauthenticationgroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationtacacsaction":{
    "name":<String_value>,
    "serverip":true,
    "serverport":true,
    "authtimeout":true,
    "tacacssecret":true,
    "authorization":true,
    "accounting":true,
    "auditfailedcmds":true,
    "defaultauthenticationgroup":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationtacacsaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationtacacsaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationtacacsaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "serverport":<Integer_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "authtimeout":<Double_value>,
    "tacacssecret":<String_value>,
    "authorization":<String_value>,
    "accounting":<String_value>,
    "auditfailedcmds":<String_value>,
    "success":<Double_value>,
    "failure":<Double_value>,
    "defaultauthenticationgroup":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationtacacsaction": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "authtimeout":<Double_value>,
    "tacacssecret":<String_value>,
    "authorization":<String_value>,
    "accounting":<String_value>,
    "auditfailedcmds":<String_value>,
    "success":<Double_value>,
    "failure":<Double_value>,
    "defaultauthenticationgroup":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacsaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationtacacsaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlpolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to authenticationsamlpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SAML policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationsamlpolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationsamlpolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlpolicy_authenticationvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationsamlpolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlpolicy_binding

May 20, 2016

Binding object which returns the resources bound to authenticationsamlpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the SAML policy. Minimum length = 1
authenticationsamlpolicy_vpnglobal_binding	<authenticationsamlpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationsamlpolicy.
authenticationsamlpolicy_vpnvserver_binding	<authenticationsamlpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationsamlpolicy.
authenticationsamlpolicy_authenticationvserver_binding	<authenticationsamlpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationsamlpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationsamlpolicy_binding": [ {
    "name":<String_value>,
    "authenticationsamlpolicy_vpnvserver_binding":<authenticationsamlpolicy_vpnvserver_binding[]_value>,
    "authenticationsamlpolicy_vpnglobal_binding":<authenticationsamlpolicy_vpnglobal_binding[]_value>,
    "authenticationsamlpolicy_authenticationvserver_binding":<authenticationsamlpolicy_authenticationvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlpolicy_vpnglobal_binding

Jul 07, 2017

Binding object showing the vpnglobal that can be bound to authenticationsamlpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SAML policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationsamlpolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationsamlpolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlpolicy_vpnglobal_binding": [ {
      "boundto":<String_value>,
      "name":<String_value>,
      "priority":<Double_value>,
      "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationsamlpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlpolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to authenticationsamlpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the SAML policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationsamlpolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationsamlpolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlpolicy_vpnvserver_binding": [ {
      "boundto":<String_value>,
      "name":<String_value>,
      "priority":<Double_value>,
      "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationsamlpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationsamlpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationtacacspolicy

May 20, 2016

Configuration for TACACS policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the TACACS+ policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after TACACS+ policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to determine whether to attempt to authenticate the user with the TACACS+ server. Minimum length = 1
reqaction	<String>	Read-write	Name of the TACACS+ action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationtacacspolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationtacacspolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationtacacspolicy":{
    "name":<String_value>,
    "rule":true,
    "reqaction":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationtacacspolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationtacacspolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationtacacspolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "reqaction":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationtacacspolicy": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationtacacspolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationtacacspolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to authenticationtacacspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the TACACS+ policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationtacacspolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationtacacspolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationtacacspolicy_authenticationvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationtacacspolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationtacacspolicy_binding

May 20, 2016

Binding object which returns the resources bound to authenticationtacacspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the TACACS+ policy. Minimum length = 1
authenticationtacacspolicy_vpnglobal_binding	<authenticationtacacspolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationtacacspolicy.
authenticationtacacspolicy_vpnvserver_binding	<authenticationtacacspolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationtacacspolicy.
authenticationtacacspolicy_systemglobal_binding	<authenticationtacacspolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationtacacspolicy.
authenticationtacacspolicy_authenticationvserver_binding	<authenticationtacacspolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationtacacspolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationtacacspolicy_binding": [ {
    "name":<String_value>,
    "authenticationtacacspolicy_vpnvserver_binding":<authenticationtacacspolicy_vpnvserver_binding[]_value>,
    "authenticationtacacspolicy_systemglobal_binding":<authenticationtacacspolicy_systemglobal_binding[]_value>,
    "authenticationtacacspolicy_authenticationvserver_binding":<authenticationtacacspolicy_authenticationvserver_binding[]_value>,
    "authenticationtacacspolicy_vpnglobal_binding":<authenticationtacacspolicy_vpnglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationtacacspolicy_systemglobal_binding

May 20, 2016

Binding object showing the systemglobal that can be bound to authenticationtacacspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the TACACS+ policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_systemglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_systemglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationtacacspolicy_systemglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_systemglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationtacacspolicy_systemglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationtacacspolicy_systemglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_systemglobal_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationtacacspolicy_systemglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver

May 20, 2016

Configuration for authentication virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the new authentication virtual server. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the authentication virtual server is added by using the rename authentication vserver command. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
servicetype	<String>	Read-write	Protocol type of the authentication virtual server. Always SSL. Default value: SSL Possible values = SSL
ipv46	<String>	Read-write	IP address of the authentication virtual server, if a single IP address is assigned to the virtual server. Minimum length = 1
range	<Double>	Read-write	If you are creating a series of virtual servers with a range of IP addresses assigned to them, the length of the range. The new range of authentication virtual servers will have IP addresses consecutively numbered, starting with the primary address specified with the IP Address parameter. Default value: 1 Minimum value = 1
port	<Integer>	Read-write	TCP port on which the virtual server accepts connections. Range 1 - 65535
state	<String>	Read-write	Initial state of the new virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
authentication	<String>	Read-write	Require users to be authenticated before sending traffic through this virtual server. Default value: ON

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
authenticationdomain	<String>	Read-write	The domain of the authentication cookie set by Authentication vserver. Minimum length = 3 Maximum length = 252
comment	<String>	Read-write	Any comments associated with this virtual server.
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
appflowlog	<String>	Read-write	Log AppFlow flow information. Default value: ENABLED Possible values = ENABLED, DISABLED
maxloginattempts	<Double>	Read-write	Maximum Number of login Attempts. Minimum value = 1 Maximum value = 255
failedlogintimeout	<Double>	Read-write	Number of minutes an account will be locked if user exceeds maximum permissible attempts. Minimum value = 1
newname	<String>	Read-write	New name of the authentication virtual server. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, 'my authentication policy' or "my authentication policy"). Minimum length = 1
ip	<String>	Read-only	The Virtual IP address of the authentication vserver.
value	<String>	Read-only	Indicates whether or not the certificate is bound or if SSL offload is disabled. Possible values = Certkey not bound, SSL feature disabled
type	<String>	Read-only	The type of Virtual Server, e.g. CONTENT based or ADDRESS based. Possible values = CONTENT, ADDRESS
curstate	<String>	Read-only	The current state of the Virtual server, e.g. UP, DOWN, BUSY, etc. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
status	<Integer>	Read-only	Whether or not this vserver responds to ARPs and whether or not round-robin selection is temporarily in effect.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cachetype	<String>	Read-only	FORWARD. Possible values = TRANSPARENT, REVERSE, FORWARD
redirect	<String>	Read-only	The cache redirect policy. The valid redirect policies are: l. CACHE - Directs all requests to the cache. 2. POLICY - Applies cache redirection policy to determine whether the request should be directed to the cache or origin. This is the default setting. 3. ORIGIN - Directs all requests to the origin server. Possible values = CACHE, POLICY, ORIGIN
precedence	<String>	Read-only	This argument is used only when configuring content switching on the specified virtual server. This is applicable only if both the URL and RULE-based policies have been configured on the same virtual server. It specifies the type of policy (URL or RULE) that takes precedence on the content switching virtual server. The default setting is RULE. l URL - In this case, the incoming request is matched against the URL-based policies before the rule-based policies. l RULE - In this case, the incoming request is matched against the rule-based policies before the URL-based policies. For all URL-based policies, the precedence hierarchy is: 1. Domain and exact URL 2. Domain, prefix and suffix 3. Domain and suffix 4. Domain and prefix 5. Domain only 6. Exact URL 7. Prefix and suffix 8. Suffix only 9. Prefix only 10. Default. Possible values = RULE, URL
redirecturl	<String>	Read-only	The URL where traffic is redirected if the virtual server in system becomes unavailable. WARNING! Make sure that the domain you specify in the URL does not match the domain specified in the -d domainName argument of the ###add cs policy### command. If the same domain is specified in both arguments, the request will be continuously redirected to the same unavailable virtual server in the system. If so, the user may not get the requested content.
curaaausers	<Double>	Read-only	The number of current users logged in to this vserver.
rule	<String>	Read-only	The name of the rule, or expression, if any, that policy for the authentication server is to use. Rules are combinations of Expressions. Expressions are simple conditions, such as a test for equality, applied to operands, such as a URL string or an IP address. Expression syntax is described in the Installation and Configuration Guide. The default rule is ns_true.
policy	<String>	Read-only	The name of the policy, if any, bound to the authentication vserver.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
weight	<Double>	Read-only	Weight for this service, if any. This weight is used when the system performs load balancing, giving greater priority to a specific service. It is useful when the services bound to a virtual server are of different capacity.
cachevserver	<String>	Read-only	The name of the default target cache virtual server, if any, to which requests are redirected.
backupvserver	<String>	Read-only	The name of the backup vpn virtual server for this vpn virtual server.
clttimeout	<Double>	Read-only	The idle time, if any, in seconds after which the client connection is terminated.
somethod	<String>	Read-only	VPN client applications are allocated from a block of Intranet IP addresses. That block may be exhausted after a certain number of connections. This switch specifies the method used to determine whether or not a new connection will spillover, or exhaust, the allocated block of Intranet IP addresses for that application. Possible values are CONNECTION or DYNAMICCONNECTION. CONNECTION means that a static integer value is the hard limit for the spillover threshold. The spillover threshold is described below. DYNAMICCONNECTION means that the spillover threshold is set according to the maximum number of connections defined for the vpn vserver. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sothreshold	<Double>	Read-only	VPN client applications are allocated from a block of Intranet IP addresses. That block may be exhausted after a certain number of connections. The value of this option is number of client connections after which the Mapped IP address is used as the client source IP address instead of an address from the allocated block of Intranet IP addresses.
sopersistence	<String>	Read-only	Whether or not cookie-based site persistance is enabled for this VPN vserver. Possible values are 'ConnectionProxy', HTTPRedirect, or NONE. Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-only	The timeout, if any, for cookie-based site persistance of this VPN vserver.
priority	<Double>	Read-only	The priority, if any, of the vpn vserver policy.
downstateflush	<String>	Read-only	Perform delayed clean up of connections on this vserver. Possible values = ENABLED, DISABLED
disableprimaryondown	<String>	Read-only	Tells whether traffic will continue reaching backup vservers even after primary comes UP from DOWN state. Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
listenpriority	<Double>	Read-only	Priority of listen policy for authentication vserver.
tcpprofilename	<String>	Read-only	The name of the TCP profile.
httpprofilename	<String>	Read-only	Name of the HTTP profile.
vstype	<Double>	Read-only	Virtual Server Type, e.g. Load Balancing, Content Switch, Cache Redirection.
ngname	<String>	Read-only	Nodegroup devno to which this authentication vsever belongs to.
secondary	<Boolean>	Read-only	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
groupextraction	<Boolean>	Read-only	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationvserver":{
    "name":<String_value>,
    "servicetype":<String_value>,
    "ipv46":<String_value>,
    "range":<Double_value>,
    "port":<Integer_value>,
    "state":<String_value>,
    "authentication":<String_value>,
    "authenticationdomain":<String_value>,
    "comment":<String_value>,
    "td":<Double_value>,
    "appflowlog":<String_value>,
    "maxloginattempts":<Double_value>,
    "failedlogintimeout":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationvserver":{
    "name":<String_value>,
    "ipv46":<String_value>,
    "authentication":<String_value>,
    "authenticationdomain":<String_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "maxloginattempts":<Double_value>,
    "failedlogintimeout":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"authenticationvserver":{
    "name":<String_value>,
    "authenticationdomain":true,
    "maxloginattempts":true,
    "authentication":true,
    "comment":true,
    "appflowlog":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"authenticationvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationvserver":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationvserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver": [ {
    "name":<String_value>,
    "ip":<String_value>,
    "td":<Double_value>,
    "ipv46":<String_value>,
    "value":<String_value>,
    "port":<Integer_value>,
    "range":<Double_value>,
    "servicetype":<String_value>,
    "type":<String_value>,
    "curstate":<String_value>,
    "status":<Integer_value>,
    "cachetype":<String_value>,
    "redirect":<String_value>,
    "precedence":<String_value>,
    "redirecturl":<String_value>,
    "authentication":<String_value>,
    "curaaausers":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "authenticationdomain":<String_value>,
    "rule":<String_value>,
    "policyname":<String_value>,
    "policy":<String_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "cachevserver":<String_value>,
    "backupvserver":<String_value>,
    "clttimeout":<Double_value>,
    "somethod":<String_value>,
    "sothreshold":<Double_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "priority":<Double_value>,
    "downstateflush":<String_value>,
    "disableprimaryondown":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "vstype":<Double_value>,
    "ngname":<String_value>,
    "maxloginattempts":<Double_value>,
    "failedlogintimeout":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver": [ {
    "name":<String_value>,
    "ip":<String_value>,
    "td":<Double_value>,
    "ipv46":<String_value>,
    "value":<String_value>,
    "port":<Integer_value>,
    "range":<Double_value>,
    "servicetype":<String_value>,
    "type":<String_value>,
    "curstate":<String_value>,
    "status":<Integer_value>,
    "cachetype":<String_value>,
    "redirect":<String_value>,
    "precedence":<String_value>,
    "redirecturl":<String_value>,
    "authentication":<String_value>,
    "curaaausers":<Double_value>,
    "authenticationdomain":<String_value>,
    "rule":<String_value>,
    "policyname":<String_value>,
    "policy":<String_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "cachevserver":<String_value>,
    "backupvserver":<String_value>,
    "clttimeout":<Double_value>,
    "somethod":<String_value>,
    "sothreshold":<Double_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "priority":<Double_value>,
    "downstateflush":<String_value>,
    "disableprimaryondown":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "vstype":<Double_value>,
    "ngname":<String_value>,
    "maxloginattempts":<Double_value>,
    "failedlogintimeout":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationtacacspolicy_vpnglobal_binding

May 20, 2016

Binding object showing the vpnglobal that can be bound to authenticationtacacspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the TACACS+ policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationtacacspolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationtacacspolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationtacacspolicy_vpnglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationtacacspolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationtacacspolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to authenticationtacacspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the TACACS+ policy. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationtacacspolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationtacacspolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationtacacspolicy_vpnvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationtacacspolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationtacacspolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_auditnslogpolicy_binding

Jul 07, 2017

Binding object showing the auditnslogpolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_auditnslogpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_auditnslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_auditnslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_auditnslogpolicy_binding": [ {
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_auditnslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_auditsyslogpolicy_binding

May 20, 2016

Binding object showing the auditsyslogpolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_auditsyslogpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_auditsyslogpolicy_binding": [ {
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_auditsyslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationcertpolicy_binding

May 20, 2016

Binding object showing the authenticationcertpolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationcertpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationcertpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationcertpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_authenticationcertpolicy_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationcertpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_authenticationcertpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationldappolicy_binding

May 20, 2016

Binding object showing the authenticationldappolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationldappolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationldappolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationldappolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "authenticationvserver_authenticationldappolicy_binding": [ {
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationldappolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_authenticationldappolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationlocalpolicy_binding

May 20, 2016

Binding object showing the authenticationlocalpolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationlocalpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationlocalpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationlocalpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_authenticationlocalpolicy_binding": [ {
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationlocalpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_authenticationlocalpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_binding

May 20, 2016

Binding object which returns the resources bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the authentication virtual server. Minimum length = 1
authenticationvserver_auditnslogpolicy_binding	<authenticationvserver_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationpolicy_binding	<authenticationvserver_authenticationpolicy_binding[]>	Read-only	authenticationpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationradiuspolicy_binding	<authenticationvserver_authenticationradiuspolicy_binding[]>	Read-only	authenticationradiuspolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationldappolicy_binding	<authenticationvserver_authenticationldappolicy_binding[]>	Read-only	authenticationldappolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationsamlidppolicy_binding	<authenticationvserver_authenticationsamlidppolicy_binding[]>	Read-only	authenticationsamlidppolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationwebauthpolicy_binding	<authenticationvserver_authenticationwebauthpolicy_binding[]>	Read-only	authenticationwebauthpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationlocalpolicy_binding	<authenticationvserver_authenticationlocalpolicy_binding[]>	Read-only	authenticationlocalpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationnegotiatepolicy_binding	<authenticationvserver_authenticationnegotiatepolicy_binding[]>	Read-only	authenticationnegotiatepolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationtacacspolicy_binding	<authenticationvserver_authenticationtacacspolicy_binding[]>	Read-only	authenticationtacacspolicy that can be bound to authenticationvserver.
authenticationvserver_vpnportaltheme_binding	<authenticationvserver_vpnportaltheme_binding[]>	Read-only	vpnportaltheme that can be bound to authenticationvserver.
authenticationvserver_authenticationsamlpolicy_binding	<authenticationvserver_authenticationsamlpolicy_binding[]>	Read-only	authenticationsamlpolicy that can be bound to authenticationvserver.
authenticationvserver_cspolicy_binding	<authenticationvserver_cspolicy_binding[]>	Read-only	cspolicy that can be bound to authenticationvserver.
authenticationvserver_auditsyslogpolicy_binding	<authenticationvserver_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationloginschemapolicy_binding	<authenticationvserver_authenticationloginschemapolicy_binding[]>	Read-only	authenticationloginschemapolicy that can be bound to authenticationvserver.
authenticationvserver_authenticationcertpolicy_binding	<authenticationvserver_authenticationcertpolicy_binding[]>	Read-only	authenticationcertpolicy that can be bound to authenticationvserver.
authenticationvserver_tmsessionpolicy_binding	<authenticationvserver_tmsessionpolicy_binding[]>	Read-only	tmsessionpolicy that can be bound to authenticationvserver.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationvserver_binding": [ {
    "name":<String_value>,
    "authenticationvserver_authenticationlocalpolicy_binding":<authenticationvserver_authenticationlocalpolicy_binding[]_value>,
    "authenticationvserver_tmsessionpolicy_binding":<authenticationvserver_tmsessionpolicy_binding[]_value>,
    "authenticationvserver_cspolicy_binding":<authenticationvserver_cspolicy_binding[]_value>,
    "authenticationvserver_auditsyslogpolicy_binding":<authenticationvserver_auditsyslogpolicy_binding[]_value>,
    "authenticationvserver_authenticationsamlidppolicy_binding":<authenticationvserver_authenticationsamlidppolicy_binding[]_value>,
    "authenticationvserver_auditnslogpolicy_binding":<authenticationvserver_auditnslogpolicy_binding[]_value>,
    "authenticationvserver_authenticationldappolicy_binding":<authenticationvserver_authenticationldappolicy_binding[]_value>,
    "authenticationvserver_authenticationcertpolicy_binding":<authenticationvserver_authenticationcertpolicy_binding[]_value>,
    "authenticationvserver_authenticationsamlpolicy_binding":<authenticationvserver_authenticationsamlpolicy_binding[]_value>,
    "authenticationvserver_authenticationpolicy_binding":<authenticationvserver_authenticationpolicy_binding[]_value>,
    "authenticationvserver_vpnportaltheme_binding":<authenticationvserver_vpnportaltheme_binding[]_value>,
    "authenticationvserver_authenticationradiuspolicy_binding":<authenticationvserver_authenticationradiuspolicy_binding[]_value>,
    "authenticationvserver_authenticationloginschemapolicy_binding":<authenticationvserver_authenticationloginschemapolicy_binding[]_value>,
    "authenticationvserver_authenticationtacacspolicy_binding":<authenticationvserver_authenticationtacacspolicy_binding[]_value>,
    "authenticationvserver_authenticationwebauthpolicy_binding":<authenticationvserver_authenticationwebauthpolicy_binding[]_value>,
    "authenticationvserver_authenticationnegotiatepolicy_binding":<authenticationvserver_authenticationnegotiatepolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationloginschemapolicy_binding

Jul 07, 2017

Binding object showing the authenticationloginschemapolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationloginschemapolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationloginschemapolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationloginschemapolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_authenticationloginschemapolicy_binding": [ {
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationloginschemapolicy_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationvserver_authenticationloginschemapolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationnegotiatepolicy_binding

May 20, 2016

Binding object showing the authenticationnegotiatepolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationnegotiatepolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationnegotiatepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationnegotiatepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_authenticationnegotiatepolicy_binding": [ {
      "priority":<Double_value>,
      "name":<String_value>,
      "nextfactor":<String_value>,
      "gotopriorityexpression":<String_value>,
      "secondary":<Boolean_value>,
      "policy":<String_value>,
      "groupextraction":<Boolean_value>,
      "acttype":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationnegotiatepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationvserver_authenticationnegotiatepolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationpolicy_binding

May 20, 2016

Binding object showing the authenticationpolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	On success invoke label.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_authenticationpolicy_binding": [ {
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "acttype":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationvserver_authenticationpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationradiuspolicy_binding

Jul 07, 2017

Binding object showing the authenticationradiuspolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationradiuspolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationradiuspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationradiuspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_authenticationradiuspolicy_binding": [ {
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "acttype":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationradiuspolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_authenticationradiuspolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationsamlidppolicy_binding

May 20, 2016

Binding object showing the authenticationsamlidppolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationsamlidppolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationsamlidppolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationsamlidppolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_authenticationsamlidppolicy_binding": [ {
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "acttype":<Double_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationsamlidppolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_authenticationsamlidppolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationsamlpolicy_binding

May 20, 2016

Binding object showing the authenticationsamlpolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationsamlpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationsamlpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationsamlpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_authenticationsamlpolicy_binding": [ {
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationsamlpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_authenticationsamlpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationtacacspolicy_binding

Jul 07, 2017

Binding object showing the authenticationtacacspolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationtacacspolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationtacacspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationtacacspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_authenticationtacacspolicy_binding": [ {
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "groupextraction":<Boolean_value>,
    "acttype":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationtacacspolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_authenticationtacacspolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_authenticationwebauthpolicy_binding

May 20, 2016

Binding object showing the authenticationwebauthpolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Bind the authentication policy to the secondary chain. Provides for multifactor authentication in which a user must authenticate via both a primary authentication method and, afterward, via a secondary authentication method. Because user groups are aggregated across authentication systems, usernames must be the same on all authentication servers. Passwords can be different.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_authenticationwebauthpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_authenticationwebauthpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_authenticationwebauthpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_authenticationwebauthpolicy_binding": [ {
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_authenticationwebauthpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_authenticationwebauthpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_cspolicy_binding

May 20, 2016

Binding object showing the cspolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_cspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_cspolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_cspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_cspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "authenticationvserver_cspolicy_binding": [ {
      "priority":<Double_value>,
      "name":<String_value>,
      "nextfactor":<String_value>,
      "gotopriorityexpression":<String_value>,
      "secondary":<Boolean_value>,
      "policy":<String_value>,
      "groupextraction":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_cspolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_cspolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_tmsessionpolicy_binding

May 20, 2016

Binding object showing the tmsessionpolicy that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the vpn vserver policy.
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
nextfactor	<String>	Read-write	Applicable only while binding advance authentication policy as classic authentication policy does not support nFactor.
gotopriorityexpression	<String>	Read-write	Applicable only to advance authentication policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
secondary	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
policy	<String>	Read-write	The name of the policy, if any, bound to the authentication vserver.
groupextraction	<Boolean>	Read-write	Applicable only while bindind classic authentication policy as advance authentication policy use nFactor.
__count	<Double>	Read-write	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_tmsessionpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationvserver_tmsessionpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_tmsessionpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_tmsessionpolicy_binding": [ {
    "priority":<Double_value>,
    "name":<String_value>,
    "nextfactor":<String_value>,
    "gotopriorityexpression":<String_value>,
    "secondary":<Boolean_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_tmsessionpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"authenticationvserver_tmsessionpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver_vpnportaltheme_binding

May 20, 2016

Binding object showing the vpnportaltheme that can be bound to authenticationvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the authentication virtual server to which to bind the policy. Minimum length = 1
portaltheme	<String>	Read-write	Theme for Authentication virtual server Login portal.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authenticationvserver_vpnportaltheme_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authenticationvserver_vpnportaltheme_binding":{
    "name":<String_value>,
    "portaltheme":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of authenticationvserver_vpnportaltheme_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationvserver_vpnportaltheme_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver_vpnportaltheme_binding": [ {
    "name":<String_value>,
    "portaltheme":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationvserver_vpnportaltheme_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationvserver_vpnportaltheme_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationwebauthaction

May 20, 2016

Configuration for Web authentication action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the Web Authentication action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication action" or 'my authentication action'). Minimum length = 1
serverip	<String>	Read-write	IP address of the web server to be used for authentication. Minimum length = 1
serverport	<Integer>	Read-write	Port on which the web server accepts connections. Minimum value = 1 Range 1 - 65535
fullreqexpr	<String>	Read-write	Exact HTTP request, in the form of a default syntax expression, which the NetScaler appliance sends to the authentication server. The NetScaler appliance does not check the validity of this request. One must manually validate the request.
scheme	<String>	Read-write	Type of scheme for the web server. Possible values = http, https
successrule	<String>	Read-write	Expression, that checks to see if authentication is successful.
defaultauthenticationgroup	<String>	Read-write	This is the default group that is chosen when the authentication succeeds in addition to extracted groups.
attribute1	<String>	Read-write	Expression that would be evaluated to extract attribute1 from the webauth response. Maximum length = 128
attribute2	<String>	Read-write	Expression that would be evaluated to extract attribute2 from the webauth response.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute3	<String>	Read-write	Expression that would be evaluated to extract attribute3 from the webauth response. Maximum length = 128
attribute4	<String>	Read-write	Expression that would be evaluated to extract attribute4 from the webauth response. Maximum length = 128
attribute5	<String>	Read-write	Expression that would be evaluated to extract attribute5 from the webauth response. Maximum length = 128
attribute6	<String>	Read-write	Expression that would be evaluated to extract attribute6 from the webauth response. Maximum length = 128
attribute7	<String>	Read-write	Expression that would be evaluated to extract attribute7 from the webauth response. Maximum length = 128
attribute8	<String>	Read-write	Expression that would be evaluated to extract attribute8 from the webauth response. Maximum length = 128
attribute9	<String>	Read-write	Expression that would be evaluated to extract attribute9 from the webauth response. Maximum length = 128
attribute10	<String>	Read-write	Expression that would be evaluated to extract attribute10 from the webauth response. Maximum length = 128
attribute11	<String>	Read-write	Expression that would be evaluated to extract attribute11 from the webauth response. Maximum length = 128
attribute12	<String>	Read-write	Expression that would be evaluated to extract attribute12 from the webauth response. Maximum length = 128
attribute13	<String>	Read-write	Expression that would be evaluated to extract attribute13 from the webauth response. Maximum length = 128
attribute14	<String>	Read-write	Expression that would be evaluated to extract attribute14 from the webauth response. Maximum length = 128
attribute15	<String>	Read-write	Expression that would be evaluated to extract attribute15 from the webauth response. Maximum length = 128
			Expression that would be evaluated to extract attribute16 from the webauth

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum length = 128
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"authenticationwebauthaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "fullreqexpr":<String_value>,
    "scheme":<String_value>,
    "successrule":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationwebauthaction":{
    "name":<String_value>,
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "fullreqexpr":<String_value>,
    "scheme":<String_value>,
    "successrule":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"authenticationwebauthaction":{
    "name":<String_value>,
    "serverip":true,
    "serverport":true,
    "fullreqexpr":true,
    "defaultauthenticationgroup":true,
    "attribute1":true,
    "attribute2":true,
    "attribute3":true,
    "attribute4":true,
    "attribute5":true,
    "attribute6":true,
    "attribute7":true,
    "attribute8":true,
    "attribute9":true,
    "attribute10":true,
    "attribute11":true,
    "attribute12":true,
    "attribute13":true,
    "attribute14":true,
    "attribute15":true,
    "attribute16":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationwebauthaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationwebauthaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationwebauthaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "fullreqexpr":<String_value>,
    "scheme":<String_value>,
    "successrule":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationwebauthaction": [ {
    "name":<String_value>,
    "serverip":<String_value>,
    "serverport":<Integer_value>,
    "fullreqexpr":<String_value>,
    "scheme":<String_value>,
    "successrule":<String_value>,
    "defaultauthenticationgroup":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationwebauthaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationwebauthpolicy

May 20, 2016

Configuration for Web authentication policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the WebAuth policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after LDAP policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my authentication policy" or 'my authentication policy'). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to determine whether to attempt to authenticate the user with the Web server. Minimum length = 1
action	<String>	Read-write	Name of the WebAuth action to perform if the policy matches. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationwebauthpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationwebauthpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationwebauthpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationwebauthpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationwebauthpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationwebauthpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "authenticationwebauthpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationwebauthpolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to authenticationwebauthpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationwebauthpolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationwebauthpolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationwebauthpolicy_authenticationvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationwebauthpolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationwebauthpolicy_binding

May 20, 2016

Binding object which returns the resources bound to authenticationwebauthpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
authenticationwebauthpolicy_vpnglobal_binding	<authenticationwebauthpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to authenticationwebauthpolicy.
authenticationwebauthpolicy_vpnvserver_binding	<authenticationwebauthpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to authenticationwebauthpolicy.
authenticationwebauthpolicy_systemglobal_binding	<authenticationwebauthpolicy_systemglobal_binding[]>	Read-only	systemglobal that can be bound to authenticationwebauthpolicy.
authenticationwebauthpolicy_authenticationvserver_binding	<authenticationwebauthpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to authenticationwebauthpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authenticationwebauthpolicy_binding": [ {
    "name":<String_value>,
    "authenticationwebauthpolicy_vpnglobal_binding":<authenticationwebauthpolicy_vpnglobal_binding[]_value>,
    "authenticationwebauthpolicy_authenticationvserver_binding":<authenticationwebauthpolicy_authenticationvserver_binding[]_value>,
    "authenticationwebauthpolicy_systemglobal_binding":<authenticationwebauthpolicy_systemglobal_binding[]_value>,
    "authenticationwebauthpolicy_vpnvserver_binding":<authenticationwebauthpolicy_vpnvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationauthnprofile

May 20, 2016

Configuration for Authentication profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the authentication profile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the RADIUS action is added. Minimum length = 1
authnvsname	<String>	Read-write	Name of the authentication vserver at which authentication should be done. Minimum length = 1 Maximum length = 128
authenticationhost	<String>	Read-write	Hostname of the authentication vserver to which user must be redirected for authentication. Minimum length = 1 Maximum length = 256
authenticationdomain	<String>	Read-write	Domain for which TM cookie must to be set. If unspecified, cookie will be set for FQDN. Minimum length = 1 Maximum length = 256
authenticationlevel	<Double>	Read-write	Authentication weight or level of the vserver to which this will bound. This is used to order TM vservers based on the protection required. A session that is created by authenticating against TM vserver at given level cannot be used to access TM vserver at a higher level. Minimum value = 0 Maximum value = 255
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationauthnprofile":{
    "name":<String_value>,
    "authnvsname":<String_value>,
    "authenticationhost":<String_value>,
    "authenticationdomain":<String_value>,
    "authenticationlevel":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationauthnprofile":{
    "name":<String_value>,
    "authnvsname":<String_value>,
    "authenticationhost":<String_value>,
    "authenticationdomain":<String_value>,
    "authenticationlevel":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile?**action=unset**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authenticationauthnprofile":{
    "name":<String_value>,
    "authenticationdomain":true,
    "authenticationlevel":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authenticationauthnprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationauthnprofile resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationauthnprofile": [ {
    "name":<String_value>,
    "authnvsname":<String_value>,
    "authenticationhost":<String_value>,
    "authenticationdomain":<String_value>,
    "authenticationlevel":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationauthnprofile": [ {
    "name":<String_value>,
    "authnvsname":<String_value>,
    "authenticationhost":<String_value>,
    "authenticationdomain":<String_value>,
    "authenticationlevel":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationauthnprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationauthnprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationwebauthpolicy_vpnglobal_binding

May 20, 2016

Binding object showing the vpnglobal that can be bound to authenticationwebauthpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationwebauthpolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationwebauthpolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationwebauthpolicy_vpnglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationwebauthpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationwebauthpolicy_vpnvserver_binding

Jul 07, 2017

Binding object showing the vpnvserver that can be bound to authenticationwebauthpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the WebAuth policy. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authenticationwebauthpolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authenticationwebauthpolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationwebauthpolicy_vpnvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authenticationwebauthpolicy_vpnvserver_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authenticationwebauthpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Authorization

May 20, 2016

Authorization configuration. Authorization services check which resources users are authorized to access, and grant permissions accordingly.

Name	Description
authorizationaction	Configuration for authorization action
authorizationpolicy	Configuration for authorization policy
authorizationpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to authorizationpolicy
authorizationpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to authorizationpolicy
authorizationpolicy_authorizationpolicylabel_binding	Binding object showing the authorizationpolicylabel that can be bound to authorizationpolicy
authorizationpolicy_binding	Binding object showing the resources that can be bound to authorizationpolicy
authorizationpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to authorizationpolicy
authorizationpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to authorizationpolicy
authorizationpolicylabel	Configuration for authorization policy label
authorizationpolicylabel_authorizationpolicy_binding	Binding object showing the authorizationpolicy that can be bound to authorizationpolicylabel
authorizationpolicylabel_binding	Binding object showing the resources that can be bound to authorizationpolicylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationaction

May 20, 2016

Configuration for authorization action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of authorization action. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authorizationaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authorizationaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authorizationaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authorizationaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authorizationaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authorizationaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationaction": [ {
    "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authorizationaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authorizationaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationaction": [ {
      "name":<String_value>,

}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationaction?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "authorizationaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicy

May 20, 2016

Configuration for authorization policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the new authorization policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the authorization policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my authorization policy? or ?my authorization policy?). Minimum length = 1
rule	<String>	Read-write	Name of the NetScaler named rule, or a default syntax expression, that the policy uses to perform the authentication.
action	<String>	Read-write	Action to perform if the policy matches: either allow or deny the request. Minimum length = 1
newname	<String>	Read-write	The new name of the author policy. Minimum length = 1
activepolicy	<Double>	Read-only	.
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced) . Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authorizationpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authorizationpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"authorizationpolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authorizationpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authorizationpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "activepolicy":<Double_value>,
      "expressiontype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "activepolicy":<Double_value>,
    "expressiontype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicy": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicylabel

May 20, 2016

Configuration for authorization policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name for the new authorization policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the authorization policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my authorization policy label? or ?authorization policy label?).
newname	<String>	Read-write	The new name of the auth policy label. Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
policyname	<String>	Read-only	Name of the authorization policy to bind to the policy label.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of invocation. Available settings function as follows: * reqvserver - Send the request to the specified request virtual server. * resvserver - Send the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
invoke_labelname	<String>	Read-only	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
flowtype	<Double>	Read-only	Flowtype of the bound authorization policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

description **Name**	<String> **Data Type**	Read-only **Permissions**	Description of the policylabel. **Description**
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"authorizationpolicylabel":{
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"authorizationpolicylabel":{
    "labelname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of authorizationpolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authorizationpolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "authorizationpolicylabel": [ {
      "labelname":<String_value>,
      "numpol":<Double_value>,
      "hits":<Double_value>,
      "policyname":<String_value>,
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "labeltype":<String_value>,
      "invoke_labelname":<String_value>,
      "flowtype":<Double_value>,
      "description":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "authorizationpolicylabel": [ {
    "labelname":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "flowtype":<Double_value>,
    "description":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicylabel": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicylabel_authorizationpolicy_binding

May 20, 2016

Binding object showing the authorizationpolicy that can be bound to authorizationpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the authorization policy to bind to the policy label.
labelname	<String>	Read-write	Name of the authorization policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is set to Policy Label.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then either forward the request or response to the specified virtual server or evaluate the specified policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Send the request to the specified request virtual server. * resvserver - Send the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"authorizationpolicylabel_authorizationpolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicylabel_authorizationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authorizationpolicylabel_authorizationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicylabel_authorizationpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel_authorizationpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authorizationpolicylabel_authorizationpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicylabel_binding

May 20, 2016

Binding object which returns the resources bound to authorizationpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the authorization policy label.
authorizationpolicylabel_authorizationpolicy_binding	<authorizationpolicylabel_authorizationpolicy_binding[]>	Read-only	authorizationpolicy that can be bound to authorizationpolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicylabel_binding/labelname_value<String>

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authorizationpolicylabel_binding": [ {
    "labelname":<String_value>,
    "authorizationpolicylabel_authorizationpolicy_binding":<authorizationpolicylabel_authorizationpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicy_aaagroup_binding

May 20, 2016

Binding object showing the aaagroup that can be bound to authorizationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
priority	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicy_aaagroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_aaagroup_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authorizationpolicy_aaagroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicy_aaagroup_binding": [ {
      "boundto":<String_value>,
      "name":<String_value>,
      "priority":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authorizationpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicy_aaauser_binding

May 20, 2016

Binding object showing the aaauser that can be bound to authorizationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
priority	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicy_aaauser_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_aaauser_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authorizationpolicy_aaauser_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicy_aaauser_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authorizationpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicy_authorizationpolicylabel_binding

May 20, 2016

Binding object showing the authorizationpolicylabel that can be bound to authorizationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
priority	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_authorizationpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_authorizationpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicy_authorizationpolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_authorizationpolicylabel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authorizationpolicy_authorizationpolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicy_authorizationpolicylabel_binding": [ {
      "boundto":<String_value>,
      "name":<String_value>,
      "priority":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_authorizationpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authorizationpolicy_authorizationpolicylabel_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicy_binding

May 20, 2016

Binding object which returns the resources bound to authorizationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
authorizationpolicy_csvserver_binding	<authorizationpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to authorizationpolicy.
authorizationpolicy_lbvserver_binding	<authorizationpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to authorizationpolicy.
authorizationpolicy_aaauser_binding	<authorizationpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to authorizationpolicy.
authorizationpolicy_authorizationpolicylabel_binding	<authorizationpolicy_authorizationpolicylabel_binding[]>	Read-only	authorizationpolicylabel that can be bound to authorizationpolicy.
authorizationpolicy_aaagroup_binding	<authorizationpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to authorizationpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "authorizationpolicy_binding": [ {
    "name":<String_value>,
    "authorizationpolicy_lbvserver_binding":<authorizationpolicy_lbvserver_binding[]_value>,
    "authorizationpolicy_aaagroup_binding":<authorizationpolicy_aaagroup_binding[]_value>,
    "authorizationpolicy_csvserver_binding":<authorizationpolicy_csvserver_binding[]_value>,
    "authorizationpolicy_authorizationpolicylabel_binding":<authorizationpolicy_authorizationpolicylabel_binding[]_value>,
    "authorizationpolicy_aaauser_binding":<authorizationpolicy_aaauser_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicy_csvserver_binding

Jul 07, 2017

Binding object showing the csvserver that can be bound to authorizationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
priority	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authorizationpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicy_csvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authorizationpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to authorizationpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the authorization policy. Minimum length = 1
priority	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of authorizationpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the authorizationpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicy_lbvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/authorizationpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"authorizationpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Autoscale

May 20, 2016

AutoScale configuration.

Name	Description
autoscaleaction	Configuration for autoscale action
autoscalepolicy	Configuration for Autoscale policy
autoscalepolicy_binding	Binding object showing the resources that can be bound to autoscalepolicy
autoscalepolicy_nstimer_binding	Binding object showing the nstimer that can be bound to autoscalepolicy
autoscaleprofile	Configuration for autoscale profile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

autoscaleaction

May 20, 2016

Configuration for autoscale action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	ActionScale action name. Minimum length = 1
type	<String>	Read-write	The type of action. Possible values = SCALE_UP, SCALE_DOWN
profilename	<String>	Read-write	AutoScale profile name. Minimum length = 1
parameters	<String>	Read-write	Parameters to use in the action. Minimum length = 1
vmdestroygraceperiod	<Double>	Read-write	Time in minutes a VM is kept in inactive state before destroying. Default value: 10
quiettime	<Double>	Read-write	Time in seconds no other policy is evaluated or action is taken. Default value: 300
vserver	<String>	Read-write	Name of the vserver on which autoscale action has to be taken.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"autoscaleaction":{
    "name":<String_value>,
    "type":<String_value>,
    "profilename":<String_value>,
    "parameters":<String_value>,
    "vmdestroygraceperiod":<Double_value>,
    "quiettime":<Double_value>,
    "vserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"autoscaleaction":{
    "name":<String_value>,
    "profilename":<String_value>,
    "parameters":<String_value>,
    "vmdestroygraceperiod":<Double_value>,
    "quiettime":<Double_value>,
    "vserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"autoscaleaction":{
    "name":<String_value>,
    "vmdestroygraceperiod":true,
    "quiettime":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of autoscaleaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the autoscaleaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "autoscaleaction": [ {
      "name":<String_value>,
      "type":<String_value>,
      "profilename":<String_value>,
      "parameters":<String_value>,
      "vmdestroygraceperiod":<Double_value>,
      "quiettime":<Double_value>,
      "vserver":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "autoscaleaction": [ {
      "name":<String_value>,
      "type":<String_value>,
      "profilename":<String_value>,
      "parameters":<String_value>,
      "vmdestroygraceperiod":<Double_value>,
      "quiettime":<Double_value>,
      "vserver":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "autoscaleaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

autoscalepolicy

May 20, 2016

Configuration for Autoscale policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the autoscale policy. Minimum length = 1
rule	<String>	Read-write	The rule associated with the policy.
action	<String>	Read-write	The autoscale profile associated with the policy. Minimum length = 1
comment	<String>	Read-write	Comments associated with this autoscale policy.
logaction	<String>	Read-write	The log action associated with the autoscale policy.
newname	<String>	Read-write	The new name of the autoscale policy. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of Undef hits.
priority	<Double>	Read-only	Specifies the priority of the policy.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"autoscalepolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

update

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"autoscalepolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy?**action=unset**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"autoscalepolicy":{
    "name":<String_value>,
    "rule":true,
    "action":true,
    "comment":true,
    "logaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"autoscalepolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of autoscalepolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the autoscalepolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "autoscalepolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Integer_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "autoscalepolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "priority":<Double_value>,
    "activepolicy":<Integer_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "autoscalepolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

autoscalepolicy_binding

May 20, 2016

Binding object which returns the resources bound to autoscalepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the autoscale policy. Minimum length = 1
autoscalepolicy_nstimer_binding	<autoscalepolicy_nstimer_binding[]>	Read-only	nstimer that can be bound to autoscalepolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "autoscalepolicy_binding": [ {
    "name":<String_value>,
    "autoscalepolicy_nstimer_binding":<autoscalepolicy_nstimer_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

autoscalepolicy_nstimer_binding

Jul 07, 2017

Binding object showing the nstimer that can be bound to autoscalepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the autoscale policy. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy_nstimer_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy_nstimer_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of autoscalepolicy_nstimer_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy_nstimer_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the autoscalepolicy_nstimer_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "autoscalepolicy_nstimer_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscalepolicy_nstimer_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"autoscalepolicy_nstimer_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

autoscaleprofile

May 20, 2016

Configuration for autoscale profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	AutoScale profile name. Minimum length = 1
type	<String>	Read-write	The type of profile. Possible values = CLOUDSTACK
url	<String>	Read-write	URL providing the service. Minimum length = 1
apikey	<String>	Read-write	api key for authentication with service. Minimum length = 1
sharedsecret	<String>	Read-write	shared secret for authentication with service. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"autoscaleprofile":{
    "name":<String_value>,
    "type":<String_value>,
    "url":<String_value>,
    "apikey":<String_value>,
    "sharedsecret":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"autoscaleprofile":{
    "name":<String_value>,
    "url":<String_value>,
    "apikey":<String_value>,
    "sharedsecret":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of autoscaleprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the autoscaleprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "autoscaleprofile": [ {
      "name":<String_value>,
      "type":<String_value>,
      "url":<String_value>,
      "apikey":<String_value>,
      "sharedsecret":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "autoscaleprofile": [ {
    "name":<String_value>,
    "type":<String_value>,
    "url":<String_value>,
    "apikey":<String_value>,
    "sharedsecret":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/autoscaleprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{ "autoscaleprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Basic

Jul 07, 2017

Basic system configuration.

Name	Description
configstatus	Configuration for packet engines status
dbsmonitors	Configuration for DB monitors
extendedmemoryparam	Configuration for Parameter for extended memory used by LSN and Subscriber Store
location	Configuration for location
locationdata	Configuration for location data
locationfile	Configuration for location file
locationfile6	Configuration for location file6
locationparameter	Configuration for location parameter
nstrace	Configuration for nstrace operations
reporting	Configuration for reporting
server	Configuration for server
server_binding	Binding object showing the resources that can be bound to server
server_gslbservice_binding	Binding object showing the gslbservice that can be bound to server
server_service_binding	Binding object showing the service that can be bound to server
server_servicegroup_binding	Binding object showing the servicegroup that can be bound to server

© 1999-2017 Citrix Systems, Inc. All rights reserved.

service	
Name	**Description**
service_binding	Binding object showing the resources that can be bound to service
service_dospolicy_binding	Binding object showing the dospolicy that can be bound to service
service_lbmonitor_binding	Binding object showing the lbmonitor that can be bound to service
service_scpolicy_binding	Binding object showing the scpolicy that can be bound to service
servicegroup	Configuration for service group
servicegroup_binding	Binding object showing the resources that can be bound to servicegroup
servicegroup_lbmonitor_binding	Binding object showing the lbmonitor that can be bound to servicegroup
servicegroup_servicegroupentitymonbindings_binding	Binding object showing the servicegroupentitymonbindings that can be bound to servicegroup
servicegroup_servicegroupmember_binding	Binding object showing the servicegroupmember that can be bound to servicegroup
servicegroupbindings	Configuration for servicegroupbind
svcbindings	Configuration for service bindings
vserver	Configuration for virtual server

© 1999-2017 Citrix Systems, Inc. All rights reserved.

configstatus

May 20, 2016

Configuration for packet engines status resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
consistent	<String>	Read-only	State of packet engines. Possible values = YES, NO
culpritcore	<Double>	Read-only	Culprit core id.
core	<Double>	Read-only	Core id.
culpritcoreconfstring	<String>	Read-only	.
coreconfstring	<String>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/configstatus

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/configstatus?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/configstatus?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of configstatus resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/configstatus?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/configstatus?**pagesize=#no&pageno=#no**

Use this query-parameter to get the configstatus resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "configstatus": [ {
    "consistent":<String_value>,
    "culpritcore":<Double_value>,
    "core":<Double_value>,
    "culpritcoreconfstring":<String_value>,
    "coreconfstring":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/configstatus?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "configstatus": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dbsmonitors

May 20, 2016

Configuration for DB monitors resource.

Operations (click to see Properties)

RESTART

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

restart

URL: http://<netscaler-ip-address>/nitro/v1/config/dbsmonitors?**action=restart**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"dbsmonitors":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

extendedmemoryparam

Jul 07, 2017

Configuration for Parameter for extended memory used by LSN and Subscriber Store resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
memlimit	<Double>	Read-write	Amount of NetScaler memory to reserve for the memory used by LSN and Subscriber Session Store feature, in multiples of 2MB. Note: If you later reduce the value of this parameter, the amount of active memory is not reduced. Changing the configured memory limit can only increase the amount of active memory.
memlimitactive	<Double>	Read-only	The active memory limit for extendedmemory on the system. Active memory limit could be different from configured memory limit. This could happen when memory limit could not be increased due to unavailability, or could not be decreased as it is already in use. This active memory limit configures the current memory limit for LSN and Subscriber Session Store.
maxmemlimit	<Double>	Read-only	The maximum value of memory limit for extendedmemory on the system. Actual available memory may be less. This is maximum memory that can be utilized by LSN and Subscriber Session Store modules.
minrequiredmemory	<Double>	Read-only	The minimum memory requirement for extendedmemory. This is minimum memory required for LSN and Subscriber Session Store Modules.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/extendedmemoryparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"extendedmemoryparam":{
    "memlimit":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/extendedmemoryparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"extendedmemoryparam":{
    "memlimit":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/extendedmemoryparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "extendedmemoryparam": [ {
    "memlimitactive":<Double_value>,
    "maxmemlimit":<Double_value>,
    "memlimit":<Double_value>,
    "minrequiredmemory":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

location

May 20, 2016

Configuration for location resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipfrom	<String>	Read-write	First IP address in the range, in dotted decimal notation. Minimum length = 1
ipto	<String>	Read-write	Last IP address in the range, in dotted decimal notation. Minimum length = 1
preferredlocation	<String>	Read-write	String of qualifiers, in dotted notation, describing the geographical location of the IP address range. Each qualifier is more specific than the one that precedes it, as in continent.country.region.city.isp.organization. For example, "NA.US.CA.San Jose.ATT.citrix". Note: A qualifier that includes a dot (.) or space ( ) must be enclosed in double quotation marks. Minimum length = 1
longitude	<Integer>	Read-write	Numerical value, in degrees, specifying the longitude of the geographical location of the IP address-range. Note: Longitude and latitude parameters are used for selecting a service with the static proximity GSLB method. If they are not specified, selection is based on the qualifiers specified for the location. Minimum value = -180 Maximum value = 180
latitude	<Integer>	Read-write	Numerical value, in degrees, specifying the latitude of the geographical location of the IP address-range. Note: Longitude and latitude parameters are used for selecting a service with the static proximity GSLB method. If they are not specified, selection is based on the qualifiers specified for the location. Minimum value = -90 Maximum value = 90
q1label	<String>	Read-only	Least specific location qualifier.
q2label	<String>	Read-only	Location qualifier 2.
q3label	<String>	Read-only	Location qualifier 3.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
q5label	<String>	Read-only	Location qualifier 5.
q6label	<String>	Read-only	Most specific location qualifier.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/location

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"location":{
    "ipfrom":<String_value>,
    "ipto":<String_value>,
    "preferredlocation":<String_value>,
    "longitude":<Integer_value>,
    "latitude":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/location/ipfrom_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/location

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/location?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/location?**filter=property-name1:property-val1,property-name2:property-val2**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of location resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/location?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/location?**pagesize=#no&pageno=#no**

Use this query-parameter to get the location resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "location": [ {
      "ipfrom":<String_value>,
      "ipto":<String_value>,
      "preferredlocation":<String_value>,
      "q1label":<String_value>,
      "q2label":<String_value>,
      "q3label":<String_value>,
      "q4label":<String_value>,
      "q5label":<String_value>,
      "q6label":<String_value>,
      "longitude":<Integer_value>,
      "latitude":<Integer_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/location/ipfrom_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/location/ipfrom_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/location/ipfrom_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "location": [ {
      "ipfrom":<String_value>,
      "ipto":<String_value>,
      "preferredlocation":<String_value>,
      "q1label":<String_value>,
      "q2label":<String_value>,
      "q3label":<String_value>,
      "q4label":<String_value>,
      "q5label":<String_value>,
      "q6label":<String_value>,
      "longitude":<Integer_value>,
      "latitude":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/location?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "location": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

locationdata

May 20, 2016

Configuration for location data resource.

Operations (click to see Properties)

CLEAR

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/locationdata?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"locationdata":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

locationfile

May 20, 2016

Configuration for location file resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
Locationfile	<String>	Read-write	Name of the location file, with or without absolute path. If the path is not included, the default path (/var/netscaler/locdb) is assumed. In a high availability setup, the static database must be stored in the same location on both NetScaler appliances. Minimum length = 1
format	<String>	Read-write	Format of the location file. Required for the NetScaler appliance to identify how to read the location file. Default value: netscaler Possible values = netscaler, ip-country, ip-country-isp, ip-country-region-city, ip-country-region-city-isp, geoip-country, geoip-region, geoip-city, geoip-country-org, geoip-country-isp, geoip-city-isp-org
src	<String>	Read-write	URL \(protocol, host, path, and file name\) from where the location file will be imported. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047

Operations (click to see Properties)

ADD | DELETE | IMPORT | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/locationfile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"locationfile":{
    "Locationfile":<String_value>,
    "format":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/locationfile

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/locationfile?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"locationfile":{
    "Locationfile":<String_value>,
    "src":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/locationfile

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "locationfile": [ {
      "Locationfile":<String_value>,
      "format":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

locationfile6

May 20, 2016

Configuration for location file6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
Locationfile	<String>	Read-write	Name of the IPv6 location file, with or without absolute path. If the path is not included, the default path (/var/netscaler/locdb) is assumed. In a high availability setup, the static database must be stored in the same location on both NetScaler appliances. Minimum length = 1
format	<String>	Read-write	Format of the IPv6 location file. Required for the NetScaler appliance to identify how to read the location file. Default value: netscaler6 Possible values = netscaler6, geoip-country6
src	<String>	Read-write	URL \(protocol, host, path, and file name\) from where the location file will be imported. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | IMPORT | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/locationfile6

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"locationfile6":{
    "Locationfile":<String_value>,
    "format":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/locationfile6

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/locationfile6?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"locationfile6":{
    "Locationfile":<String_value>,
    "src":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/locationfile6

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/locationfile6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/locationfile6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of locationfile6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/locationfile6?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/locationfile6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the locationfile6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "locationfile6": [ {
      "Locationfile":<String_value>,
      "format":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/locationfile6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "locationfile6": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

locationparameter

May 20, 2016

Configuration for location parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
context	<String>	Read-write	Context for describing locations. In geographic context, qualifier labels are assigned by default in the following sequence: Continent.Country.Region.City.ISP.Organization. In custom context, the qualifiers labels can have any meaning that you designate. Possible values = geographic, custom
q1label	<String>	Read-write	Label specifying the meaning of the first qualifier. Can be specified for custom context only. Minimum length = 1
q2label	<String>	Read-write	Label specifying the meaning of the second qualifier. Can be specified for custom context only. Minimum length = 1
q3label	<String>	Read-write	Label specifying the meaning of the third qualifier. Can be specified for custom context only. Minimum length = 1
q4label	<String>	Read-write	Label specifying the meaning of the fourth qualifier. Can be specified for custom context only. Minimum length = 1
q5label	<String>	Read-write	Label specifying the meaning of the fifth qualifier. Can be specified for custom context only. Minimum length = 1
q6label	<String>	Read-write	Label specifying the meaning of the sixth qualifier. Can be specified for custom context only. Minimum length = 1
Locationfile	<String>	Read-only	Currently loaded location database file.
format	<String>	Read-only	Location file format. Possible values = netscaler, ip-country, ip-country-isp, ip-country-region-city, ip-country-region-city-isp, geoip-country, geoip-region, geoip-city, geoip-country-org, geoip-country-isp, geoip-city-isp-org
custom	<Double>	Read-only	Number of configured custom locations.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
static	<Double>	Read-only	Number of configured locations in the database file (static locations).
lines	<Double>	Read-only	Number of lines in the databse files.
errors	<Double>	Read-only	Number of errros encountered while reading the database file.
warnings	<Double>	Read-only	Number of warnings encountered while reading the database file.
entries	<Double>	Read-only	Number of successfully added entries.
locationfile6	<String>	Read-only	Currently loaded location database file.
format6	<String>	Read-only	Location file format. Possible values = netscaler6, geoip-country6
custom6	<Double>	Read-only	Number of configured custom locations.
static6	<Double>	Read-only	Number of configured locations in the database file (static locations).
lines6	<Double>	Read-only	Number of lines in the databse files.
errors6	<Double>	Read-only	Number of errros encountered while reading the database file.
warnings6	<Double>	Read-only	Number of warnings encountered while reading the database file.
entries6	<Double>	Read-only	Number of successfully added entries.
flags	<Double>	Read-only	Information needed for display. This argument passes information from the kernel to the user space.
status	<Double>	Read-only	This argument displays when the status (success or failure) of database loading.
databasemode	<String>	Read-only	This argument displays the database mode. Possible values = File, Internal, Not applicable
flushing	<String>	Read-only	This argument displays the state of flushing. Possible values = In progress, Idle
loading	<String>	Read-only	This argument displays the state of loading. Possible values = In progress, Idle

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/locationparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"locationparameter":{
    "context":<String_value>,
    "q1label":<String_value>,
    "q2label":<String_value>,
    "q3label":<String_value>,
    "q4label":<String_value>,
    "q5label":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "q6label":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/locationparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"locationparameter":{
    "context":true,
    "q1label":true,
    "q2label":true,
    "q3label":true,
    "q4label":true,
    "q5label":true,
    "q6label":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/locationparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "locationparameter": [ {
      "context":<String_value>,
      "q1label":<String_value>,
      "q2label":<String_value>,
      "q3label":<String_value>,
      "q4label":<String_value>,
      "q5label":<String_value>,
      "q6label":<String_value>,
      "Locationfile":<String_value>,
      "format":<String_value>,
      "custom":<Double_value>,
      "Static":<Double_value>,
      "lines":<Double_value>,
      "errors":<Double_value>,
      "warnings":<Double_value>,
      "entries":<Double_value>,
      "locationfile6":<String_value>,
      "format6":<String_value>,
      "custom6":<Double_value>,
      "static6":<Double_value>,
      "lines6":<Double_value>,
      "errors6":<Double_value>,
      "warnings6":<Double_value>,
      "entries6":<Double_value>,
      "flags":<Double_value>,
      "status":<Double_value>,
      "databasemode":<String_value>,
      "flushing":<String_value>,
      "loading":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstrace

May 20, 2016

Configuration for nstrace operations resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
nf	<Double>	Read-write	Number of files to be generated in cycle. Default value: 24 Minimum value = 1 Maximum value = 100
time	<Double>	Read-write	Time per file (sec). Default value: 3600 Minimum value = 1
size	<Double>	Read-write	Size of the captured data. Set 0 for full packet trace. Default value: 164 Minimum value = 0 Maximum value = 1514
mode	<String[]>	Read-write	Capturing mode for trace. Mode can be any of the following values or combination of these values: RX Received packets before NIC pipelining (Filter does not work when RX capturing mode is ON) NEW_RX Received packets after NIC pipelining TX Transmitted packets TXB Packets buffered for transmission IPV6 Translated IPv6 packets C2C Capture C2C message NS_FR_TX TX/TXB packets are not captured in flow receiver. SSLPLAIN Decrypted SSL packets MPTCP MPTCP master flow Default mode: NEW_RX TXB . Default value: DEFAULT_MODE Possible values = TX, TXB, RX, IPV6, NEW_RX, C2C, NS_FR_TX, APPFW, SSLPLAIN, MPTCP
pernic	<String>	Read-write	Use separate trace files for each interface. Works only with tcpdump format. Default value: DISABLED Possible values = ENABLED, DISABLED
filename	<String>	Read-write	Name of the trace file.
fileid	<String>	Read-write	ID for the trace file name for uniqueness. Should be used only with -name option.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Filter expression for nstrace. Maximum length of filter is 255 and it can be of following format: <expression> [<relop> <expression>] <relop> = (&& | ||) nstrace supports two types of filter expressions: Classic Expressions: [Note: Classic Expressions are not supported in non-default partitions] <expression> = the expression string in the format: <qualifier> <operator> <qualifier-value> <qualifier> = SOURCEIP. <qualifier-value> = A valid IP address <qualifier> = SOURCEPORT. <qualifier-value> = A valid port number. <qualifier> = DESTIP. <qualifier-value> = A valid IP address. <qualifier> = DESTPORT. <qualifier-value> = A valid port number. <qualifier> = IP. <qualifier-value> = A valid IP address. <qualifier> = PORT. <qualifier-value> = A valid port number. <qualifier> = SVCNAME. <qualifier-value> = The name of a service. <qualifier> = VSVRNAME. <qualifier-value> = The name of a vserver. <qualifier> = CONNID <qualifier-value> = A valid PCB dev number. <qualifier> = VLAN <qualifier-value> = A valid VLAN ID. <qualifier> = INTF <qualifier-value> = A valid interface id in the form of x/y (n/x/y in case of cluster interface). <operator> = (== | eq | != | neq | > | gt | < | lt | >= | ge | <= | le | BETWEEN) eg: start nstrace -filter "SOURCEIP == 10.102.34.201 || (SVCNAME != s1 && SOURCEPORT > 80)" The filter expression should be given in double quotes. Default Expressions: <expression> =: CONNECTION.<qualifier>.<qualifier-method>.(<qualifier-value>) <qualifier> = SRCIP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address. example = CONNECTION.SRCIP.EQ(127.0.0.1)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			<qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address. example = CONNECTION.DSTIP.EQ(127.0.0.1)
filter	<String>	Read-write	<qualifier> = IP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address. example = CONNECTION.IP.EQ(127.0.0.1) <qualifier> = SRCIPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.SRCIPv6.EQ(2001:db8:0:0:1::1) <qualifier> = DSTIPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.DSTIPv6.EQ(2001:db8:0:0:1::1) <qualifier> = IPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.IPv6.EQ(2001:db8:0:0:1::1) <qualifier> = SRCPORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.SRCPORT.EQ(80) <qualifier> = DSTPORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.DSTPORT.EQ(80) <qualifier> = PORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.PORT.EQ(80) <qualifier> = VLANID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid VLAN ID. example = CONNECTION.VLANID.EQ(0) <qualifier> = CONNID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid PCB dev number. example = CONNECTION.CONNID.EQ(0)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			<qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid core ID. example = CONNECTION.PPEID.EQ(0) <qualifier> = SVCNAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = A valid text string. example = CONNECTION.SVCNAME.EQ("name") <qualifier> = LB_VSERVER.NAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = LB vserver name. example = CONNECTION.LB_VSERVER.NAME.EQ("name") <qualifier> = CS_VSERVER.NAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = CS vserver name. example = CONNECTION.CS_VSERVER.NAME.EQ("name") <qualifier> = INTF <qualifier-method> = [EQ | NE] <qualifier-value> = A valid interface id in the form of x/y. example = CONNECTION.INTF.EQ("x/y") <qualifier> = SERVICE_TYPE <qualifier-method> = [EQ | NE] <qualifier-value> = (SVC_HTTP | FTP | TCP | UDP | SSL | SSL_BRIDGE | SSL_TCP | NNTP | RPCSVR | RPCSVRS | RPCCLNT | SVC_DNS | ADNS | SNMP | RTSP | DHCPRA | ANY| MONITOR | MONITOR_UDP | MONITOR_PING | SIP_UDP | SVC_MYSQL | SVC_MSSQL | SERVICE_UNKNOWN) example = CONNECTION.SERVICE_TYPE.EQ(ANY) <qualifier> = TRAFFIC_DOMAIN_ID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid traffic domain ID. example = CONNECTION.TRAFFIC_DOMAIN_ID.EQ(0) eg: start nstrace -filter "CONNECTION.SRCIP.EQ(127.0.0.1) | | (CONNECTION.SVCNAME.NE("s1") && CONNECTION.SRCPORT.EQ(80))" The filter expression should be given in double quotes. common use cases: Trace capturing full sized traffic from/to ip 10.102.44.111, excluding loopback traffic start nstrace -size 0 -filter "CONNECTION.IP.NE(127.0.0.1) && CONNECTION.IP.EQ(10.102.44.111)"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Trace capturing all traffic to (terminating at) port 80 or 443 start nstrace -size 0 -filter "CONNECTION.DSTPORT.EQ(443) || CONNECTION.DSTPORT.EQ(80)" Trace capturing all backend traffic specific to service service1 along with corresponding client side traffic start nstrace -size 0 -filter "CONNECTION.SVCNAME.EQ("service1")" -link ENABLED Trace capturing all traffic through NS interface 1/1 start nstrace -filter "CONNECTION.INTF.EQ("1/1")" Trace capturing all traffic specific through vlan 2 start nstrace -filter "CONNECTION.VLANID.EQ(2)" Trace capturing all frontend (client side) traffic specific to lb vserver vserver1 along with corresponding server side traffic start nstrace -size 0 -filter "CONNECTION.LB_VSERVER.NAME.EQ("vserver1")" -link ENABLED .
link	<String>	Read-write	Includes filtered connection's peer traffic. Default value: DISABLED Possible values = ENABLED, DISABLED
nodes	<Double[]>	Read-write	Nodes on which tracing is started. Minimum value = 0 Maximum value = 32
filesize	<Double>	Read-write	File size, in MB, treshold for rollover. Default value: 0 Minimum value = 0 Maximum value = 65536
traceformat	<String>	Read-write	Format in which trace will be generated . Default value: 0 Possible values = NSCAP, PCAP
merge	<String>	Read-write	Specify how traces across PE's are merged . Default value: 0 Possible values = ONSTOP, ONTHEFLY, NOMERGE
doruntimecleanup	<String>	Read-write	Enable or disable runtime temp file cleanup. Default value: ENABLED Possible values = ENABLED, DISABLED
tracebuffers	<Double>	Read-write	Number of 16KB trace buffers. Default value: 5000 Minimum value = 1000
skiprpc	<String>	Read-write	skip RPC packets. Default value: DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
capsslkeys	<String>	Read-write	Capture SSL Master keys. Master keys will not be captured on FIPS machine. Default value: DISABLED Possible values = ENABLED, DISABLED
inmemorytrace	<String>	Read-write	Logs packets in appliance's memory and dumps the trace file on stopping the nstrace operation. Default value: DISABLED Possible values = ENABLED, DISABLED
state	<String>	Read-only	Current running state of trace. Default value: 0 Possible values = RUNNING, STOPPED
scope	<String>	Read-only	Scope of started trace, local or cluster level. Default value: 0 Possible values = CLUSTER, LOCAL
tracelocation	<String>	Read-only	Directory where current trace files are saved.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrace

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstrace": [ {
      "state":<String_value>,
      "scope":<String_value>,
      "tracelocation":<String_value>,
      "nf":<Double_value>,
      "time":<Double_value>,
      "size":<Double_value>,
      "mode":<String[]_value>,
      "traceformat":<String_value>,
      "pernic":<String_value>,
      "filename":<String_value>,
      "fileid":<String_value>,
      "filter":<String_value>,
      "link":<String_value>,
      "nodes":<Double[]_value>,
      "merge":<String_value>,
      "doruntimecleanup":<String_value>,
      "tracebuffers":<Double_value>,
      "skiprpc":<String_value>,
      "capssllkeys":<String_value>,
      "inmemorytrace":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

reporting

Jul 07, 2017

Configuration for reporting resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-only	The rule associated with the entity. Possible values = ENABLED, DISABLED

Operations (click to see Properties)

ENABLE | DISABLE | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

enable

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/reporting?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"reporting":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/reporting?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"reporting":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/reporting

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "reporting": [ {
      "state":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

server

May 20, 2016

Configuration for server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the server. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the name is created. Minimum length = 1
ipaddress	<String>	Read-write	IPv4 or IPv6 address of the server. If you create an IP address based server, you can specify the name of the server, instead of its IP address, when creating a service. Note: If you do not create a server entry, the server IP address that you enter when you create a service becomes the name of the server.
domain	<String>	Read-write	Domain name of the server. For a domain based configuration, you must create the server first. Minimum length = 1
translationip	<String>	Read-write	IP address used to transform the server's DNS-resolved IP address.
translationmask	<String>	Read-write	The netmask of the translation ip.
domainresolveretry	<Integer>	Read-write	Time, in seconds, for which the NetScaler appliance must wait, after DNS resolution fails, before sending the next DNS query to resolve the domain name. Default value: 5 Minimum value = 5 Maximum value = 20939
state	<String>	Read-write	Initial state of the server. Default value: ENABLED Possible values = ENABLED, DISABLED
ipv6address	<String>	Read-write	Support IPv6 addressing mode. If you configure a server with the IPv6 addressing mode, you cannot use the server in the IPv4 addressing mode. Default value: NO Possible values = YES, NO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
domainresolvenow	<Boolean>	Read-write	Immediately send a DNS query to resolve the server's domain name.
delay	<Double>	Read-write	Time, in seconds, after which all the services configured on the server are disabled.
graceful	<String>	Read-write	Shut down gracefully, without accepting any new connections, and disabling each service when all of its connections are closed. Default value: NO Possible values = YES, NO
Internal	<Boolean>	Read-write	Display names of the servers that have been created for internal use.
newname	<String>	Read-write	New name for the server. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
autoscale	<String>	Read-only	Auto scale option for a servicegroup. Default value: DISABLED Possible values = DISABLED, DNS, POLICY
customserverid	<String>	Read-only	A positive integer to identify the service. Used when the persistency type is set to Custom Server ID. Default value: "None"
monthreshold	<Double>	Read-only	Minimum sum of weights of the monitors that are bound to this service. Used to determine whether to mark a service as UP or DOWN. Minimum value = 0 Maximum value = 65535
maxclient	<Double>	Read-only	Maximum number of simultaneous open connections for the service group. Minimum value = 0 Maximum value = 4294967294
maxreq	<Double>	Read-only	Maximum number of requests that can be sent on a persistent connection to the service group. Note: Connection requests beyond this value are rejected. Minimum value = 0 Maximum value = 65535

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
maxbandwidth	<Double>	Read-only	Maximum bandwidth, in Kbps, allocated for all the services in the service group. Minimum value = 0 Maximum value = 4294967287
usip	<String>	Read-only	Use the client's IP address as the source IP address when initiating a connection to the server. When creating a service, if you do not set this parameter, the service inherits the global Use Source IP setting (available in the enable ns mode and disable ns mode CLI commands, or in the System > Settings > Configure modes > Configure Modes dialog box). However, you can override this setting after you create the service. Possible values = YES, NO
cka	<String>	Read-only	Enable client keep-alive for the service group. Possible values = YES, NO
tcpb	<String>	Read-only	Enable TCP buffering for the service group. Possible values = YES, NO
cmp	<String>	Read-only	Enable compression for the specified service. Possible values = YES, NO
clttimeout	<Double>	Read-only	Time, in seconds, after which to terminate an idle client connection. Minimum value = 0 Maximum value = 31536000
svrtimeout	<Double>	Read-only	Time, in seconds, after which to terminate an idle server connection. Minimum value = 0 Maximum value = 31536000
cipheader	<String>	Read-only	Name of the HTTP header whose value must be set to the IP address of the client. Used with the Client IP parameter. If client IP insertion is enabled, and the client IP header is not specified, the value of Client IP Header parameter or the value set by the set ns config command is used as client's IP header name. Minimum length = 1
cip	<String>	Read-only	Before forwarding a request to the service, insert an HTTP header with the client's IPv4 or IPv6 address as its value. Used if the server needs the client's IP address for security, accounting, or other purposes, and setting the Use Source IP parameter is not a viable option. Possible values = ENABLED, DISABLED
cacheable	<String>	Read-only	Use the transparent cache redirection virtual server to forward the request to the cache server. Default value: NO Possible values = YES, NO
sc	<String>	Read-only	State of the SureConnect feature for the service group. Default value: OFF Possible values = ON, OFF
sp	<String>	Read-only	Enable surge protection for the service group. Default value: OFF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
downstateflush	<String>	Read-only	Perform delayed clean-up of connections to all services in the service group. Default value: ENABLED Possible values = ENABLED, DISABLED
appflowlog	<String>	Read-only	Enable logging of AppFlow information for the specified service group. Default value: ENABLED Possible values = ENABLED, DISABLED
boundtd	<Double>	Read-only	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/server

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"server":{
    "name":<String_value>,
    "ipaddress":<String_value>,
    "domain":<String_value>,
    "translationip":<String_value>,
    "translationmask":<String_value>,
    "domainresolveretry":<Integer_value>,
    "state":<String_value>,
    "ipv6address":<String_value>,
    "comment":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/server/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/server

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"server":{
    "name":<String_value>,
    "ipaddress":<String_value>,
    "domainresolveretry":<Integer_value>,
    "translationip":<String_value>,
    "translationmask":<String_value>,
    "domainresolvenow":<Boolean_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/server?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"server":{
    "name":<String_value>,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/server?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"server":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/server?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"server":{
    "name":<String_value>,
    "delay":<Double_value>,
    "graceful":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/server?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"server":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/server

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/server?**args=name:<String_value>,Internal:<Boolean_value>**

Use this query-parameter to get server resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/server?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/server?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of server resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/server?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/server?**pagesize=#no&pageno=#no**

Use this query-parameter to get the server resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "server": [ {
name:<String_value>,Internal:<Boolean_value>      "ipaddress":<String_value>,
      "state":<String_value>,
      "domain":<String_value>,
      "domainresolveretry":<Integer_value>,
      "translationip":<String_value>,
      "translationmask":<String_value>,
      "comment":<String_value>,
      "statechangetimesec":<String_value>,
      "tickssincelaststatechange":<Double_value>,
      "ipv6address":<String_value>,
      "td":<Double_value>,
      "autoscale":<String_value>,
      "customserverid":<String_value>,
      "monthreshold":<Double_value>,
      "maxclient":<Double_value>,
      "maxreq":<Double_value>,
      "maxbandwidth":<Double_value>,
      "usip":<String_value>,
      "cka":<String_value>,
      "tcpb":<String_value>,
      "cmp":<String_value>,
      "clttimeout":<Double_value>,
      "svrtimeout":<Double_value>,
      "cipheader":<String_value>,
      "cip":<String_value>,
      "cacheable":<String_value>,
      "sc":<String_value>,
      "sp":<String_value>,
      "downstateflush":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "appflowlog":<String_value>,
    "boundtd":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/server/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/server/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/server/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "server": [ {
name:<String_value>,Internal:<Boolean_value>      "ipaddress":<String_value>,
    "state":<String_value>,
    "domain":<String_value>,
    "domainresolveretry":<Integer_value>,
    "translationip":<String_value>,
    "translationmask":<String_value>,
    "comment":<String_value>,
    "statechangetimesec":<String_value>,
    "tickssincelaststatechange":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ipv6address":<String_value>,
    "td":<Double_value>,
    "autoscale":<String_value>,
    "customserverid":<String_value>,
    "monthreshold":<Double_value>,
    "maxclient":<Double_value>,
    "maxreq":<Double_value>,
    "maxbandwidth":<Double_value>,
    "usip":<String_value>,
    "cka":<String_value>,
    "tcpb":<String_value>,
    "cmp":<String_value>,
    "clttimeout":<Double_value>,
    "svrtimeout":<Double_value>,
    "cipheader":<String_value>,
    "cip":<String_value>,
    "cacheable":<String_value>,
    "sc":<String_value>,
    "sp":<String_value>,
    "downstateflush":<String_value>,
    "appflowlog":<String_value>,
    "boundtd":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/server?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "server": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

server_binding

May 20, 2016

Binding object which returns the resources bound to server.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the server for which to display parameters. Minimum length = 1
server_service_binding	<server_service_binding[]>	Read-only	service that can be bound to server.
server_gslbservice_binding	<server_gslbservice_binding[]>	Read-only	gslbservice that can be bound to server.
server_servicegroup_binding	<server_servicegroup_binding[]>	Read-only	servicegroup that can be bound to server.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/server_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "server_binding": [ {
    "name":<String_value>,
    "server_servicegroup_binding":<server_servicegroup_binding[]_value>,
    "server_service_binding":<server_service_binding[]_value>,
    "server_gslbservice_binding":<server_gslbservice_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

server_gslbservice_binding

May 20, 2016

Binding object showing the gslbservice that can be bound to server.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the server for which to display parameters. Minimum length = 1
servicename	<String>	Read-write	The services attatched to the server.
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
svctype	<String>	Read-only	The type of bound service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX
svrstate	<String>	Read-only	The state of the bound service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
port	<Integer>	Read-only	The port number to be used for the bound service. Range 1 - 65535
serviceipaddress	<String>	Read-only	The IP address of the bound service.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/server_gslbservice_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/server_gslbservice_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of server_gslbservice_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/server_gslbservice_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the server_gslbservice_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "server_gslbservice_binding": [ {
    "name":<String_value>,
    "servicename":<String_value>,
    "serviceipstr":<String_value>,
    "svctype":<String_value>,
    "svrstate":<String_value>,
    "port":<Integer_value>,
    "serviceipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/server_gslbservice_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"server_gslbservice_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

server_servicegroup_binding

May 20, 2016

Binding object showing the servicegroup that can be bound to server.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the server for which to display parameters. Minimum length = 1
servicegroupname	<String>	Read-write	servicegroups bind to this server.
sp	<String>	Read-only	Enable surge protection for the service group. Default value: OFF Possible values = ON, OFF
sc	<String>	Read-only	State of the SureConnect feature for the service group. Default value: OFF Possible values = ON, OFF
svrtimeout	<Double>	Read-only	Time, in seconds, after which to terminate an idle server connection. Minimum value = 0 Maximum value = 31536000
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
tcpb	<String>	Read-only	Enable TCP buffering for the service group. Possible values = YES, NO
cip	<String>	Read-only	Before forwarding a request to the service, insert an HTTP header with the client's IPv4 or IPv6 address as its value. Used if the server needs the client's IP address for security, accounting, or other purposes, and setting the Use Source IP parameter is not a viable option. Possible values = ENABLED, DISABLED
dup_port	<Integer>	Read-only	port of the service. Range 1 - 65535
customserverid	<String>	Read-only	A positive integer to identify the service. Used when the persistency type is set to Custom Server ID. Default value: "None"
			Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 0 Maximum value = 4094
downstateflush	<String>	Read-only	Perform delayed clean-up of connections to all services in the service group. Default value: ENABLED Possible values = ENABLED, DISABLED
svctype	<String>	Read-only	The type of bound service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX
cka	<String>	Read-only	Enable client keep-alive for the service group. Possible values = YES, NO
svrstate	<String>	Read-only	The state of the bound service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
appflowlog	<String>	Read-only	Enable logging of AppFlow information for the specified service group. Default value: ENABLED Possible values = ENABLED, DISABLED
cipheader	<String>	Read-only	Name of the HTTP header whose value must be set to the IP address of the client. Used with the Client IP parameter. If client IP insertion is enabled, and the client IP header is not specified, the value of Client IP Header parameter or the value set by the set ns config command is used as client's IP header name. Minimum length = 1
maxclient	<Double>	Read-only	Maximum number of simultaneous open connections for the service group. Minimum value = 0 Maximum value = 4294967294
clttimeout	<Double>	Read-only	Time, in seconds, after which to terminate an idle client connection. Minimum value = 0 Maximum value = 31536000
port	<Integer>	Read-only	The port number to be used for the bound service. Range 1 - 65535
dup_svctype	<String>	Read-only	service type of the service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX
cmp	<String>	Read-only	Enable compression for the specified service. Possible values = YES, NO
monthreshold	<Double>	Read-only	Minimum sum of weights of the monitors that are bound to this service. Used to determine whether to mark a service as UP or DOWN. Minimum value = 0

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
maxreq	<Double>	Read-only	Maximum number of requests that can be sent on a persistent connection to the service group. Note: Connection requests beyond this value are rejected. Minimum value = 0 Maximum value = 65535
serviceipaddress	<String>	Read-only	The IP address of the bound service.
usip	<String>	Read-only	Use the client's IP address as the source IP address when initiating a connection to the server. When creating a service, if you do not set this parameter, the service inherits the global Use Source IP setting (available in the enable ns mode and disable ns mode CLI commands, or in the System > Settings > Configure modes > Configure Modes dialog box). However, you can override this setting after you create the service. Possible values = YES, NO
cacheable	<String>	Read-only	Use the transparent cache redirection virtual server to forward the request to the cache server. Default value: NO Possible values = YES, NO
maxbandwidth	<Double>	Read-only	Maximum bandwidth, in Kbps, allocated for all the services in the service group. Minimum value = 0 Maximum value = 4294967287
svrcfgflags	<Double>	Read-only	service flags to denote its a db enabled.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/server_servicegroup_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/server_servicegroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of server_servicegroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/server_servicegroup_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the server_servicegroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "server_servicegroup_binding": [ {
    "name":<String_value>,
    "servicegroupname":<String_value>,
    "sp":<String_value>,
    "sc":<String_value>,
    "svrtimeout":<Double_value>,
    "serviceipstr":<String_value>,
    "tcpb":<String_value>,
    "cip":<String_value>,
    "dup_port":<Integer_value>,
    "customserverid":<String_value>,
    "boundtd":<Double_value>,
    "downstateflush":<String_value>,
    "svctype":<String_value>,
    "cka":<String_value>,
    "svrstate":<String_value>,
    "appflowlog":<String_value>,
    "cipheader":<String_value>,
    "maxclient":<Double_value>,
    "clttimeout":<Double_value>,
    "port":<Integer_value>,
    "dup_svctype":<String_value>,
    "cmp":<String_value>,
    "monthreshold":<Double_value>,
    "maxreq":<Double_value>,
    "serviceipaddress":<String_value>,
    "usip":<String_value>,
    "cacheable":<String_value>,
    "maxbandwidth":<Double_value>,
    "svrcfgflags":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/server_servicegroup_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"server_servicegroup_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

server_service_binding

Jul 07, 2017

Binding object showing the service that can be bound to server.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the server for which to display parameters. Minimum length = 1
servicename	<String>	Read-write	The services attatched to the server.
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
svctype	<String>	Read-only	The type of bound service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX
svrstate	<String>	Read-only	The state of the bound service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
port	<Integer>	Read-only	The port number to be used for the bound service. Range 1 - 65535
serviceipaddress	<String>	Read-only	The IP address of the bound service.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/server_service_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/server_service_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of server_service_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/server_service_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the server_service_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "server_service_binding": [ {
    "name":<String_value>,
    "servicename":<String_value>,
    "serviceipstr":<String_value>,
    "svctype":<String_value>,
    "svrstate":<String_value>,
    "port":<Integer_value>,
    "serviceipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/server_service_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"server_service_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

service

May 20, 2016

Configuration for service resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the service. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the service has been created. Minimum length = 1
ip	<String>	Read-write	IP to assign to the service. Minimum length = 1
servername	<String>	Read-write	Name of the server that hosts the service. Minimum length = 1
servicetype	<String>	Read-write	Protocol in which data is exchanged with the service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX
port	<Integer>	Read-write	Port number of the service. Range 1 - 65535
cleartextport	<Integer>	Read-write	Port to which clear text data must be sent after the appliance decrypts incoming SSL traffic. Applicable to transparent SSL services. Minimum value = 1
cachetype	<String>	Read-write	Cache type supported by the cache server. Possible values = TRANSPARENT, REVERSE, FORWARD
maxclient	<Double>	Read-write	Maximum number of simultaneous open connections to the service. Minimum value = 0 Maximum value = 4294967294
healthmonitor	<String>	Read-write	Monitor the health of this service. Available settings function as follows: YES - Send probes to check the health of the service. NO - Do not send probes to check the health of the service. With the NO option, the appliance shows the service as UP at all times. Default value: YES Possible values = YES, NO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
maxreq	<Double>	Read-write	to the service. Note: Connection requests beyond this value are rejected. Minimum value = 0 Maximum value = 65535
cacheable	<String>	Read-write	Use the transparent cache redirection virtual server to forward requests to the cache server. Note: Do not specify this parameter if you set the Cache Type parameter. Default value: NO Possible values = YES, NO
cip	<String>	Read-write	Before forwarding a request to the service, insert an HTTP header with the client's IPv4 or IPv6 address as its value. Used if the server needs the client's IP address for security, accounting, or other purposes, and setting the Use Source IP parameter is not a viable option. Possible values = ENABLED, DISABLED
cipheader	<String>	Read-write	Name for the HTTP header whose value must be set to the IP address of the client. Used with the Client IP parameter. If you set the Client IP parameter, and you do not specify a name for the header, the appliance uses the header name specified for the global Client IP Header parameter (the cipHeader parameter in the set ns param CLI command or the Client IP Header parameter in the Configure HTTP Parameters dialog box at System > Settings > Change HTTP parameters). If the global Client IP Header parameter is not specified, the appliance inserts a header with the name "client-ip.". Minimum length = 1
usip	<String>	Read-write	Use the client's IP address as the source IP address when initiating a connection to the server. When creating a service, if you do not set this parameter, the service inherits the global Use Source IP setting (available in the enable ns mode and disable ns mode CLI commands, or in the System > Settings > Configure modes > Configure Modes dialog box). However, you can override this setting after you create the service. Possible values = YES, NO
pathmonitor	<String>	Read-write	Path monitoring for clustering. Possible values = YES, NO
pathmonitorindv	<String>	Read-write	Individual Path monitoring decisions. Possible values = YES, NO
useproxyport	<String>	Read-write	Use the proxy port as the source port when initiating connections with the server. With the NO setting, the client-side connection port is used as the source port for the server-side connection. Note: This parameter is available only when the Use Source IP (USIP) parameter is set to YES. Possible values = YES, NO
sc	<String>	Read-write	State of SureConnect for the service. Default value: OFF Possible values = ON, OFF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = ON, OFF
rtspsessionidremap	<String>	Read-write	Enable RTSP session ID mapping for the service. Default value: OFF Possible values = ON, OFF
clttimeout	<Double>	Read-write	Time, in seconds, after which to terminate an idle client connection. Minimum value = 0 Maximum value = 31536000
svrtimeout	<Double>	Read-write	Time, in seconds, after which to terminate an idle server connection. Minimum value = 0 Maximum value = 31536000
customserverid	<String>	Read-write	Unique identifier for the service. Used when the persistency type for the virtual server is set to Custom Server ID. Default value: "None"
serverid	<Double>	Read-write	The identifier for the service. This is used when the persistency type is set to Custom Server ID.
cka	<String>	Read-write	Enable client keep-alive for the service. Possible values = YES, NO
tcpb	<String>	Read-write	Enable TCP buffering for the service. Possible values = YES, NO
cmp	<String>	Read-write	Enable compression for the service. Possible values = YES, NO
maxbandwidth	<Double>	Read-write	Maximum bandwidth, in Kbps, allocated to the service. Minimum value = 0 Maximum value = 4294967287
accessdown	<String>	Read-write	Use Layer 2 mode to bridge the packets sent to this service if it is marked as DOWN. If the service is DOWN, and this parameter is disabled, the packets are dropped. Default value: NO Possible values = YES, NO
monthreshold	<Double>	Read-write	Minimum sum of weights of the monitors that are bound to this service. Used to determine whether to mark a service as UP or DOWN. Minimum value = 0 Maximum value = 65535
state	<String>	Read-write	Initial state of the service. Default value: ENABLED Possible values = ENABLED, DISABLED
downstateflush	<String>	Read-write	Flush all active transactions associated with a service whose state transitions from UP to DOWN. Do not enable this option for applications that must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
tcpprofilename	<String>	Read-write	Name of the TCP profile that contains TCP configuration settings for the service. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the HTTP profile that contains HTTP configuration settings for the service. Minimum length = 1 Maximum length = 127
hashid	<Double>	Read-write	A numerical identifier that can be used by hash based load balancing methods. Must be unique for each service. Minimum value = 1
comment	<String>	Read-write	Any information about the service.
appflowlog	<String>	Read-write	Enable logging of AppFlow information. Default value: ENABLED Possible values = ENABLED, DISABLED
netprofile	<String>	Read-write	Network profile to use for the service. Minimum length = 1 Maximum length = 127
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
processlocal	<String>	Read-write	By turning on this option packets destined to a service in a cluster will not under go any steering. Turn this option for single packet request response mode or when the upstream device is performing a proper RSS for connection based distribution. Default value: DISABLED Possible values = ENABLED, DISABLED
dnsprofilename	<String>	Read-write	Name of the DNS profile to be associated with the service. DNS profile properties will applied to the transactions processed by a service. This parameter is valid only for ADNS and ADNS-TCP services. Minimum length = 1 Maximum length = 127
ipaddress	<String>	Read-write	The new IP address of the service.
weight	<Double>	Read-write	Weight to assign to the monitor-service binding. When a monitor is UP, the weight assigned to its binding with the service determines how much the monitor contributes toward keeping the health of the service above the value configured for the Monitor Threshold parameter. Minimum value = 1 Maximum value = 100

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
riseapbrstatsmsgcode	<Integer>	Read-write	The code indicating the rise apbr status.
delay	<Double>	Read-write	Time, in seconds, allocated to the NetScaler appliance for a graceful shutdown of the service. During this period, new requests are sent to the service only for clients who already have persistent sessions on the appliance. Requests from new clients are load balanced among other available services. After the delay time expires, no requests are sent to the service, and the service is marked as unavailable (OUT OF SERVICE).
graceful	<String>	Read-write	Shut down gracefully, not accepting any new connections, and disabling the service when all of its connections are closed. Default value: NO Possible values = YES, NO
all	<Boolean>	Read-write	Display both user-configured and dynamically learned services.
Internal	<Boolean>	Read-write	Display only dynamically learned services.
newname	<String>	Read-write	New name for the service. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
numofconnections	<Integer>	Read-only	This will tell the number of client side connections are still open.
policyname	<String>	Read-only	The name of the policyname for which this service is bound.
serviceconftype	<Boolean>	Read-only	The configuration type of the service.
serviceconftype2	<String>	Read-only	The configuration type of the service (Internal/Dynamic/Configured). Possible values = Configured, Dynamic, Internal
value	<String>	Read-only	SSL status. Possible values = Certkey not bound, SSL feature disabled
gslb	<String>	Read-only	The GSLB option for the corresponding virtual server. Possible values = REMOTE, LOCAL
dup_state	<String>	Read-only	Added this field for getting state value from table. Possible values = ENABLED, DISABLED
publicip	<String>	Read-only	public ip.
			public port.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Range 1 - 65535
svrstate	<String>	Read-only	The state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monitor_state	<String>	Read-only	The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
lastresponse	<String>	Read-only	The string form of monstatcode.
responsetime	<Double>	Read-only	Response time of this monitor.
riseapbrstatsmsgcode2	<Integer>	Read-only	The code indicating other rise stats.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
statechangetimemsec	<Double>	Read-only	Time at which last state change happened. Milliseconds part.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
stateupdatereason	<Double>	Read-only	Checks state update reason on the secondary node.
clmonowner	<Double>	Read-only	Tells the mon owner of the service. Minimum value = 0 Maximum value = 32
clmonview	<Double>	Read-only	Tells the view id of the monitoring owner.
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
oracleserverversion	<String>	Read-only	Default value: 10G Possible values = 10G, 11G
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/service

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"service":{
    "name":<String_value>,
    "ip":<String_value>,
    "servername":<String_value>,
    "servicetype":<String_value>,
    "port":<Integer_value>,
    "cleartextport":<Integer_value>,
    "cachetype":<String_value>,
    "maxclient":<Double_value>,
    "healthmonitor":<String_value>,
    "maxreq":<Double_value>,
    "cacheable":<String_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "usip":<String_value>,
    "pathmonitor":<String_value>,
    "pathmonitorindv":<String_value>,
    "useproxyport":<String_value>,
    "sc":<String_value>,
    "sp":<String_value>,
    "rtspsessionidremap":<String_value>,
    "clttimeout":<Double_value>,
    "svrtimeout":<Double_value>,
    "customserverid":<String_value>,
    "serverid":<Double_value>,
    "cka":<String_value>,
    "tcpb":<String_value>,
    "cmp":<String_value>,
    "maxbandwidth":<Double_value>,
    "accessdown":<String_value>,
    "monthreshold":<Double_value>,
    "state":<String_value>,
    "downstateflush":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "hashid":<Double_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "td":<Double_value>,
    "processlocal":<String_value>,
    "dnsprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/service/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/service

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"service":{
    "name":<String_value>,
    "ipaddress":<String_value>,
    "maxclient":<Double_value>,
    "maxreq":<Double_value>,
    "cacheable":<String_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "usip":<String_value>,
    "pathmonitor":<String_value>,
    "pathmonitorindv":<String_value>,
    "useproxyport":<String_value>,
    "sc":<String_value>,
    "sp":<String_value>,
    "rtspsessionidremap":<String_value>,
    "healthmonitor":<String_value>,
    "clttimeout":<Double_value>,
    "svrtimeout":<Double_value>,
    "customserverid":<String_value>,
    "serverid":<Double_value>,
    "cka":<String_value>,
    "tcpb":<String_value>,
    "cmp":<String_value>,
    "maxbandwidth":<Double_value>,
    "accessdown":<String_value>,
    "monthreshold":<Double_value>,
    "weight":<Double_value>,
    "monitor_name_svc":<String_value>,
    "downstateflush":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "hashid":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "processlocal":<String_value>,
    "dnsprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/service?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"service":{
    "name":<String_value>,
    "maxclient":true,
    "maxreq":true,
    "cacheable":true,
    "cip":true,
    "usip":true,
    "pathmonitor":true,
    "pathmonitorindv":true,
    "useproxyport":true,
    "sc":true,
    "sp":true,
    "rtspsessionidremap":true,
    "customserverid":true,
    "serverid":true,
    "cka":true,
    "tcpb":true,
    "cmp":true,
    "maxbandwidth":true,
    "accessdown":true,
    "monthreshold":true,
    "clttimeout":true,
    "riseapbrstatsmsgcode":true,
    "svrtimeout":true,
    "tcpprofilename":true,
    "httpprofilename":true,
    "hashid":true,
    "appflowlog":true,
    "netprofile":true,
    "processlocal":true,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "dnsprofilename":true,
    "cipheader":true,
    "healthmonitor":true,
    "downstateflush":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/service?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"service":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/service?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"service":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "delay":<Double_value>,
    "graceful":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/service?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"service":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/service

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/service?**args=name:<String_value>,all:<Boolean_value>,Internal:<Boolean_value>**

Use this query-parameter to get service resources based on additional properties.

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/service?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/service?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of service resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/service?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/service?**pagesize=#no&pageno=#no**

Use this query-parameter to get the service resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "service": [ {
name:<String_value>,all:<Boolean_value>,Internal:<Boolean_value>      "numofconnections":<Integer_value>,
      "servername":<String_value>,
      "policyname":<String_value>,
      "servicetype":<String_value>,
      "serviceconftpye":<Boolean_value>,
      "serviceconftype":<Boolean_value>,
      "serviceconftype2":<String_value>,
      "port":<Integer_value>,
      "value":<String_value>,
      "cleartextport":<Integer_value>,
      "gslb":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cachetype":<String_value>,
    "maxclient":<Double_value>,
    "maxreq":<Double_value>,
    "cacheable":<String_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "usip":<String_value>,
    "pathmonitor":<String_value>,
    "pathmonitorindv":<String_value>,
    "useproxyport":<String_value>,
    "sc":<String_value>,
    "dup_state":<String_value>,
    "sp":<String_value>,
    "rtspsessionidremap":<String_value>,
    "clttimeout":<Double_value>,
    "svrtimeout":<Double_value>,
    "publicip":<String_value>,
    "publicport":<Integer_value>,
    "customserverid":<String_value>,
    "serverid":<Double_value>,
    "cka":<String_value>,
    "tcpb":<String_value>,
    "processlocal":<String_value>,
    "cmp":<String_value>,
    "maxbandwidth":<Double_value>,
    "accessdown":<String_value>,
    "svrstate":<String_value>,
    "delay":<Double_value>,
    "ipaddress":<String_value>,
    "monthreshold":<Double_value>,
    "monitor_state":<String_value>,
    "monstatcode":<Integer_value>,
    "lastresponse":<String_value>,
    "responsetime":<Double_value>,
    "riseapbrstatsmsgcode":<Integer_value>,
    "riseapbrstatsmsgcode2":<Integer_value>,
    "monstatparam1":<Integer_value>,
    "monstatparam2":<Integer_value>,
    "monstatparam3":<Integer_value>,
    "downstateflush":<String_value>,
    "statechangetimesec":<String_value>,
    "statechangetimemsec":<Double_value>,
    "timesincelaststatechange":<Double_value>,
    "tickssincelaststatechange":<Double_value>,
    "stateupdatereason":<Double_value>,
    "clmonowner":<Double_value>,
    "clmonview":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "hashid":<Double_value>,
    "graceful":<String_value>,
    "comment":<String_value>,
    "healthmonitor":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "serviceipstr":<String_value>,
    "weight":<Double_value>,
    "td":<Double_value>,
    "oracleserverversion":<String_value>,
    "dnsprofilename":<String_value>
}]}
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/service/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/service/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/service/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "service": [ {
name:<String_value>,all:<Boolean_value>,Internal:<Boolean_value>      "numofconnections":<Integer_value>,
    "servername":<String_value>,
    "policyname":<String_value>,
    "servicetype":<String_value>,
    "serviceconftpye":<Boolean_value>,
    "serviceconftype":<Boolean_value>,
    "serviceconftype2":<String_value>,
    "port":<Integer_value>,
    "value":<String_value>,
    "cleartextport":<Integer_value>,
    "gslb":<String_value>,
    "cachetype":<String_value>,
    "maxclient":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "maxreq":<Double_value>,
    "cacheable":<String_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "usip":<String_value>,
    "pathmonitor":<String_value>,
    "pathmonitorindv":<String_value>,
    "useproxyport":<String_value>,
    "sc":<String_value>,
    "dup_state":<String_value>,
    "sp":<String_value>,
    "rtspsessionidremap":<String_value>,
    "clttimeout":<Double_value>,
    "svrtimeout":<Double_value>,
    "publicip":<String_value>,
    "publicport":<Integer_value>,
    "customserverid":<String_value>,
    "serverid":<Double_value>,
    "cka":<String_value>,
    "tcpb":<String_value>,
    "processlocal":<String_value>,
    "cmp":<String_value>,
    "maxbandwidth":<Double_value>,
    "accessdown":<String_value>,
    "svrstate":<String_value>,
    "delay":<Double_value>,
    "ipaddress":<String_value>,
    "monthreshold":<Double_value>,
    "monitor_state":<String_value>,
    "monstatcode":<Integer_value>,
    "lastresponse":<String_value>,
    "responsetime":<Double_value>,
    "riseapbrstatsmsgcode":<Integer_value>,
    "riseapbrstatsmsgcode2":<Integer_value>,
    "monstatparam1":<Integer_value>,
    "monstatparam2":<Integer_value>,
    "monstatparam3":<Integer_value>,
    "downstateflush":<String_value>,
    "statechangetimesec":<String_value>,
    "statechangetimemsec":<Double_value>,
    "timesincelaststatechange":<Double_value>,
    "tickssincelaststatechange":<Double_value>,
    "stateupdatereason":<Double_value>,
    "clmonowner":<Double_value>,
    "clmonview":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "hashid":<Double_value>,
    "graceful":<String_value>,
    "comment":<String_value>,
    "healthmonitor":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "serviceipstr":<String_value>,
    "weight":<Double_value>,
    "td":<Double_value>,
    "oracleserverversion":<String_value>,
    "dnsprofilename":<String_value>
}]}
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/service?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "service": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

servicegroup

May 20, 2016

Configuration for service group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Name of the service group. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the name is created. Minimum length = 1
servicetype	<String>	Read-write	Protocol used to exchange data with the service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX
cachetype	<String>	Read-write	Cache type supported by the cache server. Possible values = TRANSPARENT, REVERSE, FORWARD
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
maxclient	<Double>	Read-write	Maximum number of simultaneous open connections for the service group. Minimum value = 0 Maximum value = 4294967294
maxreq	<Double>	Read-write	Maximum number of requests that can be sent on a persistent connection to the service group. Note: Connection requests beyond this value are rejected. Minimum value = 0 Maximum value = 65535
cacheable	<String>	Read-write	Use the transparent cache redirection virtual server to forward the request to the cache server. Note: Do not set this parameter if you set the Cache Type. Default value: NO Possible values = YES, NO
cip	<String>	Read-write	Insert the Client IP header in requests forwarded to the service. Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cipheader	<String>	Read-write	Name of the HTTP header whose value must be set to the IP address of the client. Used with the Client IP parameter. If client IP insertion is enabled, and the client IP header is not specified, the value of Client IP Header parameter or the value set by the set ns config command is used as client's IP header name. Minimum length = 1
usip	<String>	Read-write	Use client's IP address as the source IP address when initiating connection to the server. With the NO setting, which is the default, a mapped IP (MIP) address or subnet IP (SNIP) address is used as the source IP address to initiate server side connections. Possible values = YES, NO
pathmonitor	<String>	Read-write	Path monitoring for clustering. Possible values = YES, NO
pathmonitorindv	<String>	Read-write	Individual Path monitoring decisions. Possible values = YES, NO
useproxyport	<String>	Read-write	Use the proxy port as the source port when initiating connections with the server. With the NO setting, the client-side connection port is used as the source port for the server-side connection. Note: This parameter is available only when the Use Source IP (USIP) parameter is set to YES. Possible values = YES, NO
healthmonitor	<String>	Read-write	Monitor the health of this service. Available settings function as follows: YES - Send probes to check the health of the service. NO - Do not send probes to check the health of the service. With the NO option, the appliance shows the service as UP at all times. Default value: YES Possible values = YES, NO
sc	<String>	Read-write	State of the SureConnect feature for the service group. Default value: OFF Possible values = ON, OFF
sp	<String>	Read-write	Enable surge protection for the service group. Default value: OFF Possible values = ON, OFF
rtspsessionidremap	<String>	Read-write	Enable RTSP session ID mapping for the service group. Default value: OFF Possible values = ON, OFF
clttimeout	<Double>	Read-write	Time, in seconds, after which to terminate an idle client connection. Minimum value = 0 Maximum value = 31536000
svrtimeout	<Double>	Read-write	Time, in seconds, after which to terminate an idle server connection. Minimum value = 0 Maximum value = 31536000
cka	<String>	Read-write	Enable client keep-alive for the service group. Possible values = YES, NO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
tcp	<String>	Read-write	Enable TCP buffering for the service group. Possible values = YES, NO
cmp	<String>	Read-write	Enable compression for the specified service. Possible values = YES, NO
maxbandwidth	<Double>	Read-write	Maximum bandwidth, in Kbps, allocated for all the services in the service group. Minimum value = 0 Maximum value = 4294967287
monthreshold	<Double>	Read-write	Minimum sum of weights of the monitors that are bound to this service. Used to determine whether to mark a service as UP or DOWN. Minimum value = 0 Maximum value = 65535
state	<String>	Read-write	Initial state of the service group. Default value: ENABLED Possible values = ENABLED, DISABLED
downstateflush	<String>	Read-write	Flush all active transactions associated with all the services in the service group whose state transitions from UP to DOWN. Do not enable this option for applications that must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED
tcpprofilename	<String>	Read-write	Name of the TCP profile that contains TCP configuration settings for the service group. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the HTTP profile that contains HTTP configuration settings for the service group. Minimum length = 1 Maximum length = 127
comment	<String>	Read-write	Any information about the service group.
appflowlog	<String>	Read-write	Enable logging of AppFlow information for the specified service group. Default value: ENABLED Possible values = ENABLED, DISABLED
netprofile	<String>	Read-write	Network profile for the service group. Minimum length = 1 Maximum length = 127
autoscale	<String>	Read-write	Auto scale option for a servicegroup. Default value: DISABLED Possible values = DISABLED, DNS, POLICY
memberport	<Integer>	Read-write	member port.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
port	<Integer>	Read-write	Server port number. Range 1 - 65535
weight	<Double>	Read-write	Weight to assign to the servers in the service group. Specifies the capacity of the servers relative to the other servers in the load balancing configuration. The higher the weight, the higher the percentage of requests sent to the service. Minimum value = 1 Maximum value = 100
customserverid	<String>	Read-write	The identifier for this IP:Port pair. Used when the persistency type is set to Custom Server ID. Default value: "None"
serverid	<Double>	Read-write	The identifier for the service. This is used when the persistency type is set to Custom Server ID.
hashid	<Double>	Read-write	The hash identifier for the service. This must be unique for each service. This parameter is used by hash based load balancing methods. Minimum value = 1
monitor_name_svc	<String>	Read-write	Name of the monitor bound to the service group. Used to assign a weight to the monitor. Minimum length = 1
dup_weight	<Double>	Read-write	weight of the monitor that is bound to servicegroup. Minimum value = 1
riseapbrstatsmsgcode	<Integer>	Read-write	The code indicating the rise apbr status.
delay	<Double>	Read-write	Time, in seconds, allocated for a shutdown of the services in the service group. During this period, new requests are sent to the service only for clients who already have persistent sessions on the appliance. Requests from new clients are load balanced among other available services. After the delay time expires, no requests are sent to the service, and the service is marked as unavailable (OUT OF SERVICE).
graceful	<String>	Read-write	Wait for all existing connections to the service to terminate before shutting down the service. Default value: NO Possible values = YES, NO
includemembers	<Boolean>	Read-write	Display the members of the listed service groups in addition to their settings. Can be specified when no service group name is provided in the command. In that case, the details displayed for each service group are identical to the details displayed when a service group name is provided, except that bound monitors are not displayed.
newname	<String>	Read-write	New name for the service group. Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
servicegroupconftype	<Boolean>	Read-only	The configuration type of the service group.
value	<String>	Read-only	SSL Status. Possible values = Certkey not bound, SSL feature disabled
svrstate	<String>	Read-only	The state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
ip	<String>	Read-only	IP Address.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
statechangetimemsec	<Double>	Read-only	Time when last state change occurred. Milliseconds part.
stateupdatereason	<Double>	Read-only	Checks state update reason on the secondary node.
clmonowner	<Double>	Read-only	Tells the mon owner of the service.
clmonview	<Double>	Read-only	Tells the view id of the monitoring owner.
groupcount	<Double>	Read-only	Servicegroup Count.
riseapbrstatsmsgcode2	<Integer>	Read-only	The code indicating other rise stats.
serviceipstr	<String>	Read-only	This field has been intorduced to show the dbs services ip.
servicegroupeffectivestate	<String>	Read-only	Indicates the effective servicegroup state based on the state of the bound service items.If all services are UP the effective state is UP, if all are DOWN its DOWN,if all are OFS its OFS.If atleast one serviceis UP and rest are either DOWN or OFS, the effective state is PARTIAL-UP.If atleast one bound service is DOWN and rest are OFS the effective state is PARTIAL DOWN. Possible values = UP, DOWN, OUT OF SERVICE, PARTIAL-UP, PARTIAL-DOWN

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"servicegroup":{
    "servicegroupname":<String_value>,
    "servicetype":<String_value>,
    "cachetype":<String_value>,
    "td":<Double_value>,
    "maxclient":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "maxreq":<Double_value>,
    "cacheable":<String_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "usip":<String_value>,
    "pathmonitor":<String_value>,
    "pathmonitorindv":<String_value>,
    "useproxyport":<String_value>,
    "healthmonitor":<String_value>,
    "sc":<String_value>,
    "sp":<String_value>,
    "rtspsessionidremap":<String_value>,
    "clttimeout":<Double_value>,
    "svrtimeout":<Double_value>,
    "cka":<String_value>,
    "tcpb":<String_value>,
    "cmp":<String_value>,
    "maxbandwidth":<Double_value>,
    "monthreshold":<Double_value>,
    "state":<String_value>,
    "downstateflush":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "autoscale":<String_value>,
    "memberport":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup/servicegroupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"servicegroup":{
    "servicegroupname":<String_value>,
    "servername":<String_value>,
    "port":<Integer_value>,
    "weight":<Double_value>,
    "customserverid":<String_value>,
    "serverid":<Double_value>,
    "hashid":<Double_value>,
    "monitor_name_svc":<String_value>,
    "dup_weight":<Double_value>,
    "maxclient":<Double_value>,
    "maxreq":<Double_value>,
    "healthmonitor":<String_value>,
    "cacheable":<String_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "usip":<String_value>,
    "pathmonitor":<String_value>,
    "pathmonitorindv":<String_value>,
    "useproxyport":<String_value>,
    "sc":<String_value>,
    "sp":<String_value>,
    "rtspsessionidremap":<String_value>,
    "clttimeout":<Double_value>,
    "svrtimeout":<Double_value>,
    "cka":<String_value>,
    "tcpb":<String_value>,
    "cmp":<String_value>,
    "maxbandwidth":<Double_value>,
    "monthreshold":<Double_value>,
    "downstateflush":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"servicegroup":{
    "servicegroupname":<String_value>,
    "servername":true,
    "port":true,
    "weight":true,
    "customserverid":true,
    "serverid":true,
    "hashid":true,
    "riseapbrstatsmsgcode":true,
    "maxclient":true,
    "maxreq":true,
    "cacheable":true,
    "cip":true,
    "usip":true,
    "useproxyport":true,
    "sc":true,
    "sp":true,
    "rtspsessionidremap":true,
    "clttimeout":true,
    "svrtimeout":true,
    "cka":true,
    "tcpb":true,
    "cmp":true,
    "maxbandwidth":true,
    "monthreshold":true,
    "tcpprofilename":true,
    "httpprofilename":true,
    "appflowlog":true,
    "netprofile":true,
    "monitor_name_svc":true,
    "dup_weight":true,
    "healthmonitor":true,
    "cipheader":true,
    "pathmonitor":true,
    "pathmonitorindv":true,
    "downstateflush":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"servicegroup":{
    "servicegroupname":<String_value>,
    "servername":<String_value>,
    "port":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"servicegroup":{
    "servicegroupname":<String_value>,
    "servername":<String_value>,
    "port":<Integer_value>,
    "delay":<Double_value>,
    "graceful":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"servicegroup":{
    "servicegroupname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/servicegroup?**args=servicegroupname:<String_value>,includemembers:<Boolean_value>**

Use this query-parameter to get servicegroup resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/servicegroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/servicegroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of servicegroup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/servicegroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/servicegroup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the servicegroup resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroup": [ {
servicegroupname:<String_value>,includemembers:<Boolean_value>      "numofconnections":<Integer_value>,
      "servicetype":<String_value>,
      "port":<Integer_value>,
      "td":<Double_value>,
      "serviceconftpye":<Boolean_value>,
      "serviceconftype":<Boolean_value>,
      "value":<String_value>,
      "cachetype":<String_value>,
      "maxclient":<Double_value>,
      "maxreq":<Double_value>,
      "cacheable":<String_value>,
      "cip":<String_value>,
      "cipheader":<String_value>,
      "usip":<String_value>,
      "pathmonitor":<String_value>,
      "pathmonitorindv":<String_value>,
      "useproxyport":<String_value>,
      "sc":<String_value>,
      "sp":<String_value>,
      "rtspsessionidremap":<String_value>,
      "clttimeout":<Double_value>,
      "svrtimeout":<Double_value>,
      "cka":<String_value>,
      "tcpb":<String_value>,
      "cmp":<String_value>,
      "maxbandwidth":<Double_value>,
      "state":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "svrstate":<String_value>,
    "delay":<Double_value>,
    "ip":<String_value>,
    "servername":<String_value>,
    "monthreshold":<Double_value>,
    "weight":<Double_value>,
    "customserverid":<String_value>,
    "serverid":<Double_value>,
    "monstatcode":<Integer_value>,
    "monstatparam1":<Integer_value>,
    "monstatparam2":<Integer_value>,
    "monstatparam3":<Integer_value>,
    "downstateflush":<String_value>,
    "statechangetimemsec":<Double_value>,
    "timesincelaststatechange":<Double_value>,
    "stateupdatereason":<Double_value>,
    "clmonowner":<Double_value>,
    "clmonview":<Double_value>,
    "groupcount":<Double_value>,
    "comment":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "hashid":<Double_value>,
    "riseapbrstatsmsgcode":<Integer_value>,
    "riseapbrstatsmsgcode2":<Integer_value>,
    "graceful":<String_value>,
    "healthmonitor":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "autoscale":<String_value>,
    "memberport":<Integer_value>,
    "serviceipstr":<String_value>,
    "servicegroupeffectivestate":<String_value>
}]}
```


get

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup/servicegroupname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/servicegroup/servicegroupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/servicegroup/servicegroupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "servicegroup": [ {
servicegroupname:<String_value>,includemembers:<Boolean_value>      "numofconnections":<Integer_value>,
      "servicetype":<String_value>,
      "port":<Integer_value>,
      "td":<Double_value>,
      "serviceconftpye":<Boolean_value>,
      "serviceconftype":<Boolean_value>,
      "value":<String_value>,
      "cachetype":<String_value>,
      "maxclient":<Double_value>,
      "maxreq":<Double_value>,
      "cacheable":<String_value>,
      "cip":<String_value>,
      "cipheader":<String_value>,
      "usip":<String_value>,
      "pathmonitor":<String_value>,
      "pathmonitorindv":<String_value>,
      "useproxyport":<String_value>,
      "sc":<String_value>,
      "sp":<String_value>,
      "rtspsessionidremap":<String_value>,
      "clttimeout":<Double_value>,
      "svrtimeout":<Double_value>,
      "cka":<String_value>,
      "tcpb":<String_value>,
      "cmp":<String_value>,
      "maxbandwidth":<Double_value>,
      "state":<String_value>,
      "svrstate":<String_value>,
      "delay":<Double_value>,
      "ip":<String_value>,
      "servername":<String_value>,
      "monthreshold":<Double_value>,
      "weight":<Double_value>,
      "customserverid":<String_value>,
      "serverid":<Double_value>,
      "monstatcode":<Integer_value>,
      "monstatparam1":<Integer_value>,
      "monstatparam2":<Integer_value>,
      "monstatparam3":<Integer_value>,
      "downstateflush":<String_value>,
      "statechangetimemsec":<Double_value>,
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "timesincelaststatechange":<Double_value>,
    "stateupdatereason":<Double_value>,
    "clmonowner":<Double_value>,
    "clmonview":<Double_value>,
    "groupcount":<Double_value>,
    "comment":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "hashid":<Double_value>,
    "riseapbrstatsmsgcode":<Integer_value>,
    "riseapbrstatsmsgcode2":<Integer_value>,
    "graceful":<String_value>,
    "healthmonitor":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "autoscale":<String_value>,
    "memberport":<Integer_value>,
    "serviceipstr":<String_value>,
    "servicegroupeffectivestate":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroup": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

servicegroupbindings

May 20, 2016

Configuration for servicegroupbind resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	The name of the service. Minimum length = 1
ipaddress	<String>	Read-only	The IP address of the vserver.
port	<Integer>	Read-only	The port of the vserver. Range 1 - 65535
state	<String>	Read-only	The state of the service group. Possible values = ENABLED, DISABLED
svrstate	<String>	Read-only	The state of the vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
vservername	<String>	Read-only	The name of the vserver.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroupbindings

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/servicegroupbindings?**args=servicegroupname:<String_value>,**

Use this query-parameter to get servicegroupbindings resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/servicegroupbindings?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/servicegroupbindings?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of servicegroupbindings resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/servicegroupbindings?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/servicegroupbindings?**pagesize=#no&pageno=#no**

Use this query-parameter to get the servicegroupbindings resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroupbindings": [ {
servicegroupname:<String_value>,      "ipaddress":<String_value>,
      "port":<Integer_value>,
      "state":<String_value>,
      "svrstate":<String_value>,
      "vservername":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroupbindings/servicegroupname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/servicegroupbindings/servicegroupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/servicegroupbindings/servicegroupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroupbindings": [ {
servicegroupname:<String_value>,      "ipaddress":<String_value>,
      "port":<Integer_value>,
      "state":<String_value>,
      "svrstate":<String_value>,
      "vservername":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroupbindings?**args=servicegroupname:<String_value>,&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroupbindings": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

svcbindings

Jul 07, 2017

Configuration for service bindings resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	The name of the service. Minimum length = 1
ipaddress	<String>	Read-only	The IP address of the vserver.
port	<Integer>	Read-only	The port of the vserver. Range 1 - 65535
svrstate	<String>	Read-only	The state of the vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
vservername	<String>	Read-only	The name of the vserver.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/svcbindings

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/svcbindings?**args=servicename:<String_value>,**

Use this query-parameter to get svcbindings resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/svcbindings?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/svcbindings?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of svcbindings resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/svcbindings?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/svcbindings?**pagesize=#no&pageno=#no**

Use this query-parameter to get the svcbindings resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "svcbindings": [ {
servicename:<String_value>,      "ipaddress":<String_value>,
    "port":<Integer_value>,
    "svrstate":<String_value>,
    "vservername":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/svcbindings/servicename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/svcbindings/servicename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/svcbindings/servicename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "svcbindings": [ {
servicename:<String_value>,      "ipaddress":<String_value>,
    "port":<Integer_value>,
    "svrstate":<String_value>,
    "vservername":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/svcbindings?**args=servicename:<String_value>,&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "svcbindings": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

servicegroup_binding

May 20, 2016

Binding object which returns the resources bound to servicegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
servicegroup_servicegroupentitymonbindings_binding	<servicegroup_servicegroupentitymonbindings_binding[]>	Read-only	servicegroupentitymonbindings that can be bound to servicegroup.
servicegroup_lbmonitor_binding	<servicegroup_lbmonitor_binding[]>	Read-only	lbmonitor that can be bound to servicegroup.
servicegroup_servicegroupmember_binding	<servicegroup_servicegroupmember_binding[]>	Read-only	servicegroupmember that can be bound to servicegroup.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup_binding/servicegroupname_value<String>

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "servicegroup_binding": [ {
      "servicegroupname":<String_value>,
      "servicegroup_lbmonitor_binding":<servicegroup_lbmonitor_binding[]_value>,
      "servicegroup_servicegroupmember_binding":<servicegroup_servicegroupmember_binding[]_value>,
      "servicegroup_servicegroupentitymonbindings_binding":<servicegroup_servicegroupentitymonbindings_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

servicegroup_lbmonitor_binding

May 20, 2016

Binding object showing the lbmonitor that can be bound to servicegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
port	<Integer>	Read-write	Port number of the service. Each service must have a unique port number. Range 1 - 65535
state	<String>	Read-write	Initial state of the service after binding. Default value: ENABLED Possible values = ENABLED, DISABLED
hashid	<Double>	Read-write	Unique numerical identifier used by hash based load balancing methods to identify a service. Minimum value = 1
serverid	<Double>	Read-write	The identifier for the service. This is used when the persistency type is set to Custom Server ID.
customserverid	<String>	Read-write	Unique service identifier. Used when the persistency type for the virtual server is set to Custom Server ID. Default value: "None"
weight	<Double>	Read-write	Weight to assign to the servers in the service group. Specifies the capacity of the servers relative to the other servers in the load balancing configuration. The higher the weight, the higher the percentage of requests sent to the service. Minimum value = 1 Maximum value = 100
monitor_name	<String>	Read-write	Monitor name.
passive	<Boolean>	Read-write	Indicates if load monitor is passive. A passive load monitor does not remove service from LB decision when threshold is breached.
monstate	<String>	Read-write	Monitor state. Possible values = ENABLED, DISABLED
monweight	<Double>	Read-only	weight of the monitor that is bound to servicegroup.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/servicegroup_lbmonitor_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"servicegroup_lbmonitor_binding":{
    "servicegroupname":<String_value>,
    "port":<Integer_value>,
    "monitor_name":<String_value>,
    "monstate":<String_value>,
    "passive":<Boolean_value>,
    "weight":<Double_value>,
    "customserverid":<String_value>,
    "serverid":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "state":<String_value>,
    "hashid":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of servicegroup_lbmonitor_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the servicegroup_lbmonitor_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroup_lbmonitor_binding": [ {
    "servicegroupname":<String_value>,
    "port":<Integer_value>,
    "state":<String_value>,
    "hashid":<Double_value>,
    "serverid":<Double_value>,
    "customserverid":<String_value>,
    "weight":<Double_value>,
    "monitor_name":<String_value>,
    "passive":<Boolean_value>,
    "monstate":<String_value>,
    "monweight":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup_lbmonitor_binding/servicegroupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"servicegroup_lbmonitor_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

servicegroup_servicegroupentitymonbindings_binding

May 20, 2016

Binding object showing the servicegroupentitymonbindings that can be bound to servicegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
servicegroupentname2	<String>	Read-write	.
port	<Integer>	Read-write	Port number of the service. Each service must have a unique port number. Range 1 - 65535
state	<String>	Read-write	Initial state of the service after binding. Default value: ENABLED Possible values = ENABLED, DISABLED
hashid	<Double>	Read-write	Unique numerical identifier used by hash based load balancing methods to identify a service. Minimum value = 1
serverid	<Double>	Read-write	The identifier for the service. This is used when the persistency type is set to Custom Server ID.
customserverid	<String>	Read-write	Unique service identifier. Used when the persistency type for the virtual server is set to Custom Server ID. Default value: "None"
weight	<Double>	Read-write	. Default value: 1 Minimum value = 1 Maximum value = 100
monitor_name	<String>	Read-write	Monitor name.
passive	<Boolean>	Read-write	Indicates if load monitor is passive. A passive load monitor does not remove service from LB decision when threshold is breached.
lastresponse	<String>	Read-only	The string form of monstatcode.
monitor_state	<String>	Read-only	The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monitortotalprobes	<Double>	Read-only	Total number of probes sent to monitor this service.
monitortotalfailedprobes	<Double>	Read-only	Total number of failed probes.
monitorcurrentfailedprobes	<Double>	Read-only	Total number of currently failed probes.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/servicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of servicegroup_servicegroupentitymonbindings_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/servicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the servicegroup_servicegroupentitymonbindings_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroup_servicegroupentitymonbindings_binding": [ {
    "servicegroupname":<String_value>,
    "servicegroupentname2":<String_value>,
    "port":<Integer_value>,
    "state":<String_value>,
    "hashid":<Double_value>,
    "serverid":<Double_value>,
    "customserverid":<String_value>,
    "weight":<Double_value>,
    "monitor_name":<String_value>,
    "passive":<Boolean_value>,
    "lastresponse":<String_value>,
    "monitor_state":<String_value>,
    "monitortotalprobes":<Double_value>,
    "monitortotalfailedprobes":<Double_value>,
    "monitorcurrentfailedprobes":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup_servicegroupentitymonbindings_binding/servicegroupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"servicegroup_servicegroupentitymonbindings_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

servicegroup_servicegroupmember_binding

May 20, 2016

Binding object showing the servicegroupmember that can be bound to servicegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
ip	<String>	Read-write	IP Address.
port	<Integer>	Read-write	Server port number. Range 1 - 65535
state	<String>	Read-write	Initial state of the service group. Default value: ENABLED Possible values = ENABLED, DISABLED
hashid	<Double>	Read-write	The hash identifier for the service. This must be unique for each service. This parameter is used by hash based load balancing methods. Minimum value = 1
serverid	<Double>	Read-write	The identifier for the service. This is used when the persistency type is set to Custom Server ID.
servername	<String>	Read-write	Name of the server to which to bind the service group. Minimum length = 1
customserverid	<String>	Read-write	The identifier for this IP:Port pair. Used when the persistency type is set to Custom Server ID. Default value: "None"
weight	<Double>	Read-write	Weight to assign to the servers in the service group. Specifies the capacity of the servers relative to the other servers in the load balancing configuration. The higher the weight, the higher the percentage of requests sent to the service. Minimum value = 1 Maximum value = 100
delay	<Double>	Read-only	Time, in seconds, allocated for a shutdown of the services in the service group. During this period, new requests are sent to the service only for clients who already have persistent sessions on the appliance. Requests from new clients are load balanced among other available services. After the delay time expires, no requests are sent to the service, and the service is marked as unavailable (OUT OF SERVICE).
statechangetimesec	<String>	Read-only	Time when last state change occurred. Seconds part.
svrstate	<String>	Read-only	The state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
graceful	<String>	Read-only	Wait for all existing connections to the service to terminate before shutting down the service. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/servicegroup_servicegroupmember_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"servicegroup_servicegroupmember_binding":{
    "servicegroupname":<String_value>,
    "ip":<String_value>,
    "servername":<String_value>,
    "port":<Integer_value>,
    "weight":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "customserverid":<String_value>,
    "serverid":<Double_value>,
    "state":<String_value>,
    "hashid":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of servicegroup_servicegroupmember_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the servicegroup_servicegroupmember_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroup_servicegroupmember_binding": [ {
    "servicegroupname":<String_value>,
    "ip":<String_value>,
    "port":<Integer_value>,
    "state":<String_value>,
    "hashid":<Double_value>,
    "serverid":<Double_value>,
    "servername":<String_value>,
    "customserverid":<String_value>,
    "weight":<Double_value>,
    "delay":<Double_value>,
    "statechangetimesec":<String_value>,
    "svrstate":<String_value>,
    "tickssincelaststatechange":<Double_value>,
    "graceful":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/servicegroup_servicegroupmember_binding/servicegroupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"servicegroup_servicegroupmember_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

service_binding

May 20, 2016

Binding object which returns the resources bound to service.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the service for which to display configuration details. Minimum length = 1
service_lbmonitor_binding	<service_lbmonitor_binding[]>	Read-only	lbmonitor that can be bound to service.
service_dospolicy_binding	<service_dospolicy_binding[]>	Read-only	dospolicy that can be bound to service.
service_scpolicy_binding	<service_scpolicy_binding[]>	Read-only	scpolicy that can be bound to service.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/service_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "service_binding": [ {
    "name":<String_value>,
    "service_scpolicy_binding":<service_scpolicy_binding[]_value>,
    "service_lbmonitor_binding":<service_lbmonitor_binding[]_value>,
    "service_dospolicy_binding":<service_dospolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

service_dospolicy_binding

May 20, 2016

Binding object showing the dospolicy that can be bound to service.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	The name of the policyname for which this service is bound.
name	<String>	Read-write	Name of the service to which to bind a policy or monitor. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/service_dospolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"service_dospolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/service_dospolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/service_dospolicy_binding/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/service_dospolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of service_dospolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/service_dospolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the service_dospolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "service_dospolicy_binding": [ {
    "policyname":<String_value>,
    "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/service_dospolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"service_dospolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

service_lbmonitor_binding

May 20, 2016

Binding object showing the lbmonitor that can be bound to service.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
weight	<Double>	Read-write	Weight to assign to the monitor-service binding. When a monitor is UP, the weight assigned to its binding with the service determines how much the monitor contributes toward keeping the health of the service above the value configured for the Monitor Threshold parameter. Minimum value = 1 Maximum value = 100
name	<String>	Read-write	Name of the service to which to bind a policy or monitor. Minimum length = 1
passive	<Boolean>	Read-write	Indicates if load monitor is passive. A passive load monitor does not remove service from LB decision when threshold is breached.
monitor_name	<String>	Read-write	The monitor Names.
monstate	<String>	Read-write	The configured state (enable/disable) of the monitor on this server. Possible values = ENABLED, DISABLED
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
dup_weight	<Double>	Read-only	The weight of the monitor.
responsetime	<Double>	Read-only	Response time of this monitor.
totalfailedprobes	<Double>	Read-only	The total number of failed probs.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
lastresponse	<String>	Read-only	The string form of monstatcode.
failedprobes	<Double>	Read-only	Number of the current failed monitoring probes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
totalprobes	<Double>	Read-only	The total number of probs sent.
monitor_state	<String>	Read-only	The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
dup_state	<String>	Read-only	Added this field for getting state value from table. Possible values = ENABLED, DISABLED
monitortotalprobes	<Double>	Read-only	Total number of probes sent to monitor this service.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monitortotalfailedprobes	<Double>	Read-only	Total number of failed probes.
monitorcurrentfailedprobes	<Double>	Read-only	Total number of currently failed probes.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/service_lbmonitor_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"service_lbmonitor_binding":{
    "name":<String_value>,
    "monitor_name":<String_value>,
    "monstate":<String_value>,
    "weight":<Double_value>,
    "passive":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/service_lbmonitor_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/service_lbmonitor_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/service_lbmonitor_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of service_lbmonitor_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/service_lbmonitor_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the service_lbmonitor_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "service_lbmonitor_binding": [ {
    "weight":<Double_value>,
    "name":<String_value>,
    "passive":<Boolean_value>,
    "monitor_name":<String_value>,
    "monstate":<String_value>,
    "monstatcode":<Integer_value>,
    "dup_weight":<Double_value>,
    "responsetime":<Double_value>,
    "totalfailedprobes":<Double_value>,
    "monstatparam2":<Integer_value>,
    "lastresponse":<String_value>,
    "failedprobes":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "monstatparam3":<Integer_value>,
    "totalprobes":<Double_value>,
    "monitor_state":<String_value>,
    "dup_state":<String_value>,
    "monitortotalprobes":<Double_value>,
    "monstatparam1":<Integer_value>,
    "monitortotalfailedprobes":<Double_value>,
    "monitorcurrentfailedprobes":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/service_lbmonitor_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"service_lbmonitor_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

service_scpolicy_binding

May 20, 2016

Binding object showing the scpolicy that can be bound to service.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	The name of the policyname for which this service is bound.
name	<String>	Read-write	Name of the service to which to bind a policy or monitor. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/service_scpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"service_scpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/service_scpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/service_scpolicy_binding/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/service_scpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of service_scpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/service_scpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the service_scpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "service_scpolicy_binding": [ {
      "policyname":<String_value>,
      "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/service_scpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"service_scpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vserver

Jul 07, 2017

Configuration for virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the virtual server to be removed. Minimum length = 1
backupvserver	<String>	Read-write	The name of the backup virtual server for this virtual server. Minimum length = 1
redirecturl	<String>	Read-write	The URL where traffic is redirected if the virtual server in the system becomes unavailable. Minimum length = 1
cacheable	<String>	Read-write	Use this option to specify whether a virtual server (used for load balancing or content switching) routes requests to the cache redirection virtual server before sending it to the configured servers. Possible values = YES, NO
clttimeout	<Double>	Read-write	The timeout value in seconds for idle client connection. Minimum value = 0 Maximum value = 31536000
somethod	<String>	Read-write	The spillover factor. The system will use this value to determine if it should send traffic to the backupvserver when the main virtual server reaches the spillover threshold. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sopersistence	<String>	Read-write	The state of the spillover persistence. Default value: DISABLED Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-write	The spillover persistence entry timeout. Default value: 2 Minimum value = 2 Maximum value = 1440
sothreshold	<Double>	Read-write	The spillver threshold value. Minimum value = 1 Maximum value = 4294967294
			The lb vserver of type PUSH/SSL_PUSH to which server pushes the updates

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1

Operations (click to see Properties)

DELETE | UPDATE | ENABLE | DISABLE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vserver/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vserver

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vserver":{
    "name":<String_value>,
    "backupvserver":<String_value>,
    "redirecturl":<String_value>,
    "cacheable":<String_value>,
    "clttimeout":<Double_value>,
    "somethod":<String_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "sothreshold":<Double_value>,
    "pushvserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/vserver?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/vserver?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Ca

May 20, 2016

Content Accelerator Configuration. The Content Accelerator feature is used to cache static and dynamic web application data.

Name	Description
caaction	Configuration for ca action
caglobal_binding	Binding object showing the resources that can be bound to caglobal
caglobal_capolicy_binding	Binding object showing the capolicy that can be bound to caglobal
capolicy	Configuration for contentaccelerator policy
capolicy_binding	Binding object showing the resources that can be bound to capolicy
capolicy_caglobal_binding	Binding object showing the caglobal that can be bound to capolicy
capolicy_csvserver_binding	Binding object showing the csvserver that can be bound to capolicy
capolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to capolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

caaction

May 20, 2016

Configuration for ca action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the content accelerator action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters.
accumressize	<Double>	Read-write	Size of the data, in KB, that the server must respond with. The NetScaler uses this data to compute a hash which is then used to lookup within the T2100 appliance.
lbvserver	<String>	Read-write	Name of the load balancing virtual server that has the T2100 appliances as services.
comment	<String>	Read-write	Information about the content accelerator action.
type	<String>	Read-write	Specifies whether the NetScaler must lookup for the response on the T2100 appliance or serve the response directly from the server. Possible values = nolookup, lookup, noop
newname	<String>	Read-write	New name for the content accelerator action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the content accelerator policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my content accelerator action" or ?my content accelerator action?).!,. Minimum length = 1
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default content accelerator action.
hits	<Double>	Read-only	The number of times the action has been taken.
builtin	<String[]>	Read-only	Flag to determine whether content accelerator action is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

Name	Data Type	Permissions	Description

ADD | UPDATE | UNSET | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/caaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"caaction":{
    "name":<String_value>,
    "accumressize":<Double_value>,
    "lbvserver":<String_value>,
    "comment":<String_value>,
    "type":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/caaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"caaction":{
    "name":<String_value>,
    "accumressize":<Double_value>,
    "type":<String_value>,
    "lbvserver":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/caaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"caaction":{
    "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "accumressize":true,
    "type":true,
    "lbvserver":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/caaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/caaction?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"caaction":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/caaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/caaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/caaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of caaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/caaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/caaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the caaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "caaction": [ {
    "name":<String_value>,
    "type":<String_value>,
    "accumressize":<Double_value>,
    "lbvserver":<String_value>,
    "isdefault":<Boolean_value>,
    "hits":<Double_value>,
    "builtin":<String[]_value>,
    "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/caaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/caaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/caaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "caaction": [ {
    "name":<String_value>,
    "type":<String_value>,
    "accumressize":<Double_value>,
    "lbvserver":<String_value>,
    "isdefault":<Boolean_value>,
    "hits":<Double_value>,
    "builtin":<String[]_value>,
    "comment":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/caaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "caaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

caglobal_binding

May 20, 2016

Binding object which returns the resources bound to caglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
caglobal_capolicy_binding	<caglobal_capolicy_binding[]>	Read-only	capolicy that can be bound to caglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/caglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "caglobal_binding": [ {
      "caglobal_capolicy_binding":<caglobal_capolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

caglobal_capolicy_binding

Jul 07, 2017

Binding object showing the capolicy that can be bound to caglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the content accelerator policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the content accelerator policy.
gotopriorityexpression	<String>	Read-write	.
type	<String>	Read-write	. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
numpol	<Double>	Read-only	Number of polices bound to label.
flowtype	<Double>	Read-only	flowtype of the bound content accelerator policy.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/caglobal_capolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"caglobal_capolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/caglobal_capolicy_binding

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/caglobal_capolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/caglobal_capolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of caglobal_capolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/caglobal_capolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the caglobal_capolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "caglobal_capolicy_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "policyname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "numpol":<Double_value>,
    "flowtype":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/caglobal_capolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "caglobal_capolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

capolicy

May 20, 2016

Configuration for content accelerator policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the content accelerator policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the policy is created.
rule	<String>	Read-write	Expression that determines which requests or responses match the content accelerator policy. When specifying the rule in the CLI, the description must be enclosed within double quotes.
action	<String>	Read-write	Name of content accelerator action to be executed when the rule is evaluated to true.
undefaction	<String>	Read-write	.
comment	<String>	Read-write	Information about the content accelerator policy.
logaction	<String>	Read-write	Name of messagelog action to use when a request matches this policy.
newname	<String>	Read-write	New name for the content accelerator policy. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of Undef hits.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default content accelerator policy.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"capolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"capolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "undefaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"capolicy":{
    "name":<String_value>,
    "comment":true,
    "logaction":true,
    "undefaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"capolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/capolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/capolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of capolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/capolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/capolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the capolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "capolicy": [ {
    "name":<String_value>,
    "undefaction":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "action":<String_value>,
    "rule":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "isdefault":<Boolean_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/capolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/capolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "capolicy": [ {
    "name":<String_value>,
    "undefaction":<String_value>,
    "action":<String_value>,
    "rule":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "isdefault":<Boolean_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "capolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

capolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to capolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the content accelerator policy to be displayed.
capolicy_lbvserver_binding	<capolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to capolicy.
capolicy_caglobal_binding	<capolicy_caglobal_binding[]>	Read-only	caglobal that can be bound to capolicy.
capolicy_csvserver_binding	<capolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to capolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "capolicy_binding": [ {
      "name":<String_value>,
      "capolicy_csvserver_binding":<capolicy_csvserver_binding[]_value>,
      "capolicy_caglobal_binding":<capolicy_caglobal_binding[]_value>,
      "capolicy_lbvserver_binding":<capolicy_lbvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

capolicy_caglobal_binding

May 20, 2016

Binding object showing the caglobal that can be bound to capolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the content accelerator policy to be displayed.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy_caglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/capolicy_caglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of capolicy_caglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/capolicy_caglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the capolicy_caglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "capolicy_caglobal_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy_caglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"capolicy_caglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

capolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to capolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the content accelerator policy to be displayed.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/capolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of capolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/capolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the capolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "capolicy_csvserver_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"capolicy_csvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

capolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to capolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the content accelerator policy to be displayed.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/capolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of capolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/capolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the capolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "capolicy_lbvserver_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/capolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"capolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Integrated Caching

Jul 07, 2017

Integrated Caching Configuration. The Integrated Caching feature is used to cache static and dynamic web application data.

Name	Description
cachecontentgroup	Configuration for Integrated Cache content group
cacheforwardproxy	Configuration for forward proxy
cacheglobal_binding	
cacheglobal_cachepolicy_binding	
cacheobject	Configuration for cache object
cacheparameter	Configuration for cache parameter
cachepolicy	Configuration for Integrated Cache policy
cachepolicy_binding	Binding object showing the resources that can be bound to cachepolicy
cachepolicy_cacheglobal_binding	Binding object showing the cacheglobal that can be bound to cachepolicy
cachepolicy_cachepolicylabel_binding	Binding object showing the cachepolicylabel that can be bound to cachepolicy
cachepolicy_csvserver_binding	Binding object showing the csvserver that can be bound to cachepolicy
cachepolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to cachepolicy
cachepolicylabel	Configuration for cache policy label
cachepolicylabel_binding	Binding object showing the resources that can be bound to cachepolicylabel
cachepolicylabel_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to cachepolicylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicylabel_policybinding_binding **Name**	Binding object showing the policybinding that can be bound to cachepolicylabel **Description**
cacheselector	Configuration for cache selector

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachecontentgroup

Jul 07, 2017

Configuration for Integrated Cache content group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the content group. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the content group is created. Minimum length = 1
weakposrelexpiry	<Double>	Read-write	Relative expiry time, in seconds, for expiring positive responses with response codes between 200 and 399. Cannot be used in combination with other Expiry attributes. Similar to -relExpiry but has lower precedence. Minimum value = 0 Maximum value = 31536000
heurexpiryparam	<Double>	Read-write	Heuristic expiry time, in percent of the duration, since the object was last modified. Minimum value = 0 Maximum value = 100
relexpiry	<Double>	Read-write	Relative expiry time, in seconds, after which to expire an object cached in this content group. Minimum value = 0 Maximum value = 31536000
relexpirymillisec	<Double>	Read-write	Relative expiry time, in milliseconds, after which to expire an object cached in this content group. Minimum value = 0 Maximum value = 86400000
absexpiry	<String[]>	Read-write	Local time, up to 4 times a day, at which all objects in the content group must expire. CLI Users: For example, to specify that the objects in the content group should expire by 11:00 PM, type the following command: add cache contentgroup <contentgroup name> -absexpiry 23:00 To specify that the objects in the content group should expire at 10:00 AM, 3 PM, 6 PM, and 11:00 PM, type: add cache contentgroup <contentgroup name> -absexpiry 10:00 15:00 18:00 23:00.
			Coordinated Universal Time (GMT), up to 4 times a day, when all objects in the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
weaknegrelexpiry	<Double>	Read-write	Relative expiry time, in seconds, for expiring negative responses. This value is used only if the expiry time cannot be determined from any other source. It is applicable only to the following status codes: 307, 403, 404, and 410. Minimum value = 0 Maximum value = 31536000
hitparams	<String[]>	Read-write	Parameters to use for parameterized hit evaluation of an object. Up to 128 parameters can be specified. Mutually exclusive with the Hit Selector parameter. Minimum length = 1
invalparams	<String[]>	Read-write	Parameters for parameterized invalidation of an object. You can specify up to 8 parameters. Mutually exclusive with invalSelector. Minimum length = 1
ignoreparamvaluecase	<String>	Read-write	Ignore case when comparing parameter values during parameterized hit evaluation. (Parameter value case is ignored by default during parameterized invalidation.). Possible values = YES, NO
matchcookies	<String>	Read-write	Evaluate for parameters in the cookie header also. Possible values = YES, NO
invalrestrictedtohost	<String>	Read-write	Take the host header into account during parameterized invalidation. Possible values = YES, NO
polleverytime	<String>	Read-write	Always poll for the objects in this content group. That is, retrieve the objects from the origin server whenever they are requested. Default value: NO Possible values = YES, NO
ignorereloadreq	<String>	Read-write	Ignore any request to reload a cached object from the origin server. To guard against Denial of Service attacks, set this parameter to YES. For RFC-compliant behavior, set it to NO. Default value: YES Possible values = YES, NO
removecookies	<String>	Read-write	Remove cookies from responses. Default value: YES Possible values = YES, NO
prefetch	<String>	Read-write	Attempt to refresh objects that are about to go stale. Default value: YES Possible values = YES, NO
prefetchperiod	<Double>	Read-write	Time period, in seconds before an object's calculated expiry time, during which to attempt prefetch. Minimum value = 0 Maximum value = 4294967294
prefetchperiodmillisec	<Double>	Read-write	Time period, in milliseconds before an object's calculated expiry time, during which to attempt prefetch. Minimum value = 0 Maximum value = 4294967290

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
prefetchmaxpending	<Double>	Read-write	Maximum number of outstanding prefetches that can be queued for the content group. Minimum value = 0 Maximum value = 4294967294
flashcache	<String>	Read-write	Perform flash cache. Mutually exclusive with Poll Every Time (PET) on the same content group. Default value: NO Possible values = YES, NO
expireatlastbyte	<String>	Read-write	Force expiration of the content immediately after the response is downloaded (upon receipt of the last byte of the response body). Applicable only to positive responses. Default value: NO Possible values = YES, NO
insertvia	<String>	Read-write	Insert a Via header into the response. Default value: YES Possible values = YES, NO
insertage	<String>	Read-write	Insert an Age header into the response. An Age header contains information about the age of the object, in seconds, as calculated by the integrated cache. Default value: YES Possible values = YES, NO
insertetag	<String>	Read-write	Insert an ETag header in the response. With ETag header insertion, the integrated cache does not serve full responses on repeat requests. Default value: YES Possible values = YES, NO
cachecontrol	<String>	Read-write	Insert a Cache-Control header into the response. Minimum length = 1
quickabortsize	<Double>	Read-write	If the size of an object that is being downloaded is less than or equal to the quick abort value, and a client aborts during the download, the cache stops downloading the response. If the object is larger than the quick abort size, the cache continues to download the response. Default value: 4194303 Minimum value = 0 Maximum value = 4194303
minressize	<Double>	Read-write	Minimum size of a response that can be cached in this content group. Default minimum response size is 0. Minimum value = 0 Maximum value = 2097151
maxressize	<Double>	Read-write	Maximum size of a response that can be cached in this content group. Default value: 80 Minimum value = 0 Maximum value = 2097151
memlimit	<Double>	Read-write	Maximum amount of memory that the cache can use. The effective limit is based on the available memory of the NetScaler appliance. Default value: 65536

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ignorereqcachinghdrs	<String>	Read-write	Ignore Cache-Control and Pragma headers in the incoming request. Default value: YES Possible values = YES, NO
minhits	<Integer>	Read-write	Number of hits that qualifies a response for storage in this content group. Default value: 0
alwaysevalpolicies	<String>	Read-write	Force policy evaluation for each response arriving from the origin server. Cannot be set to YES if the Prefetch parameter is also set to YES. Default value: NO Possible values = YES, NO
persistha	<String>	Read-write	Setting persistHA to YES causes IC to save objects in contentgroup to Secondary node in HA deployment. Default value: NO Possible values = YES, NO
pinned	<String>	Read-write	Do not flush objects from this content group under memory pressure. Default value: NO Possible values = YES, NO
lazydnsresolve	<String>	Read-write	Perform DNS resolution for responses only if the destination IP address in the request does not match the destination IP address of the cached response. Default value: YES Possible values = YES, NO
hitselector	<String>	Read-write	Selector for evaluating whether an object gets stored in a particular content group. A selector is an abstraction for a collection of PIXL expressions.
invalselector	<String>	Read-write	Selector for invalidating objects in the content group. A selector is an abstraction for a collection of PIXL expressions.
type	<String>	Read-write	The type of the content group. Default value: HTTP Possible values = HTTP, MYSQL, MSSQL
query	<String>	Read-write	Query string specifying individual objects to flush from this group by using parameterized invalidation. If this parameter is not set, all objects are flushed from the group. Minimum length = 1
host	<String>	Read-write	Flush only objects that belong to the specified host. Do not use except with parameterized invalidation. Also, the Invalidation Restricted to Host parameter for the group must be set to YES. Minimum length = 1
selectorvalue	<String>	Read-write	Value of the selector to be used for flushing objects from the content group. Requires that an invalidation selector be configured for the content group. Minimum length = 1
tosecondary	<String>	Read-write	content group whose objects are to be sent to secondary. Default value: NO Possible values = YES, NO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
prefetchcur	<Double>	Read-only	Current outstanding prefetches.
memusage	<Double>	Read-only	Current memory usage.
memdusage	<Double>	Read-only	Current disk memory usage.
disklimit	<Double>	Read-only	Maximum amount of disk that the cache can use. The effective limit is based on the available memory of the NetScaler appliance. Default value: 0
cachenon304hits	<Double>	Read-only	Cache non 304 hits.
cache304hits	<Double>	Read-only	Cache 304 hits.
cachecells	<Double>	Read-only	Number of cells.
cachegroupincarnation	<Double>	Read-only	Cache group incarnation.
persist	<String>	Read-only	Setting persist to YES causes IC to save objects in contentgroup to disk. Default value: NO Possible values = YES, NO
policyname	<String[]>	Read-only	Active cache policies refering to this group.
cachenuminvalpolicy	<Double>	Read-only	Number of active Invalidation policies refering to this group.
markercells	<Double>	Read-only	Numbers of marker cells in this group.
builtin	<String[]>	Read-only	. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | EXPIRE | FLUSH | SAVE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachecontentgroup":{
    "name":<String_value>,
    "weakposrelexpiry":<Double_value>,
    "heurexpiryparam":<Double_value>,
    "relexpiry":<Double_value>,
    "relexpirymillisec":<Double_value>,
    "absexpiry":<String[]_value>,
    "absexpirygmt":<String[]_value>,
    "weaknegrelexpiry":<Double_value>,
    "hitparams":<String[]_value>,
    "invalparams":<String[]_value>,
    "ignoreparamvaluecase":<String_value>,
    "matchcookies":<String_value>,
    "invalrestrictedtohost":<String_value>,
    "polleverytime":<String_value>,
    "ignorereloadreq":<String_value>,
    "removecookies":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "prefetch":<String_value>,
    "prefetchperiod":<Double_value>,
    "prefetchperiodmillisec":<Double_value>,
    "prefetchmaxpending":<Double_value>,
    "flashcache":<String_value>,
    "expireatlastbyte":<String_value>,
    "insertvia":<String_value>,
    "insertage":<String_value>,
    "insertetag":<String_value>,
    "cachecontrol":<String_value>,
    "quickabortsize":<Double_value>,
    "minressize":<Double_value>,
    "maxressize":<Double_value>,
    "memlimit":<Double_value>,
    "ignorereqcachinghdrs":<String_value>,
    "minhits":<Integer_value>,
    "alwaysevalpolicies":<String_value>,
    "persistha":<String_value>,
    "pinned":<String_value>,
    "lazydnsresolve":<String_value>,
    "hitselector":<String_value>,
    "invalselector":<String_value>,
    "type":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachecontentgroup":{
    "name":<String_value>,
    "weakposrelexpiry":<Double_value>,
    "heurexpiryparam":<Double_value>,
    "relexpiry":<Double_value>,
    "relexpirymillisec":<Double_value>,
    "absexpiry":<String[]_value>,
    "absexpirygmt":<String[]_value>,
    "weaknegrelexpiry":<Double_value>,
    "hitparams":<String[]_value>,
    "invalparams":<String[]_value>,
    "ignoreparamvaluecase":<String_value>,
    "matchcookies":<String_value>,
    "invalrestrictedtohost":<String_value>,
    "polleverytime":<String_value>,
    "ignorereloadreq":<String_value>,
    "removecookies":<String_value>,
    "prefetch":<String_value>,
    "prefetchperiod":<Double_value>,
    "prefetchperiodmillisec":<Double_value>,
    "prefetchmaxpending":<Double_value>,
    "flashcache":<String_value>,
    "expireatlastbyte":<String_value>,
    "insertvia":<String_value>,
    "insertage":<String_value>,
    "insertetag":<String_value>,
    "cachecontrol":<String_value>,
    "quickabortsize":<Double_value>,
    "minressize":<Double_value>,
    "maxressize":<Double_value>,
    "memlimit":<Double_value>,
    "ignorereqcachinghdrs":<String_value>,
    "minhits":<Integer_value>,
    "alwaysevalpolicies":<String_value>,
    "persistha":<String_value>,
    "pinned":<String_value>,
    "lazydnsresolve":<String_value>,
    "hitselector":<String_value>,
    "invalselector":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachecontentgroup":{
    "name":<String_value>,
    "weakposrelexpiry":true,
    "heurexpiryparam":true,
    "relexpiry":true,
    "relexpirymillisec":true,
    "absexpiry":true,
    "absexpirygmt":true,
    "weaknegrelexpiry":true,
    "hitparams":true,
    "invalparams":true,
    "ignoreparamvaluecase":true,
    "matchcookies":true,
    "invalrestrictedtohost":true,
    "polleverytime":true,
    "ignorereloadreq":true,
    "removecookies":true,
    "prefetch":true,
    "prefetchperiod":true,
    "prefetchperiodmillisec":true,
    "prefetchmaxpending":true,
    "flashcache":true,
    "expireatlastbyte":true,
    "insertvia":true,
    "insertage":true,
    "insertetag":true,
    "cachecontrol":true,
    "quickabortsize":true,
    "minressize":true,
    "maxressize":true,
    "memlimit":true,
    "ignorereqcachinghdrs":true,
    "minhits":true,
    "alwaysevalpolicies":true,
    "persistha":true,
    "pinned":true,
    "lazydnsresolve":true,
    "hitselector":true,
    "invalselector":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

expire

URL: http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup?**action=expire**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachecontentgroup":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

flush

URL: http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup?**action=flush**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachecontentgroup":{
    "name":<String_value>,
    "query":<String_value>,
    "host":<String_value>,
    "selectorvalue":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

save

URL: http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup?**action=save**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachecontentgroup":{
    "name":<String_value>,
    "tosecondary":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cachecontentgroup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cachecontentgroup resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachecontentgroup": [ {
    "name":<String_value>,
    "flags":<Double_value>,
    "type":<String_value>,
    "relexpiry":<Double_value>,
    "relexpirymillisec":<Double_value>,
    "absexpiry":<String[]_value>,
    "absexpirygmt":<String[]_value>,
    "heurexpiryparam":<Double_value>,
    "weakposrelexpiry":<Double_value>,
    "weaknegrelexpiry":<Double_value>,
    "hitparams":<String[]_value>,
    "invalparams":<String[]_value>,
    "ignoreparamvaluecase":<String_value>,
    "matchcookies":<String_value>,
    "invalrestrictedtohost":<String_value>,
    "polleverytime":<String_value>,
    "ignorereloadreq":<String_value>,
    "removecookies":<String_value>,
    "prefetch":<String_value>,
    "prefetchperiod":<Double_value>,
    "prefetchperiodmillisec":<Double_value>,
    "prefetchcur":<Double_value>,
    "prefetchmaxpending":<Double_value>,
    "flashcache":<String_value>,
    "expireatlastbyte":<String_value>,
    "insertvia":<String_value>,
    "insertage":<String_value>,
    "insertetag":<String_value>,
    "cachecontrol":<String_value>,
    "quickabortsize":<Double_value>,
    "minressize":<Double_value>,
    "maxressize":<Double_value>,
    "memusage":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "memdusage":<Double_value>,
    "disklimit":<Double_value>,
    "memlimit":<Double_value>,
    "ignorereqcachinghdrs":<String_value>,
    "cachenon304hits":<Double_value>,
    "cache304hits":<Double_value>,
    "cachecells":<Double_value>,
    "cachegroupincarnation":<Double_value>,
    "minhits":<Integer_value>,
    "alwaysevalpolicies":<String_value>,
    "persist":<String_value>,
    "persistha":<String_value>,
    "pinned":<String_value>,
    "lazydnsresolve":<String_value>,
    "hitselector":<String_value>,
    "invalselector":<String_value>,
    "policyname":<String[]_value>,
    "cachenuminvalpolicy":<Double_value>,
    "markercells":<Double_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachecontentgroup": [ {
    "name":<String_value>,
    "flags":<Double_value>,
    "type":<String_value>,
    "relexpiry":<Double_value>,
    "relexpirymillisec":<Double_value>,
    "absexpiry":<String[]_value>,
    "absexpirygmt":<String[]_value>,
    "heurexpiryparam":<Double_value>,
    "weakposrelexpiry":<Double_value>,
    "weaknegrelexpiry":<Double_value>,
    "hitparams":<String[]_value>,
    "invalparams":<String[]_value>,
    "ignoreparamvaluecase":<String_value>,
    "matchcookies":<String_value>,
    "invalrestrictedtohost":<String_value>,
    "polleverytime":<String_value>,
    "ignorereloadreq":<String_value>,
    "removecookies":<String_value>,
    "prefetch":<String_value>,
    "prefetchperiod":<Double_value>,
    "prefetchperiodmillisec":<Double_value>,
    "prefetchcur":<Double_value>,
    "prefetchmaxpending":<Double_value>,
    "flashcache":<String_value>,
    "expireatlastbyte":<String_value>,
    "insertvia":<String_value>,
    "insertage":<String_value>,
    "insertetag":<String_value>,
    "cachecontrol":<String_value>,
    "quickabortsize":<Double_value>,
    "minressize":<Double_value>,
    "maxressize":<Double_value>,
    "memusage":<Double_value>,
    "memdusage":<Double_value>,
    "disklimit":<Double_value>,
    "memlimit":<Double_value>,
    "ignorereqcachinghdrs":<String_value>,
    "cachenon304hits":<Double_value>,
    "cache304hits":<Double_value>,
    "cachecells":<Double_value>,
    "cachegroupincarnation":<Double_value>,
    "minhits":<Integer_value>,
    "alwaysevalpolicies":<String_value>,
    "persist":<String_value>,
    "persistha":<String_value>,
    "pinned":<String_value>,
    "lazydnsresolve":<String_value>,
    "hitselector":<String_value>,
    "invalselector":<String_value>,
    "policyname":<String[]_value>,
    "cachenuminvalpolicy":<Double_value>,
    "markercells":<Double_value>,
    "builtin":<String[]_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cachecontentgroup?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachecontentgroup": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cacheforwardproxy

May 20, 2016

Configuration for forward proxy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipaddress	<String>	Read-write	IP address of the NetScaler appliance or a cache server for which the cache acts as a proxy. Requests coming to the NetScaler with the configured IP address are forwarded to the particular address, without involving the Integrated Cache in any way. Minimum length = 1
port	<Integer>	Read-write	Port on the NetScaler appliance or a server for which the cache acts as a proxy. Minimum value = 1 Range 1 - 65535
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheforwardproxy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cacheforwardproxy":{
    "ipaddress":<String_value>,
    "port":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheforwardproxy/ipaddress_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheforwardproxy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cacheforwardproxy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cacheforwardproxy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cacheforwardproxy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/cacheforwardproxy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/cacheforwardproxy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cacheforwardproxy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cacheforwardproxy": [ {
      "ipaddress":<String_value>,
      "port":<Integer_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheforwardproxy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "cacheforwardproxy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cacheglobal_binding

May 20, 2016

Binding object which returns the resources bound to cacheglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cacheglobal_cachepolicy_binding	<cacheglobal_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to cacheglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "cacheglobal_binding": [ {
      "cacheglobal_cachepolicy_binding":<cacheglobal_cachepolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cacheglobal_cachepolicy_binding

May 20, 2016

Binding object showing the cachepolicy that can be bound to cacheglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policy	<String>	Read-write	Name of the cache policy.
type	<String>	Read-write	The bind point to which policy is bound. When you specify the type, detailed information about that bind point appears. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
precededefrules	<String>	Read-write	Specify whether this policy should be evaluated. Default value: NO Possible values = YES, NO
labeltype	<String>	Read-write	Type of policy label to invoke. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE. (To invoke a label associated with a virtual server, specify the name of the virtual server.).
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority. Applicable only to default-syntax policies.
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
flowtype	<Double>	Read-only	flowtype of the bound cache policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/cacheglobal_cachepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"cacheglobal_cachepolicy_binding":{
    "policy":<String_value>,
    "priority":<Double_value>,
    "precededefrules":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheglobal_cachepolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheglobal_cachepolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cacheglobal_cachepolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cacheglobal_cachepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cacheglobal_cachepolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cacheglobal_cachepolicy_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cacheglobal_cachepolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "gotoprioritiyexpression":<String_value>,
    "policy":<String_value>,
    "type":<String_value>,
    "precededefrules":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "numpol":<Double_value>,
    "flowtype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheglobal_cachepolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "cacheglobal_cachepolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cacheobject

May 20, 2016

Configuration for cache object resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
url	<String>	Read-write	URL of the particular object whose details is required. Parameter "host" must be specified along with the URL. Minimum length = 1
locator	<Double>	Read-write	ID of the cached object.
httpstatus	<Double>	Read-write	HTTP status of the object.
host	<String>	Read-write	Host name of the object. Parameter "url" must be specified. Minimum length = 1
port	<Integer>	Read-write	Host port of the object. You must also set the Host parameter. Default value: 80 Minimum value = 1
groupname	<String>	Read-write	Name of the content group to which the object belongs. It will display only the objects belonging to the specified content group. You must also set the Host parameter.
httpmethod	<String>	Read-write	HTTP request method that caused the object to be stored. Default value: GET Possible values = GET, POST
group	<String>	Read-write	Name of the content group whose objects should be listed.
ignoremarkerobjects	<String>	Read-write	Ignore marker objects. Marker objects are created when a response exceeds the maximum or minimum response size for the content group or has not yet received the minimum number of hits for the content group. Possible values = ON, OFF
includenotreadyobjects	<String>	Read-write	Include responses that have not yet reached a minimum number of hits before being cached. Possible values = ON, OFF
tosecondary	<String>	Read-write	Object will be saved onto Secondary. Default value: NO Possible values = YES, NO
cacheressize	<Double>	Read-only	Cache response size of the object.
cachereshdrsize	<Double>	Read-only	Cache response header size of the object.
cacheetag	<String>	Read-only	Cache ETag of the object.
httpstatusoutput	<Double>	Read-only	HTTP status of the object.
cachereslastmod	<String>	Read-only	Value of "Last-modified" header.
cachecontrol	<String>	Read-only	Cache-Control header of the object.
cacheresdate	<String>	Read-only	Value of "Date" header.
contentgroup	<String>	Read-only	Name of the contentgroup in which it is stored.
destipv46	<String>	Read-only	Destination IP.
destport	<Integer>	Read-only	Destination Port. Range 1 - 65535
cachecellcomplex	<String>	Read-only	The state of the parameterized caching on this cell. Possible values = YES, NO
hitparams	<String[]>	Read-only	Parameterized hit evaluation of an object.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cachecellreqtime	<Double>	Read-only	Required time of the cache cell object.
cachecellrestime	<Double>	Read-only	Response time to the cache cell object.
cachecurage	<Double>	Read-only	Current age of the cache object.
cachecellexpires	<Double>	Read-only	Expiry time of the cache cell object in seconds.
cachecellexpiresmillisec	<Double>	Read-only	Expiry time of the cache cell object in milliseconds.
flushed	<String>	Read-only	Specifies whether the object is flushed. Possible values = YES, NO
prefetch	<String>	Read-only	Specifies whether Integrated Cache should attempt to refresh an object immediately before it goes stale. Possible values = YES, NO
prefetchperiod	<Double>	Read-only	The duration in seconds of the period during which prefetch should be attempted, immediately before the object's calculated expiry time.
prefetchperiodmillisec	<Double>	Read-only	The duration in milliseconds of the period during which prefetch should be attempted, immediately before the object's calculated expiry time.
cachecellcurreaders	<Double>	Read-only	Current readers of the cache cell object.
cachecellcurmisses	<Double>	Read-only	Current misses of the cache cell object.
cachecellhits	<Double>	Read-only	Cache cell hits.
cachecellmisses	<Double>	Read-only	Cache cell misses.
cachecelldhits	<Double>	Read-only	Cache cell disk hits.
cachecellcompressionformat	<String>	Read-only	Compression format of this object. Identity means not compressed.
cachecellappfwmetadataexists	<String>	Read-only	AppFirewall cache object. Possible values = YES, NO
cachecellhttp11	<String>	Read-only	The state of the response to be HTTP/1.1. Possible values = YES, NO
cachecellweaketag	<String>	Read-only	The state of the weak HTTP Entity Tag in the cell. Possible values = YES, NO
cachecellresbadsize	<String>	Read-only	The marked state of the cell. Possible values = YES, NO
markerreason	<String>	Read-only	Reason for marking the cell. Possible values = Waiting for min hit, Response header is too big, Content-length header said response size is not in group size limit, Content-length response received more data, Content-length response received less data, Content-length response data is not in group size limit, Chunk response received more data, Chunk response data is not in group size limit, Bad chunk format, Fin terminated response data is not in group size limit
cachecellpolleverytime	<String>	Read-only	The state to poll every time on object. Possible values = YES, NO
cachecelletaginserted	<String>	Read-only	The state of the ETag to be inserted by IC for this object. Possible values = YES, NO
cachecellreadywithlastbyte	<String>	Read-only	The state of the complete arrived response. Possible values = YES, NO
cacheinmemory	<String>	Read-only	The cache data is available in memory. Possible values = YES, NO
cacheindisk	<String>	Read-only	The cache data is available in disk. Possible values = YES, NO
cacheinsecondary	<String>	Read-only	The cache data is available in secondary in HA deployment. Possible values = YES, NO
cachedirname	<String>	Read-only	The directory name used if saved.
cachefilename	<String>	Read-only	The filename used if saved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cachecellidesupverified	<String>	Read-only	Possible values = YES, NO
cachecellfwpxyobj	<String>	Read-only	The state of the object to be stored on a request to a forward proxy. Possible values = YES, NO
cachecellbasefile	<String>	Read-only	The state of delta being used as a basefile. Possible values = YES, NO
cachecellminhitflag	<String>	Read-only	The state of the minhit feature on this cell. Possible values = YES, NO
cachecellminhit	<Integer>	Read-only	Min hit value for the object.
policy	<Integer>	Read-only	Policy info for the object.
policyname	<String>	Read-only	Policy which created the object.
selectorname	<String[]>	Read-only	The hit selector for the object.
rule	<String[]>	Read-only	Selectors for this object.
selectorvalue	<String[]>	Read-only	The HTTP request method that caused the object to be stored.
cacheurls	<String>	Read-only	List of cache object URLs.
warnbucketskip	<Double>	Read-only	Bucket skipped warning.
totalobjs	<Double>	Read-only	Total objects.
httpcalloutcell	<String>	Read-only	Is it a http callout cell ?. Possible values = YES, NO
httpcalloutname	<String>	Read-only	Name of the http callout.
returntype	<String>	Read-only	Return type of the http callout. Possible values = BOOL, NUM, TEXT
httpcalloutresult	<String>	Read-only	First few bytes of http callout response.
locatorshow	<Double>	Read-only	ID of the cached object.
ceflags	<Double>	Read-only	Indicates state and type of cached cell.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

EXPIRE | FLUSH | SAVE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

expire

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheobject?**action=expire**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cacheobject":{
    "locator":<Double_value>,
    "url":<String_value>,
    "host":<String_value>,
    "port":<Integer_value>,
    "groupname":<String_value>,
    "httpmethod":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

flush

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheobject?**action=flush**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cacheobject":{
    "locator":<Double_value>,
    "url":<String_value>,
    "host":<String_value>,
    "port":<Integer_value>,
    "groupname":<String_value>,
    "httpmethod":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

save

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheobject?**action=save**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cacheobject":{
    "locator":<Double_value>,
    "tosecondary":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheobject

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/cacheobject?**args=url:<String_value>,locator:<Double_value>,httpstatus:<Double_value>,host:<String_value>,port:<Integer_value>,groupname:<String_va**

Use this query-parameter to get cacheobject resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/cacheobject?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cacheobject?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cacheobject resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/cacheobject?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/cacheobject?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cacheobject resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cacheobject": [ {
url:<String_value>,locator:<Double_value>,httpstatus:<Double_value>,host:<String_value>,port:<Integer_value>,groupname:<String_value>,httpmethod:<String_value>,group:<String_value>,ignoremarkerobjects:<String_value>,includenot
    "cachereshdrsize":<Double_value>,
    "cacheetag":<String_value>,
    "httpstatusoutput":<Double_value>,
    "cachereslastmod":<String_value>,
    "cachecontrol":<String_value>,
    "cacheresdate":<String_value>,
    "contentgroup":<String_value>,
    "destip":<Double_value>,
    "destipv46":<String_value>,
    "destport":<Integer_value>,
    "cachecellcomplex":<String_value>,
    "hitparams":<String[]_value>,
    "hitvalues":<String[]_value>,
    "cachecellreqtime":<Double_value>,
    "cachecellrestime":<Double_value>,
    "cachecurage":<Double_value>,
    "cachecellexpires":<Double_value>,
    "cachecellexpiresmillisec":<Double_value>,
    "flushed":<String_value>,
    "prefetch":<String_value>,
    "prefetchperiod":<Double_value>,
    "prefetchperiodmillisec":<Double_value>,
    "cachecellcurreaders":<Double_value>,
    "cachecellcurmisses":<Double_value>,
    "cachecellhits":<Double_value>,
    "cachecellmisses":<Double_value>,
    "cachecelldhits":<Double_value>,
    "cachecellgzipcompressed":<String_value>,
    "cachecelldeflatecompressed":<String_value>,
    "cachecellcompressionformat":<String_value>,
    "cachecellappfwmetadataexists":<String_value>,
    "cachecellhttp11":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cachecellweaketag":<String_value>,
    "cachecellresbadsize":<String_value>,
    "markerreason":<String_value>,
    "cachecellpolleverytime":<String_value>,
    "cachecelletaginserted":<String_value>,
    "cachecellreadywithlastbyte":<String_value>,
    "cacheinmemory":<String_value>,
    "cacheindisk":<String_value>,
    "cacheinsecondary":<String_value>,
    "cachedirname":<String_value>,
    "cachefilename":<String_value>,
    "cachecelldestipverified":<String_value>,
    "cachecellfwpxyobj":<String_value>,
    "cachecellbasefile":<String_value>,
    "cachecellminhitflag":<String_value>,
    "cachecellminhit":<Integer_value>,
    "policy":<Integer_value>,
    "policyname":<String_value>,
    "selectorname":<String[]_value>,
    "rule":<String[]_value>,
    "selectorvalue":<String[]_value>,
    "cacheurls":<String_value>,
    "warnbucketskip":<Double_value>,
    "totalobjs":<Double_value>,
    "httpcalloutcell":<String_value>,
    "httpcalloutname":<String_value>,
    "returntype":<String_value>,
    "httpcalloutresult":<String_value>,
    "locatorshow":<Double_value>,
    "ceflags":<Double_value>
}}}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheobject?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cacheobject": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cacheparameter

Jul 07, 2017

Configuration for cache parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
memlimit	<Double>	Read-write	Amount of memory available for storing the cache objects. In practice, the amount of memory available for caching can be less than half the total memory of the NetScaler appliance.
via	<String>	Read-write	String to include in the Via header. A Via header is inserted into all responses served from a content group if its Insert Via flag is set. Minimum length = 1
verifyusing	<String>	Read-write	Criteria for deciding whether a cached object can be served for an incoming HTTP request. Available settings function as follows: HOSTNAME - The URL, host name, and host port values in the incoming HTTP request header must match the cache policy. The IP address and the TCP port of the destination host are not evaluated. Do not use the HOSTNAME setting unless you are certain that no rogue client can access a rogue server through the cache. HOSTNAME_AND_IP - The URL, host name, host port in the incoming HTTP request header, and the IP address and TCP port of the destination server, must match the cache policy. DNS - The URL, host name and host port in the incoming HTTP request, and the TCP port must match the cache policy. The host name is used for DNS lookup of the destination server's IP address, and is compared with the set of addresses returned by the DNS lookup. Possible values = HOSTNAME, HOSTNAME_AND_IP, DNS
maxpostlen	<Double>	Read-write	Maximum number of POST body bytes to consider when evaluating parameters for a content group for which you have configured hit parameters and invalidation parameters. Default value: 4096 Minimum value = 0 Maximum value = 131072
prefetchmaxpending	<Double>	Read-write	Maximum number of outstanding prefetches in the Integrated Cache.
enablebypass	<String>	Read-write	Evaluate the request-time policies before attempting hit selection. If set to NO, an incoming request for which a matching object is found in cache storage results in a response regardless of the policy configuration. If the request matches a policy with a NOCACHE action, the request bypasses all cache processing. This parameter does not affect processing of requests that match any

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = YES, NO
undefaction	<String>	Read-write	Action to take when a policy cannot be evaluated. Possible values = NOCACHE, RESET
enablehaobjpersist	<String>	Read-write	The HA object persisting parameter. When this value is set to YES, cache objects can be synced to Secondary in a HA deployment. If set to NO, objects will never be synced to Secondary node. Default value: NO Possible values = YES, NO
disklimit	<Double>	Read-only	The disk limit for the Integrated Cache.
maxdisklimit	<Double>	Read-only	The maximum value of the memory limit for the Integrated Cache.
memlimitactive	<Double>	Read-only	Active value of the memory limit for the Integrated Cache.
maxmemlimit	<Double>	Read-only	The maximum value of the memory limit for the Integrated Cache.
prefetchcur	<Double>	Read-only	Number of current outstanding prefetches in the IC.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cacheparameter":{
    "memlimit":<Double_value>,
    "via":<String_value>,
    "verifyusing":<String_value>,
    "maxpostlen":<Double_value>,
    "prefetchmaxpending":<Double_value>,
    "enablebypass":<String_value>,
    "undefaction":<String_value>,
    "enablehaobjpersist":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cacheparameter":{
    "memlimit":true,
    "via":true,
    "verifyusing":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "maxpostlen":true,
    "prefetchmaxpending":true,
    "enablebypass":true,
    "undefaction":true,
    "enablehaobjpersist":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cacheparameter": [ {
    "disklimit":<Double_value>,
    "maxdisklimit":<Double_value>,
    "memlimit":<Double_value>,
    "memlimitactive":<Double_value>,
    "maxmemlimit":<Double_value>,
    "via":<String_value>,
    "verifyusing":<String_value>,
    "maxpostlen":<Double_value>,
    "prefetchcur":<Double_value>,
    "prefetchmaxpending":<Double_value>,
    "enablebypass":<String_value>,
    "undefaction":<String_value>,
    "enablehaobjpersist":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicy

Jul 07, 2017

Configuration for Integrated Cache policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name for the policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the policy is created. Minimum length = 1
rule	<String>	Read-write	Expression against which the traffic is evaluated. Note: Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Action to apply to content that matches the policy. * CACHE or MAY_CACHE action - positive cachability policy * NOCACHE or MAY_NOCACHE action - negative cachability policy * INVAL action - Dynamic Invalidation Policy. Possible values = CACHE, NOCACHE, MAY_CACHE, MAY_NOCACHE, INVAL
storeingroup	<String>	Read-write	Name of the content group in which to store the object when the final result of policy evaluation is CACHE. The content group must exist before being mentioned here. Use the "show cache contentgroup" command to view the list of existing content groups. Minimum length = 1
invalgroups	<String[]>	Read-write	Content group(s) to be invalidated when the INVAL action is applied. Maximum number of content groups that can be specified is 16. Minimum length = 1
invalobjects	<String[]>	Read-write	Content groups(s) in which the objects will be invalidated if the action is INVAL. Minimum length = 1
undefaction	<String>	Read-write	Action to be performed when the result of rule evaluation is undefined. Possible values = NOCACHE, RESET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
newname	<String>	Read-write	character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
hits	<Double>	Read-only	Hits.
undefhits	<Double>	Read-only	Number of Undef hits.
flags	<Double>	Read-only	Flag.
builtin	<String[]>	Read-only	. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachepolicy":{
    "policyname":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "storeingroup":<String_value>,
    "invalgroups":<String[]_value>,
    "invalobjects":<String[]_value>,
    "undefaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy/policyname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachepolicy":{
    "policyname":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "storeingroup":<String_value>,
    "invalgroups":<String[]_value>,
    "invalobjects":<String[]_value>,
    "undefaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachepolicy":{
    "policyname":<String_value>,
    "storeingroup":true,
    "invalgroups":true,
    "invalobjects":true,
    "undefaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachepolicy":{
    "policyname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cachepolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cachepolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicy": [ {
      "policyname":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "storeingroup":<String_value>,
      "invalgroups":<String[]_value>,
      "invalobjects":<String[]_value>,
      "hits":<Double_value>,
      "undefaction":<String_value>,
      "undefhits":<Double_value>,
      "flags":<Double_value>,
      "precededefrules":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy/policyname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy/policyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy/policyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicy": [ {
      "policyname":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "storeingroup":<String_value>,
      "invalgroups":<String[]_value>,
      "invalobjects":<String[]_value>,
      "hits":<Double_value>,
      "undefaction":<String_value>,
      "undefhits":<Double_value>,
      "flags":<Double_value>,
      "precededefrules":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "cachepolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicylabel

May 20, 2016

Configuration for cache policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name for the label. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the label is created.
evaluates	<String>	Read-write	When to evaluate policies bound to this label: request-time or response-time. Possible values = REQ, RES, MSSQL_REQ, MSSQL_RES, MYSQL_REQ, MYSQL_RES
newname	<String>	Read-write	New name for the cache-policy label. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label to invoke: an unnamed label associated with a virtual server, or user-defined policy label. Possible values = reqvserver, resvserver, policylabel
invoke_labelname	<String>	Read-only	Name of the policy label to invoke if the current policy rule evaluates to TRUE.
flowtype	<Double>	Read-only	Flowtype of the bound cache policy.
builtin	<String[]>	Read-only	. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachepolicylabel":{
    "labelname":<String_value>,
    "evaluates":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cachepolicylabel":{
    "labelname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cachepolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cachepolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicylabel": [ {
    "labelname":<String_value>,
    "evaluates":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "flowtype":<Double_value>,
    "builtin":<String[]_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicylabel": [ {
    "labelname":<String_value>,
    "evaluates":<String_value>,
    "numpol":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "hits":<Double_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "flowtype":<Double_value>,
    "builtin":<String[]_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicylabel": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicylabel_binding

May 20, 2016

Binding object which returns the resources bound to cachepolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the cache-policy label about which to display information.
cachepolicylabel_policybinding_binding	<cachepolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to cachepolicylabel.
cachepolicylabel_cachepolicy_binding	<cachepolicylabel_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to cachepolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "cachepolicylabel_binding": [ {
    "labelname":<String_value>,
    "cachepolicylabel_policybinding_binding":<cachepolicylabel_policybinding_binding[]_value>,
    "cachepolicylabel_cachepolicy_binding":<cachepolicylabel_cachepolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicylabel_cachepolicy_binding

May 20, 2016

Binding object showing the cachepolicy that can be bound to cachepolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the cache policy to bind to the policy label.
labeltype	<String>	Read-write	Type of policy label to invoke: an unnamed label associated with a virtual server, or user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the cache policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	Name of the policy label to invoke if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next-lower priority.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/cachepolicylabel_cachepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"cachepolicylabel_cachepolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicylabel_cachepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cachepolicylabel_cachepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicylabel_cachepolicy_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel_cachepolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cachepolicylabel_cachepolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicylabel_policybinding_binding

May 20, 2016

Binding object showing the policybinding that can be bound to cachepolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the cache policy to bind to the policy label.
labeltype	<String>	Read-write	Type of policy label to invoke: an unnamed label associated with a virtual server, or user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the cache policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	Name of the policy label to invoke if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next-lower priority.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform

© 1999-2017 Citrix Systems, Inc. All rights reserved.

individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicylabel_policybinding_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cachepolicylabel_policybinding_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicylabel_policybinding_binding": [ {
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "policyname":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cachepolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicy_binding

May 20, 2016

Binding object which returns the resources bound to cachepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the cache policy about which to display details. Minimum length = 1
cachepolicy_cacheglobal_binding	<cachepolicy_cacheglobal_binding[]>	Read-only	cacheglobal that can be bound to cachepolicy.
cachepolicy_cachepolicylabel_binding	<cachepolicy_cachepolicylabel_binding[]>	Read-only	cachepolicylabel that can be bound to cachepolicy.
cachepolicy_csvserver_binding	<cachepolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to cachepolicy.
cachepolicy_lbvserver_binding	<cachepolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to cachepolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_binding/policyname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "cachepolicy_binding": [ {
      "policyname":<String_value>,
      "cachepolicy_cachepolicylabel_binding":<cachepolicy_cachepolicylabel_binding[]_value>,
      "cachepolicy_lbvserver_binding":<cachepolicy_lbvserver_binding[]_value>,
      "cachepolicy_cacheglobal_binding":<cachepolicy_cacheglobal_binding[]_value>,
      "cachepolicy_csvserver_binding":<cachepolicy_csvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicy_cacheglobal_binding

Jul 07, 2017

Binding object showing the cacheglobal that can be bound to cachepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the cache policy about which to display details. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Priority.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_cacheglobal_binding/policyname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_cacheglobal_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicy_cacheglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_cacheglobal_binding/policyname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cachepolicy_cacheglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicy_cacheglobal_binding": [ {
      "policyname":<String_value>,
      "boundto":<String_value>,
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "labeltype":<String_value>,
      "activepolicy":<Integer_value>,
      "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_cacheglobal_binding/policyname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cachepolicy_cacheglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicy_cachepolicylabel_binding

May 20, 2016

Binding object showing the cachepolicylabel that can be bound to cachepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the cache policy about which to display details. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Priority.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_cachepolicylabel_binding/policyname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_cachepolicylabel_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicy_cachepolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_cachepolicylabel_binding/policyname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cachepolicy_cachepolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "cachepolicy_cachepolicylabel_binding": [ {
    "policyname":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_cachepolicylabel_binding/policyname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cachepolicy_cachepolicylabel_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicy_csvserver_binding

Jul 07, 2017

Binding object showing the csvserver that can be bound to cachepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the cache policy about which to display details. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Priority.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_csvserver_binding/policyname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_csvserver_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_csvserver_binding/policyname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cachepolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicy_csvserver_binding": [ {
    "policyname":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_csvserver_binding/policyname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cachepolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicy_lbvserver_binding

Jul 07, 2017

Binding object showing the lbvserver that can be bound to cachepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the cache policy about which to display details. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Priority.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_lbvserver_binding/policyname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_lbvserver_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cachepolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_lbvserver_binding/policyname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cachepolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicy_lbvserver_binding": [ {
    "policyname":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cachepolicy_lbvserver_binding/policyname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cachepolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cacheselector

May 20, 2016

Configuration for cache selector resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
selectorname	<String>	Read-write	Name for the selector. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters.
rule	<String[]>	Read-write	One or multiple PIXL expressions for evaluating an HTTP request or response. Minimum length = 1
flags	<Double>	Read-only	Flags.
builtin	<String[]>	Read-only	. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheselector

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cacheselector":{
    "selectorname":<String_value>,
    "rule":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheselector/selectorname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheselector

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cacheselector":{
    "selectorname":<String_value>,
    "rule":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheselector

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cacheselector?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cacheselector?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cacheselector resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/cacheselector?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/cacheselector?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cacheselector resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cacheselector": [ {
      "selectorname":<String_value>,
      "flags":<Double_value>,
      "rule":<String[]_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheselector/selectorname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cacheselector/selectorname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/cacheselector/selectorname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cacheselector": [ {
      "selectorname":<String_value>,
      "flags":<Double_value>,
      "rule":<String[]_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cacheselector?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cacheselector": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cluster

May 20, 2016

Cluster commands.

Name	Description
cluster	
clusterfiles	Configuration for files
clusterinstance	Configuration for cluster instance
clusterinstance_binding	Binding object showing the resources that can be bound to clusterinstance
clusterinstance_clusternode_binding	Binding object showing the clusternode that can be bound to clusterinstance
clusternode	Configuration for cluster node
clusternodegroup	Configuration for Node group object type
clusternodegroup_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to clusternodegroup
clusternodegroup_binding	Binding object showing the resources that can be bound to clusternodegroup
clusternodegroup_clusternode_binding	Binding object showing the clusternode that can be bound to clusternodegroup
clusternodegroup_crvserver_binding	Binding object showing the crvserver that can be bound to clusternodegroup
clusternodegroup_csvserver_binding	Binding object showing the csvserver that can be bound to clusternodegroup
clusternodegroup_gslbsite_binding	Binding object showing the gslbsite that can be bound to clusternodegroup
clusternodegroup_gslbvserver_binding	Binding object showing the gslbvserver that can be bound to clusternodegroup
clusternodegroup_lbvserver_binding	Binding object showing the lbvserver that can be bound to clusternodegroup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_nslimitidentifier_binding	
Name	**Description**
clusternodegroup_service_binding	Binding object showing the service that can be bound to clusternodegroup
clusternodegroup_streamidentifier_binding	Binding object showing the streamidentifier that can be bound to clusternodegroup
clusternodegroup_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to clusternodegroup
clustersync	Configuration for sync

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cluster

May 20, 2016

Configuration for 0 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clip	<String>	Read-write	Cluster IP address to which to add the node.
password	<String>	Read-write	Password for the nsroot account of the configuration coordinator (CCO).

Operations (click to see Properties)

JOIN

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

join

URL: http://<netscaler-ip-address>/nitro/v1/config/cluster?**action=join**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cluster":{
    "clip":<String_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusterfiles

May 20, 2016

Configuration for files resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
mode	<String[]>	Read-write	The directories and files to be synchronized. The available settings function as follows: Mode Paths all /nsconfig/ssl/ /var/netscaler/ssl/ /var/vpn/bookmark/ /nsconfig/dns/ /nsconfig/htmlinjection/ /netscaler/htmlinjection/ens/ /nsconfig/monitors/ /nsconfig/nstemplates/ /nsconfig/ssh/ /nsconfig/rc.netscaler /nsconfig/resolv.conf /nsconfig/inetd.conf /nsconfig/syslog.conf /nsconfig/snmpd.conf /nsconfig/ntp.conf /nsconfig/httpd.conf /nsconfig/sshd_config /nsconfig/hosts /nsconfig/enckey /var/nslw.bin/etc/krb5.conf /var/nslw.bin/etc/krb5.keytab /var/lib/likewise/db/ /var/download/ /var/wi/tomcat/webapps/ /var/wi/tomcat/conf/Catalina/localhost/ /var/wi/java_home/lib/security/cacerts /var/wi/java_home/jre/lib/security/cacerts /var/netscaler/locdb/ ssl /nsconfig/ssl/ /var/netscaler/ssl/ bookmarks /var/vpn/bookmark/ dns /nsconfig/dns/ htmlinjection /nsconfig/htmlinjection/ imports /var/download/ misc /nsconfig/license/ /nsconfig/rc.conf all_plus_misc Includes *all* files and /nsconfig/license/ and /nsconfig/rc.conf.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = all, bookmarks, sst, htmlinjection, imports, misc, dns, krb, all_plus_misc, all_minus_misc

Operations (click to see Properties)

SYNC

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

sync

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterfiles?**action=sync**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusterfiles":{
    "mode":<String[]_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusterinstance

May 20, 2016

Configuration for cluster instance resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clid	<Double>	Read-write	Unique number that identifies the cluster. Minimum value = 1 Maximum value = 16
deadinterval	<Double>	Read-write	Amount of time, in seconds, after which nodes that do not respond to the heartbeats are assumed to be down.If the value is less than 3 sec, set the helloInterval parameter to 200 msec. Default value: 3 Minimum value = 1 Maximum value = 60
hellointerval	<Double>	Read-write	Interval, in milliseconds, at which heartbeats are sent to each cluster node to check the health status.Set the value to 200 msec, if the deadInterval parameter is less than 3 sec. Default value: 200 Minimum value = 200 Maximum value = 1000
preemption	<String>	Read-write	Preempt a cluster node that is configured as a SPARE if an ACTIVE node becomes available. Default value: DISABLED Possible values = ENABLED, DISABLED
quorumtype	<String>	Read-write	Quorum Configuration Choices - "Majority" (recommended) requires majority of nodes to be online for the cluster to be UP. "None" relaxes this requirement. Default value: MAJORITY Possible values = MAJORITY, NONE
inc	<String>	Read-write	This option is required if the cluster nodes reside on different networks. Default value: DISABLED Possible values = ENABLED, DISABLED
processlocal	<String>	Read-write	By turning on this option packets destined to a service in a cluster will not under go any steering. Default value: DISABLED Possible values = ENABLED, DISABLED
nodegroup	<String>	Read-write	The node group in a Cluster system used for transition from L2 to L3.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
adminstate	<String>	Read-only	Cluster Admin State. Default value: DISABLED Possible values = ENABLED, DISABLED
propstate	<String>	Read-only	Enable/Disable the execution of commands on the cluster. This will not impact the execution of commands on individual cluster nodes by using the NSIP. Default value: ENABLED Possible values = ENABLED, DISABLED
operationalstate	<String>	Read-only	Cluster Operational State. Possible values = ENABLED, DISABLED
status	<String>	Read-only	Cluster Operational State. Possible values = DOWN, UP, PARTIAL-UP, UNKNOWN
rsskeymismatch	<Boolean>	Read-only	This argument is used to determine if there is a RSS key mismatch at cluster instance level.
nodegroupstatewarning	<Boolean>	Read-only	This argumnt is used to determind whether all the cluster nodes are bound to nodegroup with state set.
licensemismatch	<Boolean>	Read-only	This argument is used to determine if there is a License mismatch at cluster instance level.
jumbonotsupported	<Boolean>	Read-only	This argument is used to determine if Jumbo framework is not supported at cluster instance level.
clusternoheartbeatonnode	<Boolean>	Read-only	HB is not seen on the backplane interface of member node.
clusternolinksetmbf	<Boolean>	Read-only	MBF is enabled but linkset is not configured .
clusternospottedip	<Boolean>	Read-only	There is no spotted SNIP or MIP.
operationalpropstate	<String>	Read-only	Cluster Operational Propagation State. Default value: ENABLED Possible values = UNKNOWN, ENABLED, DISABLED, AUTO DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusterinstance":{
    "clid":<Double_value>,
    "deadinterval":<Double_value>,
    "hellointerval":<Double_value>,
    "preemption":<String_value>,
    "quorumtype":<String_value>,
    "inc":<String_value>,
    "processlocal":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance/clid_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusterinstance":{
    "clid":<Double_value>,
    "deadinterval":<Double_value>,
    "hellointerval":<Double_value>,
    "preemption":<String_value>,
    "quorumtype":<String_value>,
    "inc":<String_value>,
    "processlocal":<String_value>,
    "nodegroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance?**action=unset**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusterinstance":{
    "clid":<Double_value>,
    "deadinterval":true,
    "hellointerval":true,
    "preemption":true,
    "quorumtype":true,
    "inc":true,
    "processlocal":true,
    "nodegroup":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusterinstance":{
    "clid":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusterinstance":{
    "clid":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/clusterinstance?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusterinstance?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of clusterinstance resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/clusterinstance?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusterinstance?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusterinstance resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusterinstance": [ {
      "clid":<Double_value>,
      "deadinterval":<Double_value>,
      "hellointerval":<Double_value>,
      "preemption":<String_value>,
      "adminstate":<String_value>,
      "quorumtype":<String_value>,
      "propstate":<String_value>,
      "inc":<String_value>,
      "nodegroup":<String_value>,
      "processlocal":<String_value>,
      "operationalstate":<String_value>,
      "status":<String_value>,
      "rsskeymismatch":<Boolean_value>,
      "nodegroupstatewarning":<Boolean_value>,
      "licensemismatch":<Boolean_value>,
      "jumbonotsupported":<Boolean_value>,
      "clusternoheartbeatonnode":<Boolean_value>,
      "clusternolinksetmbf":<Boolean_value>,
      "clusternospottedip":<Boolean_value>,
      "operationalpropstate":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance/clid_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/clusterinstance/clid_value<Double>?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/clusterinstance/clid_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusterinstance": [ {
    "clid":<Double_value>,
    "deadinterval":<Double_value>,
    "hellointerval":<Double_value>,
    "preemption":<String_value>,
    "adminstate":<String_value>,
    "quorumtype":<String_value>,
    "propstate":<String_value>,
    "inc":<String_value>,
    "nodegroup":<String_value>,
    "processlocal":<String_value>,
    "operationalstate":<String_value>,
    "status":<String_value>,
    "rsskeymismatch":<Boolean_value>,
    "nodegroupstatewarning":<Boolean_value>,
    "licensemismatch":<Boolean_value>,
    "jumbonotsupported":<Boolean_value>,
    "clusternoheartbeatonnode":<Boolean_value>,
    "clusternolinksetmbf":<Boolean_value>,
    "clusternospottedip":<Boolean_value>,
    "operationalpropstate":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "clusterinstance": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusterinstance_binding

May 20, 2016

Binding object which returns the resources bound to clusterinstance.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clid	<Double>	Read-write	Unique number that identifies the cluster. Minimum value = 1 Maximum value = 16
clusterinstance_clusternode_binding	<clusterinstance_clusternode_binding[]>	Read-only	clusternode that can be bound to clusterinstance.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance_binding/clid_value<Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "clusterinstance_binding": [ {
      "clid":<Double_value>,
      "clusterinstance_clusternode_binding":<clusterinstance_clusternode_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusterinstance_clusternode_binding

Jul 07, 2017

Binding object showing the clusternode that can be bound to clusterinstance.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
nodeid	<Double>	Read-write	The unique number that identiies a cluster. Minimum value = 0 Maximum value = 31
clid	<Double>	Read-write	Unique number that identifies the cluster. Minimum value = 1 Maximum value = 16
state	<String>	Read-only	Active, Spare or Passive. An active node serves traffic. A spare node serves as a backup for active nodes. A passive node does not serve traffic. This may be useful during temporary maintenance activity where it is desirable that the node takes part in the consensus protocol, but not serve traffic. Default value: PASSIVE Possible values = ACTIVE, SPARE, PASSIVE
nodersskeymismatch	<Boolean>	Read-only	This argument is used to determine if there is a RSS key mismatch at cluster node level.
nodelicensemismatch	<Boolean>	Read-only	This argument is used to determine if there is a License mismatch at cluster node level.
effectivestate	<String>	Read-only	Node effective health state. Possible values = UP, NOT UP, UNKNOWN, INIT
islocalnode	<Boolean>	Read-only	This argument is used to determine whether it is local node.
isconfigurationcoordinator	<Boolean>	Read-only	This argument is used to determine whether the node is configuration coordinator (CCO).
nodejumbonotsupported	<Boolean>	Read-only	This argument is used to determine if Jumbo framework not supported at cluster node level.
health	<String>	Read-only	Node Health state. Possible values = UNKNOWN, INIT, DOWN, UP, Some enabled and HAMON interfaces of the node are down, All interfaces of the node are down or disabled, SSL card(s) of the node have failed, Route Monitor(s) of the node have failed, Service state is being synchronized with the cluster, The backplane interface is not set, is down, or is disabled, The CLAG

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			synchronized with the cluster, The Syn Cookie is being synchronized with the cluster, Unknown Health, AAA keys are being sychronized with the cluster
clusterhealth	<String>	Read-only	Node clusterd state. Possible values = UP, Configurations of the node are lagging behind the cluster by more than 256 commands, The config sync operation has exceeded the time limit of 60 seconds, The config sync operation (full sync) is in progress, The node is not in sync with the cluster configurations as sync is disabled on this node, The execution of a configuration command has failed on this node, UNKNOWN
masterstate	<String>	Read-only	Master state. Possible values = INACTIVE, ACTIVE, UNKNOWN
ipaddress	<String>	Read-only	The IP Address of the node.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance_clusternode_binding/clid_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/clusterinstance_clusternode_binding/clid_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusterinstance_clusternode_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusterinstance_clusternode_binding/clid_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusterinstance_clusternode_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusterinstance_clusternode_binding": [ {
    "nodeid":<Double_value>,
    "clid":<Double_value>,
    "state":<String_value>,
    "nodersskeymismatch":<Boolean_value>,
    "nodelicensemismatch":<Boolean_value>,
    "effectivestate":<String_value>,
    "islocalnode":<Boolean_value>,
    "isconfigurationcoordinator":<Boolean_value>,
    "nodejumbonotsupported":<Boolean_value>,
    "health":<String_value>,
    "clusterhealth":<String_value>,
    "masterstate":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ipaddress":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusterinstance_clusternode_binding/clid_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"clusterinstance_clusternode_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternode

May 20, 2016

Configuration for cluster node resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
nodeid	<Double>	Read-write	Unique number that identifies the cluster node. Minimum value = 0 Maximum value = 31
ipaddress	<String>	Read-write	NetScaler IP (NSIP) address of the appliance to add to the cluster. Must be an IPv4 address. Minimum length = 1
state	<String>	Read-write	Admin state of the cluster node. The available settings function as follows: ACTIVE - The node serves traffic. SPARE - The node does not serve traffic unless an ACTIVE node goes down. PASSIVE - The node does not serve traffic, unless you change its state. PASSIVE state is useful during temporary maintenance activities in which you want the node to take part in the consensus protocol but not to serve traffic. Default value: PASSIVE Possible values = ACTIVE, SPARE, PASSIVE
backplane	<String>	Read-write	Interface through which the node communicates with the other nodes in the cluster. Must be specified in the three-tuple form n/c/u, where n represents the node ID and c/u refers to the interface on the appliance. Minimum length = 1
priority	<Double>	Read-write	Preference for selecting a node as the configuration coordinator. The node with the lowest priority value is selected as the configuration coordinator. When the current configuration coordinator goes down, the node with the next lowest priority is made the new configuration coordinator. When the original node comes back up, it will preempt the new configuration coordinator and take over as the configuration coordinator. Note: When priority is not configured for any of the nodes or if multiple nodes have the same priority, the cluster elects one of the nodes as the configuration coordinator. Default value: 31 Minimum value = 0 Maximum value = 31
			The default node group in a Cluster system.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
clearnodegroupconfig	<String>	Read-write	Option to remove nodegroup config. Default value: YES Possible values = YES, NO
clusterhealth	<String>	Read-only	Node clusterd state. Possible values = UP, Configurations of the node are lagging behind the cluster by more than 256 commands, The config sync operation has exceeded the time limit of 60 seconds, The config sync operation (full sync) is in progress, The node is not in sync with the cluster configurations as sync is disabled on this node, The execution of a configuration command has failed on this node, UNKNOWN
effectivestate	<String>	Read-only	Node effective health state. Possible values = UP, NOT UP, UNKNOWN, INIT
operationalsyncstate	<String>	Read-only	Node Operational Reconciliation state. Default value: ENABLED Possible values = ENABLED, SUCCESS, IN PROGRESS, FAILED, INCREMENTAL SYNC DISABLED, DISABLED, UNKNOWN
masterstate	<String>	Read-only	Node Master state. Possible values = INACTIVE, ACTIVE, UNKNOWN
health	<String>	Read-only	Node Health state. Possible values = UNKNOWN, INIT, DOWN, UP, Some enabled and HAMON interfaces of the node are down, All interfaces of the node are down or disabled, SSL card(s) of the node have failed, Route Monitor(s) of the node have failed, Service state is being synchronized with the cluster, The backplane interface is not set, is down, or is disabled, The CLAG member(s) of the node are down, Persistence sessions are being synchronized with the cluster, The Syn Cookie is being synchronized with the cluster, Unknown Health, AAA keys are being sychronized with the cluster
isconfigurationcoordinator	<Boolean>	Read-only	This argument is used to determine whether the node is configuration coordinator (CCO).
islocalnode	<Boolean>	Read-only	This argument is used to determine whether it is local node.
nodersskeymismatch	<Boolean>	Read-only	This argument is used to determine if there is a RSS key mismatch at cluster node level.
nodelicensemismatch	<Boolean>	Read-only	This argument is used to determine if there is a License mismatch at cluster node level.
nodejumbonotsupported	<Boolean>	Read-only	This argument is used to determine if Jumbo framework not supported at cluster node level.
nodelist	<Double[]>	Read-only	Nodelist for displaying Heartbeat not seen interfaces on a cluster node.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			on a cluster node.
enabledifaces	<String>	Read-only	Enabled Interfaces on a cluster node.
disabledifaces	<String>	Read-only	Disabled Interfaces on a cluster node.
partialfailifaces	<String>	Read-only	Partial Failure Interfaces on a cluster node.
hamonifaces	<String>	Read-only	Hamon Interfaces on a cluster node.
name	<String>	Read-only	Name of the state specific nodegroup. Minimum length = 1
cfgflags	<Double>	Read-only	Flag indicates whether the node is bound to cluster nodegroup.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternode

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusternode":{
    "nodeid":<Double_value>,
    "ipaddress":<String_value>,
    "state":<String_value>,
    "backplane":<String_value>,
    "priority":<Double_value>,
    "nodegroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternode

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusternode":{
    "nodeid":<Double_value>,
    "state":<String_value>,
    "backplane":<String_value>,
    "priority":<Double_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternode?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusternode":{
    "nodeid":<Double_value>,
    "state":true,
    "backplane":true,
    "priority":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternode/nodeid_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternode

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/clusternode?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternode?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of clusternode resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/clusternode?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternode?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternode resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternode": [ {
    "nodeid":<Double_value>,
    "ipaddress":<String_value>,
    "clusterhealth":<String_value>,
    "effectivestate":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "operationalsyncstate":<String_value>,
    "masterstate":<String_value>,
    "health":<String_value>,
    "state":<String_value>,
    "backplane":<String_value>,
    "priority":<Double_value>,
    "isconfigurationcoordinator":<Boolean_value>,
    "islocalnode":<Boolean_value>,
    "nodersskeymismatch":<Boolean_value>,
    "nodelicensemismatch":<Boolean_value>,
    "nodejumbonotsupported":<Boolean_value>,
    "nodelist":<Double[]_value>,
    "ifaceslist":<String[]_value>,
    "enabledifaces":<String_value>,
    "disabledifaces":<String_value>,
    "partialfailifaces":<String_value>,
    "hamonifaces":<String_value>,
    "nodegroup":<String_value>,
    "name":<String_value>,
    "cfgflags":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternode/nodeid_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/clusternode/nodeid_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/clusternode/nodeid_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternode": [ {
    "nodeid":<Double_value>,
    "ipaddress":<String_value>,
    "clusterhealth":<String_value>,
    "effectivestate":<String_value>,
    "operationalsyncstate":<String_value>,
    "masterstate":<String_value>,
    "health":<String_value>,
    "state":<String_value>,
    "backplane":<String_value>,
    "priority":<Double_value>,
    "isconfigurationcoordinator":<Boolean_value>,
    "islocalnode":<Boolean_value>,
    "nodersskeymismatch":<Boolean_value>,
    "nodelicensemismatch":<Boolean_value>,
    "nodejumbonotsupported":<Boolean_value>,
    "nodelist":<Double[]_value>,
    "ifaceslist":<String[]_value>,
    "enabledifaces":<String_value>,
    "disabledifaces":<String_value>,
    "partialfailifaces":<String_value>,
    "hamonifaces":<String_value>,
    "nodegroup":<String_value>,
    "name":<String_value>,
    "cfgflags":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternode?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{ "clusternode": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup

May 20, 2016

Configuration for Node group object type resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
strict	<String>	Read-write	Specifies whether cluster nodes, that are not part of the nodegroup, will be used as backup for the nodegroup. * Enabled - When one of the nodes goes down, no other cluster node is picked up to replace it. When the node comes up, it will continue being part of the nodegroup. * Disabled - When one of the nodes goes down, a non-nodegroup cluster node is picked up and acts as part of the nodegroup. When the original node of the nodegroup comes up, the backup node will be replaced. Default value: NO Possible values = YES, NO
sticky	<String>	Read-write	Only one node can be bound to nodegroup with this option enabled. It specifies whether to prempt the traffic for the entities bound to nodegroup when owner node goes down and rejoins the cluster. * Enabled - When owner node goes down, backup node will become the owner node and takes the traffic for the entities bound to the nodegroup. When bound node rejoins the cluster, traffic for the entities bound to nodegroup will not be steered back to this bound node. Current owner will have the ownership till it goes down. * Disabled - When one of the nodes goes down, a non-nodegroup cluster node is picked up and acts as part of the nodegroup. When the original node of the nodegroup comes up, the backup node will be replaced. Default value: NO Possible values = YES, NO
state	<String>	Read-write	State of the nodegroup. All the nodes binding to this nodegroup must have the same state. ACTIVE/SPARE/PASSIVE. Possible values = ACTIVE, SPARE, PASSIVE
priority	<Double>	Read-write	Priority of Nodegroup. This priority is used for all the nodes bound to the nodegroup for Nodegroup selection. Minimum value = 0 Maximum value = 31
currentnodemask	<Double>	Read-only	Bitmap of current nodes in this nodegroup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
backupnodemask	<Double>	Read-only	Bitmap of backup nodes in this nodegroup.
boundedentitiescntfrompe	<Double>	Read-only	Count of bounded entities to this nodegroup accoding to PE.
activelist	<Double[]>	Read-only	Active node list of this nodegroup.
backuplist	<Double[]>	Read-only	Backup node list of this nodegroup.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusternodegroup":{
    "name":<String_value>,
    "strict":<String_value>,
    "sticky":<String_value>,
    "state":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusternodegroup":{
    "name":<String_value>,
    "strict":<String_value>,
    "state":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"clusternodegroup":{
    "name":<String_value>,
    "strict":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of clusternodegroup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup": [ {
    "name":<String_value>,
    "strict":<String_value>,
    "sticky":<String_value>,
    "currentnodemask":<Double_value>,
    "backupnodemask":<Double_value>,
    "boundedentitiescntfrompe":<Double_value>,
    "activelist":<Double[]_value>,
    "backuplist":<Double[]_value>,
    "state":<String_value>,
    "priority":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup": [ {
    "name":<String_value>,
    "strict":<String_value>,
    "sticky":<String_value>,
    "currentnodemask":<Double_value>,
    "backupnodemask":<Double_value>,
    "boundedentitiescntfrompe":<Double_value>,
    "activelist":<Double[]_value>,
    "backuplist":<Double[]_value>,
    "state":<String_value>,
    "priority":<Double_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_authenticationvserver_binding

Jul 07, 2017

Binding object showing the authenticationvserver that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_authenticationvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_authenticationvserver_binding":{
    "name":<String_value>,
    "vserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_authenticationvserver_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_authenticationvserver_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_authenticationvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_authenticationvserver_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_authenticationvserver_binding": [ {
    "vserver":<String_value>,
    "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_authenticationvserver_binding?**args=&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "clusternodegroup_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_binding

May 20, 2016

Binding object which returns the resources bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the nodegroup to be displayed. If a name is not provided, information about all nodegroups is displayed. Minimum length = 1
clusternodegroup_csvserver_binding	<clusternodegroup_csvserver_binding[]>	Read-only	csvserver that can be bound to clusternodegroup.
clusternodegroup_gslbsite_binding	<clusternodegroup_gslbsite_binding[]>	Read-only	gslbsite that can be bound to clusternodegroup.
clusternodegroup_nslimitidentifier_binding	<clusternodegroup_nslimitidentifier_binding[]>	Read-only	nslimitidentifier that can be bound to clusternodegroup.
clusternodegroup_clusternode_binding	<clusternodegroup_clusternode_binding[]>	Read-only	clusternode that can be bound to clusternodegroup.
clusternodegroup_lbvserver_binding	<clusternodegroup_lbvserver_binding[]>	Read-only	lbvserver that can be bound to clusternodegroup.
clusternodegroup_vpnvserver_binding	<clusternodegroup_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to clusternodegroup.
clusternodegroup_service_binding	<clusternodegroup_service_binding[]>	Read-only	service that can be bound to clusternodegroup.
clusternodegroup_streamidentifier_binding	<clusternodegroup_streamidentifier_binding[]>	Read-only	streamidentifier that can be bound to clusternodegroup.
clusternodegroup_gslbvserver_binding	<clusternodegroup_gslbvserver_binding[]>	Read-only	gslbvserver that can be bound to clusternodegroup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_authenticationvserver_binding **Name**	<clusternodegroup_authenticationvserver_binding[]> **Data Type**	Read-only **Permissions**	that can be bound to clusternodegroup. **Description**
clusternodegroup_crvserver_binding	<clusternodegroup_crvserver_binding[]>	Read-only	crvserver that can be bound to clusternodegroup.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response payload provides details of the error

Response Payload:

```
{ "clusternodegroup_binding": [ {
    "name":<String_value>,
    "clusternodegroup_csvserver_binding":<clusternodegroup_csvserver_binding[]_value>,
    "clusternodegroup_vpnvserver_binding":<clusternodegroup_vpnvserver_binding[]_value>,
    "clusternodegroup_crvserver_binding":<clusternodegroup_crvserver_binding[]_value>,
    "clusternodegroup_nslimitidentifier_binding":<clusternodegroup_nslimitidentifier_binding[]_value>,
    "clusternodegroup_gslbsite_binding":<clusternodegroup_gslbsite_binding[]_value>,
    "clusternodegroup_clusternode_binding":<clusternodegroup_clusternode_binding[]_value>,
    "clusternodegroup_authenticationvserver_binding":<clusternodegroup_authenticationvserver_binding[]_value>,
    "clusternodegroup_lbvserver_binding":<clusternodegroup_lbvserver_binding[]_value>,
    "clusternodegroup_gslbvserver_binding":<clusternodegroup_gslbvserver_binding[]_value>,
    "clusternodegroup_streamidentifier_binding":<clusternodegroup_streamidentifier_binding[]_value>,
    "clusternodegroup_service_binding":<clusternodegroup_service_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_clusternode_binding

May 20, 2016

Binding object showing the clusternode that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
node	<Double>	Read-write	Nodes in the nodegroup. Minimum value = 0 Maximum value = 31
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_clusternode_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_clusternode_binding":{
    "name":<String_value>,
    "node":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_clusternode_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_clusternode_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_clusternode_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_clusternode_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_clusternode_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_clusternode_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_clusternode_binding": [ {
    "name":<String_value>,
    "node":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_clusternode_binding?**args=&count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "clusternodegroup_clusternode_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_crvserver_binding

Jul 07, 2017

Binding object showing the crvserver that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_crvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_crvserver_binding":{
    "name":<String_value>,
    "vserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_crvserver_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_crvserver_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_crvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_crvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_crvserver_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_crvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_crvserver_binding": [ {
    "vserver":<String_value>,
    "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_crvserver_binding?**args=&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "clusternodegroup_crvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_csvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_csvserver_binding":{
    "name":<String_value>,
    "vserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_csvserver_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_csvserver_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_csvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_csvserver_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_csvserver_binding": [ {
    "vserver":<String_value>,
    "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_csvserver_binding?**args=&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "clusternodegroup_csvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_gslbsite_binding

May 20, 2016

Binding object showing the gslbsite that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
gslbsite	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_gslbsite_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_gslbsite_binding":{
    "name":<String_value>,
    "gslbsite":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_gslbsite_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_gslbsite_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_gslbsite_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_gslbsite_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_gslbsite_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_gslbsite_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_gslbsite_binding": [ {
    "gslbsite":<String_value>,
    "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_gslbsite_binding?**args=&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "clusternodegroup_gslbsite_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_gslbvserver_binding

May 20, 2016

Binding object showing the gslbvserver that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_gslbvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_gslbvserver_binding":{
    "name":<String_value>,
    "vserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_gslbvserver_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_gslbvserver_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_gslbvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_gslbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_gslbvserver_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_gslbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_gslbvserver_binding": [ {
    "vserver":<String_value>,
    "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_gslbvserver_binding?**args=&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "clusternodegroup_gslbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_lbvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_lbvserver_binding":{
    "name":<String_value>,
    "vserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_lbvserver_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_lbvserver_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_lbvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_lbvserver_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_lbvserver_binding": [ {
    "vserver":<String_value>,
    "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_lbvserver_binding?**args=&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "clusternodegroup_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_nslimitidentifier_binding

May 20, 2016

Binding object showing the nslimitidentifier that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the nodegroup to which you want to bind a cluster node or an entity. Minimum length = 1
identifiername	<String>	Read-write	stream identifier and rate limit identifier that need to be bound to this nodegroup.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_nslimitidentifier_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_nslimitidentifier_binding":{
    "name":<String_value>,
    "identifiername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of clusternodegroup_nslimitidentifier_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_nslimitidentifier_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_nslimitidentifier_binding": [ {
      "name":<String_value>,
      "identifiername":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_nslimitidentifier_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"clusternodegroup_nslimitidentifier_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_service_binding

May 20, 2016

Binding object showing the service that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
service	<String>	Read-write	name of the service bound to this nodegroup.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_service_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_service_binding":{
    "name":<String_value>,
    "service":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_service_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_service_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_service_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_service_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_service_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_service_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_service_binding": [ {
      "name":<String_value>,
      "service":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_service_binding?**args=&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "clusternodegroup_service_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_streamidentifier_binding

May 20, 2016

Binding object showing the streamidentifier that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the nodegroup to which you want to bind a cluster node or an entity. Minimum length = 1
identifiername	<String>	Read-write	stream identifier and rate limit identifier that need to be bound to this nodegroup.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_streamidentifier_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_streamidentifier_binding":{
    "name":<String_value>,
    "identifiername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of clusternodegroup_streamidentifier_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_streamidentifier_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_streamidentifier_binding": [ {
      "name":<String_value>,
      "identifiername":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_streamidentifier_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"clusternodegroup_streamidentifier_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternodegroup_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to clusternodegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<String>	Read-write	vserver that need to be bound to this nodegroup.
name	<String>	Read-write	Name of the nodegroup. The name uniquely identifies the nodegroup on the cluster. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/clusternodegroup_vpnvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"clusternodegroup_vpnvserver_binding":{
    "name":<String_value>,
    "vserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_vpnvserver_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_vpnvserver_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_vpnvserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of clusternodegroup_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_vpnvserver_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the clusternodegroup_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternodegroup_vpnvserver_binding": [ {
    "vserver":<String_value>,
    "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/clusternodegroup_vpnvserver_binding?**args=&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "clusternodegroup_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

clustersync

May 20, 2016

Configuration for sync resource.

Operations (click to see Properties)

FORCE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Force

URL: http://<netscaler-ip-address>/nitro/v1/config/clustersync?**action=Force**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"clustersync":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Compression

May 20, 2016

Compression configuration. The system's feature for compressing HTTP responses to compression-aware browsers.

Name	Description
cmpaction	Configuration for compression action
cmpglobal_binding	Binding object showing the resources that can be bound to cmpglobal
cmpglobal_cmppolicy_binding	Binding object showing the cmppolicy that can be bound to cmpglobal
cmpparameter	Configuration for CMP parameter
cmppolicy	Configuration for compression policy
cmppolicy_binding	Binding object showing the resources that can be bound to cmppolicy
cmppolicy_cmpglobal_binding	Binding object showing the cmpglobal that can be bound to cmppolicy
cmppolicy_cmppolicylabel_binding	Binding object showing the cmppolicylabel that can be bound to cmppolicy
cmppolicy_crvserver_binding	Binding object showing the crvserver that can be bound to cmppolicy
cmppolicy_csvserver_binding	Binding object showing the csvserver that can be bound to cmppolicy
cmppolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to cmppolicy
cmppolicylabel	Configuration for compression policy label
cmppolicylabel_binding	Binding object showing the resources that can be bound to cmppolicylabel
cmppolicylabel_cmppolicy_binding	Binding object showing the cmppolicy that can be bound to cmppolicylabel
cmppolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to cmppolicylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmpaction

May 20, 2016

Configuration for compression action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the compression action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the action is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp action" or 'my cmp action'). Minimum length = 1
cmptype	<String>	Read-write	Type of compression performed by this action. Available settings function as follows: * COMPRESS - Apply GZIP or DEFLATE compression to the response, depending on the request header. Prefer GZIP. * GZIP - Apply GZIP compression. * DEFLATE - Apply DEFLATE compression. * NOCOMPRESS - Do not compress the response if the request matches a policy that uses this action. Possible values = compress, gzip, deflate, nocompress
addvaryheader	<String>	Read-write	Control insertion of the Vary header in HTTP responses compressed by NetScaler. Intermediate caches store different versions of the response for different values of the headers present in the Vary response header. Default value: GLOBAL Possible values = GLOBAL, DISABLED, ENABLED
varyheadervalue	<String>	Read-write	The value of the HTTP Vary header for compressed responses. Minimum length = 1
deltatype	<String>	Read-write	The type of delta action (if delta type compression action is defined). Default value: PERURL Possible values = PERURL, PERPOLICY
newname	<String>	Read-write	New name for the compression action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Choose a name that can be correlated with the function that the action performs.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp action" or 'my cmp action'). Minimum length = 1
builtin	<String[]>	Read-only	Flag to determine whether compression is default or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default policy.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpaction

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cmpaction":{
    "name":<String_value>,
    "cmptype":<String_value>,
    "addvaryheader":<String_value>,
    "varyheadervalue":<String_value>,
    "deltatype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"cmpaction":{
    "name":<String_value>,
    "cmptype":<String_value>,
    "addvaryheader":<String_value>,
    "varyheadervalue":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cmpaction":{
    "name":<String_value>,
    "addvaryheader":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpaction?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"cmpaction":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cmpaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cmpaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cmpaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/cmpaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmpaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmpaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmpaction": [ {
    "name":<String_value>,
    "cmptype":<String_value>,
    "deltatype":<String_value>,
    "addvaryheader":<String_value>,
    "varyheadervalue":<String_value>,
    "builtin":<String[]_value>,
    "isdefault":<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cmpaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/cmpaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmpaction": [ {
    "name":<String_value>,
    "cmptype":<String_value>,
    "deltatype":<String_value>,
    "addvaryheader":<String_value>,
    "varyheadervalue":<String_value>,
    "builtin":<String[]_value>,
    "isdefault":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmpaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmpglobal_binding

May 20, 2016

Binding object which returns the resources bound to cmpglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cmpglobal_cmppolicy_binding	<cmpglobal_cmppolicy_binding[]>	Read-only	cmppolicy that can be bound to cmpglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "cmpglobal_binding": [ {
      "cmpglobal_cmppolicy_binding":<cmpglobal_cmppolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmpglobal_cmppolicy_binding

May 20, 2016

Binding object showing the cmppolicy that can be bound to cmpglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Positive integer specifying the priority of the policy. The lower the number, the higher the priority. By default, polices within a label are evaluated in the order of their priority numbers. In the configuration utility, you can click the Priority field and edit the priority level or drag the entry to a new position in the list. If you drag the entry to a new position, the priority level is updated automatically.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
state	<String>	Read-write	The current state of the policy binding. This attribute is relevant only for CLASSIC policies. Possible values = ENABLED, DISABLED
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the priority of the next policy, within the policy label, to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher numbered priority. * END - Stop evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * An expression that evaluates to a number. If you specify an expression, it's evaluation result determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, that policy is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher priority number is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest priority number, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	The name of the globally bound HTTP compression policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
type	<String>	Read-write	Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
labeltype	<String>	Read-write	Type of policy label invocation. This argument is relevant only for advanced (default-syntax) policies. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE. Applicable only to advanced (default-syntax) policies.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority. Applicable only for default-syntax policies.
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
policytype	<String>	Read-only	Policy type (Classic/Advanced) to be bound.Used for display. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/cmpglobal_cmppolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"cmpglobal_cmppolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "state":<String_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpglobal_cmppolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpglobal_cmppolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cmpglobal_cmppolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmpglobal_cmppolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmpglobal_cmppolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmpglobal_cmppolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmpglobal_cmppolicy_binding": [ {
      "priority":<Double_value>,
      "globalbindtype":<String_value>,
      "state":<String_value>,
      "gotopriorityexpression":<String_value>,
      "policyname":<String_value>,
      "type":<String_value>,
      "labeltype":<String_value>,
      "labelname":<String_value>,
      "invoke":<Boolean_value>,
      "numpol":<Double_value>,
      "policytype":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpglobal_cmppolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "cmpglobal_cmppolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmpparameter

Jul 07, 2017

Configuration for CMP parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cmplevel	<String>	Read-write	Specify a compression level. Available settings function as follows: * Optimal - Corresponds to a gzip GZIP level of 5-7. * Best speed - Corresponds to a gzip level of 1. * Best compression - Corresponds to a gzip level of 9. Default value: optimal Possible values = optimal, bestspeed, bestcompression
quantumsize	<Double>	Read-write	Minimum quantum of data to be filled before compression begins. Default value: 57344 Minimum value = 8 Maximum value = 63488
servercmp	<String>	Read-write	Allow the server to send compressed data to the NetScaler appliance. With the default setting, the NetScaler appliance handles all compression. Default value: ON Possible values = ON, OFF
heurexpiry	<String>	Read-write	Heuristic basefile expiry. Default value: OFF Possible values = ON, OFF
heurexpirythres	<Double>	Read-write	Threshold compression ratio for heuristic basefile expiry, multiplied by 100. For example, to set the threshold ratio to 1.25, specify 125. Default value: 100 Minimum value = 1 Maximum value = 1000
heurexpiryhistwt	<Double>	Read-write	For heuristic basefile expiry, weightage to be given to historical delta compression ratio, specified as percentage. For example, to give 25% weightage to historical ratio (and therefore 75% weightage to the ratio for current delta compression transaction), specify 25. Default value: 50 Minimum value = 1 Maximum value = 100
minressize	<Double>	Read-write	Smallest response size, in bytes, to be compressed.
			NetScaler CPU threshold after which compression is not performed. Range: 0 - 100.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 0 Maximum value = 100
cmponpush	<String>	Read-write	NetScaler appliance does not wait for the quantum to be filled before starting to compress data. Upon receipt of a packet with a PUSH flag, the appliance immediately begins compression of the accumulated packets. Default value: DISABLED Possible values = ENABLED, DISABLED
policytype	<String>	Read-write	Type of policy. Available settings function as follows: * Classic - Classic policies evaluate basic characteristics of traffic and other data. * Advanced - Advanced policies (which have been renamed as default syntax policies) can perform the same type of evaluations as classic policies. They also enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example, transforming data in the body of a request into an HTTP header). Default value: CLASSIC Possible values = CLASSIC, ADVANCED
addvaryheader	<String>	Read-write	Control insertion of the Vary header in HTTP responses compressed by NetScaler. Intermediate caches store different versions of the response for different values of the headers present in the Vary response header. Default value: DISABLED Possible values = ENABLED, DISABLED
varyheadervalue	<String>	Read-write	The value of the HTTP Vary header for compressed responses. If this argument is not specified, a default value of "Accept-Encoding" will be used. Minimum length = 1
externalcache	<String>	Read-write	Enable insertion of Cache-Control: private response directive to indicate response message is intended for a single user and must not be cached by a shared or proxy cache. Default value: NO Possible values = YES, NO
builtin	<String[]>	Read-only	Flag to determine whether compression is default or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cmpparameter":{
    "cmplevel":<String_value>,
    "quantumsize":<Double_value>,
    "servercmp":<String_value>,
    "heurexpiry":<String_value>,
    "heurexpirythres":<Double_value>,
    "heurexpiryhistwt":<Double_value>,
    "minressize":<Double_value>,
    "cmpbypasspct":<Double_value>,
    "cmponpush":<String_value>,
    "policytype":<String_value>,
    "addvaryheader":<String_value>,
    "varyheadervalue":<String_value>,
    "externalcache":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cmpparameter":{
    "cmplevel":true,
    "quantumsize":true,
    "servercmp":true,
    "heurexpiry":true,
    "heurexpirythres":true,
    "heurexpiryhistwt":true,
    "minressize":true,
    "cmpbypasspct":true,
    "cmponpush":true,
    "policytype":true,
    "addvaryheader":true,
    "varyheadervalue":true,
    "externalcache":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cmpparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "cmpparameter": [ {
    "cmplevel":<String_value>,
    "quantumsize":<Double_value>,
    "servercmp":<String_value>,
    "heurexpiry":<String_value>,
    "heurexpirythres":<Double_value>,
    "heurexpiryhistwt":<Double_value>,
    "minressize":<Double_value>,
    "cmpbypasspct":<Double_value>,
    "cmponpush":<String_value>,
    "policytype":<String_value>,
    "addvaryheader":<String_value>,
    "varyheadervalue":<String_value>,
    "externalcache":<String_value>,
    "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicy

May 20, 2016

Configuration for compression policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the HTTP compression policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp policy" or 'my cmp policy'). Minimum length = 1
rule	<String>	Read-write	Expression that determines which HTTP requests or responses match the compression policy. Can be a classic expression or a default-syntax expression. Note: Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
resaction	<String>	Read-write	The built-in or user-defined compression action to apply to the response when the policy matches a request or response. Minimum length = 1
newname	<String>	Read-write	New name for the compression policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Choose a name that reflects the function that the policy performs. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp policy" or 'my cmp policy').

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced) . Possible values = Classic Policy, Advanced Policy
reqaction	<String>	Read-only	The compression action to be performed on requests.
hits	<Double>	Read-only	Number of hits.
txbytes	<Double>	Read-only	Number of bytes transferred.
rxbytes	<Double>	Read-only	Number of bytes received.
clientttlb	<Double>	Read-only	Total client TTLB value.
clienttransactions	<Double>	Read-only	Number of client transactions.
serverttlb	<Double>	Read-only	Total server TTLB value.
servertransactions	<Double>	Read-only	Number of server transactions.
description	<String>	Read-only	Description of the policy.
builtin	<String[]>	Read-only	Flag to determine if compression policy is builtin or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default policy.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cmppolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "resaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cmppolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "resaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cmppolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cmppolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmppolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicy": [ {
    "name":<String_value>,
    "expressiontype":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>,
    "resaction":<String_value>,
    "hits":<Double_value>,
    "txbytes":<Double_value>,
    "rxbytes":<Double_value>,
    "clientttlb":<Double_value>,
    "clienttransactions":<Double_value>,
    "serverttlb":<Double_value>,
    "servertransactions":<Double_value>,
    "description":<String_value>,
    "builtin":<String[]_value>,
    "isdefault":<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicy": [ {
    "name":<String_value>,
    "expressiontype":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>,
    "resaction":<String_value>,
    "hits":<Double_value>,
    "txbytes":<Double_value>,
    "rxbytes":<Double_value>,
    "clientttlb":<Double_value>,
    "clienttransactions":<Double_value>,
    "serverttlb":<Double_value>,
    "servertransactions":<Double_value>,
    "description":<String_value>,
    "builtin":<String[]_value>,
    "isdefault":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicylabel

May 20, 2016

Configuration for compression policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the HTTP compression policy label. Must begin with a letter, number, or the underscore character (_). Additional characters allowed, after the first character, are the hyphen (-), period (.) pound sign (#), space ( ), at sign (@), equals (=), and colon (:). The name must be unique within the list of policy labels for compression policies. Can be renamed after the policy label is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp policylabel" or 'my cmp policylabel'). Minimum length = 1
type	<String>	Read-write	Type of packets (request packets or response) against which to match the policies bound to this policy label. Possible values = REQ, RES
newname	<String>	Read-write	New name for the compression policy label. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cmp policylabel" or 'my cmp policylabel'). Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
flowtype	<Double>	Read-only	Flowtype of the bound compression policy.
description	<String>	Read-only	Description of the policylabel.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cmppolicylabel":{
    "labelname":<String_value>,
    "type":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cmppolicylabel":{
    "labelname":<String_value>,
    "newname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cmppolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmppolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicylabel": [ {
      "labelname":<String_value>,
      "type":<String_value>,
      "numpol":<Double_value>,
      "hits":<Double_value>,
      "priority":<Double_value>,
      "gotoprioritiyexpression":<String_value>,
      "labeltype":<String_value>,
      "invoke_labelname":<String_value>,
      "flowtype":<Double_value>,
      "description":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicylabel": [ {
    "labelname":<String_value>,
    "type":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "flowtype":<Double_value>,
    "description":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicylabel": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicylabel_binding

May 20, 2016

Binding object which returns the resources bound to cmppolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the HTTP compression policy label for which to display details. Minimum length = 1
cmppolicylabel_cmppolicy_binding	<cmppolicylabel_cmppolicy_binding[]>	Read-only	cmppolicy that can be bound to cmppolicylabel.
cmppolicylabel_policybinding_binding	<cmppolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to cmppolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "cmppolicylabel_binding": [ {
    "labelname":<String_value>,
    "cmppolicylabel_policybinding_binding":<cmppolicylabel_policybinding_binding[]_value>,
    "cmppolicylabel_cmppolicy_binding":<cmppolicylabel_cmppolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicylabel_cmppolicy_binding

Jul 07, 2017

Binding object showing the cmppolicy that can be bound to cmppolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	The compression policy name.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the HTTP compression policy label to which to bind the policy. Minimum length = 1
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next higher priority number in the original label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/cmppolicylabel_cmppolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"cmppolicylabel_cmppolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicylabel_cmppolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmppolicylabel_cmppolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "cmppolicylabel_cmppolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel_cmppolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cmppolicylabel_cmppolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicylabel_policybinding_binding

May 20, 2016

Binding object showing the policybinding that can be bound to cmppolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	The compression policy name.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the HTTP compression policy label to which to bind the policy. Minimum length = 1
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or a user-defined policy label. After the invoked policies are evaluated, the flow returns to the policy with the next higher priority number in the original label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicylabel_policybinding_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmppolicylabel_policybinding_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicylabel_policybinding_binding": [ {
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "policyname":<String_value>,
      "labeltype":<String_value>,
      "labelname":<String_value>,
      "invoke_labelname":<String_value>,
      "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cmppolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicy_binding

May 20, 2016

Binding object which returns the resources bound to cmppolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
cmppolicy_lbvserver_binding	<cmppolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to cmppolicy.
cmppolicy_cmpglobal_binding	<cmppolicy_cmpglobal_binding[]>	Read-only	cmpglobal that can be bound to cmppolicy.
cmppolicy_crvserver_binding	<cmppolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to cmppolicy.
cmppolicy_csvserver_binding	<cmppolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to cmppolicy.
cmppolicy_cmppolicylabel_binding	<cmppolicy_cmppolicylabel_binding[]>	Read-only	cmppolicylabel that can be bound to cmppolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "cmppolicy_binding": [ {
    "name":<String_value>,
    "cmppolicy_crvserver_binding":<cmppolicy_crvserver_binding[]_value>,
    "cmppolicy_csvserver_binding":<cmppolicy_csvserver_binding[]_value>,
    "cmppolicy_cmpglobal_binding":<cmppolicy_cmpglobal_binding[]_value>,
    "cmppolicy_lbvserver_binding":<cmppolicy_lbvserver_binding[]_value>,
    "cmppolicy_cmppolicylabel_binding":<cmppolicy_cmppolicylabel_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicy_cmpglobal_binding

Jul 07, 2017

Binding object showing the cmpglobal that can be bound to cmppolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_cmpglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_cmpglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicy_cmpglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_cmpglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmppolicy_cmpglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicy_cmpglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_cmpglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cmppolicy_cmpglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicy_cmppolicylabel_binding

May 20, 2016

Binding object showing the cmppolicylabel that can be bound to cmppolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_cmppolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_cmppolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicy_cmppolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_cmppolicylabel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmppolicy_cmppolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicy_cmppolicylabel_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_cmppolicylabel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cmppolicy_cmppolicylabel_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicy_crvserver_binding

May 20, 2016

Binding object showing the crvserver that can be bound to cmppolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_crvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_crvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicy_crvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_crvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmppolicy_crvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicy_crvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_crvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cmppolicy_crvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to cmppolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmppolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicy_csvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cmppolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to cmppolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the HTTP compression policy for which to display details. Minimum length = 1
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cmppolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cmppolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicy_lbvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cmppolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cmppolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cache Redirection

May 20, 2016

Cache redirection configuration. The cache redirection feature can transparently redirect cacheable HTTP requests to a cache and send non-cacheable or dynamic HTTP requests to the origin server. A cache stores or caches frequently requested web content and serves such web content to a client on behalf of the origin servers, alleviating the load on the origin server farm.

Name	Description
craction	Configuration for cache redirection action
crpolicy	Configuration for cache redirection policy
crpolicy_binding	Binding object showing the resources that can be bound to crpolicy
crpolicy_crvserver_binding	Binding object showing the crvserver that can be bound to crpolicy
crvserver	Configuration for CR virtual server
crvserver_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to crvserver
crvserver_appfwpolicy_binding	Binding object showing the appfwpolicy that can be bound to crvserver
crvserver_binding	Binding object showing the resources that can be bound to crvserver
crvserver_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to crvserver
crvserver_cmppolicy_binding	Binding object showing the cmppolicy that can be bound to crvserver
crvserver_crpolicy_binding	Binding object showing the crpolicy that can be bound to crvserver
crvserver_cspolicy_binding	Binding object showing the cspolicy that can be bound to crvserver
crvserver_feopolicy_binding	Binding object showing the feopolicy that can be bound to crvserver
crvserver_filterpolicy_binding	Binding object showing the filterpolicy that can be bound to crvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_icapolicy_binding	
Name	**Description**
crvserver_lbvserver_binding	Binding object showing the lbvserver that can be bound to crvserver
crvserver_policymap_binding	Binding object showing the policymap that can be bound to crvserver
crvserver_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to crvserver
crvserver_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to crvserver
crvserver_spilloverpolicy_binding	Binding object showing the spilloverpolicy that can be bound to crvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

craction

May 20, 2016

Configuration for cache redirection action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the action for which to display detailed information. Minimum length = 1
crtype	<String>	Read-only	Type to decide where to redirect the requests if the cache redirection policy is hit. The valid options are as follows: *CACHE - Directs all the requests to the cache if cache redirection policy is hit. *ORIGIN - Directs all requests to the origin server if the cache redirection policy is hit. Possible values = origin, cache
builtin	<String[]>	Read-only	Flag to determine whether CRACTION is default or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default cr action.
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<String>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
comment	<String>	Read-only	Comment. Any type of information about this responder action.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/craction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/craction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/craction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of craction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/craction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/craction?**pagesize=#no&pageno=#no**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the craction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "craction": [ {
    "name":<String_value>,
    "crtype":<String_value>,
    "builtin":<String[]_value>,
    "isdefault":<Boolean_value>,
    "hits":<Double_value>,
    "referencecount":<String_value>,
    "undefhits":<Double_value>,
    "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/craction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/craction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/craction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "craction": [ {
      "name":<String_value>,
      "crtype":<String_value>,
      "builtin":<String[]_value>,
      "isdefault":<Boolean_value>,
      "hits":<Double_value>,
      "referencecount":<String_value>,
      "undefhits":<Double_value>,
      "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/craction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "craction": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crpolicy

May 20, 2016

Configuration for cache redirection policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name for the cache redirection policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my policy? or ?my policy?). Minimum length = 1
rule	<String>	Read-write	Expression, or name of a named expression, against which traffic is evaluated. Written in the classic syntax. Note:Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: "<string of 255 characters>" + "<string of 245 characters>" The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the built-in cache redirection action: CACHE/ORIGIN.
logaction	<String>	Read-write	The log action associated with the cache redirection policy.
newname	<String>	Read-write	The new name of the content switching policy. Minimum length = 1
domain	<String>	Read-only	Domain name.
vstype	<Double>	Read-only	Virtual server type.
hits	<Double>	Read-only	Total number of hits.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
bindhits	<Double>	Read-only	Total number of hits.
priority	<Double>	Read-only	priority of bound policy.
activepolicy	<Boolean>	Read-only	Indicates whether policy is bound or not.
cspolicytype	<String>	Read-only	Indicates whether policy is PI or not.(used only during display). Possible values = Classic Policy, Advanced Policy
labelname	<String>	Read-only	Name of the label invoked.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
builtin	<String[]>	Read-only	. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default cr policy.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"crpolicy":{
    "policyname":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy/policyname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"crpolicy":{
    "policyname":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"crpolicy":{
    "policyname":<String_value>,
    "logaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"crpolicy":{
    "policyname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/crpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/crpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of crpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/crpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/crpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crpolicy": [ {
    "policyname":<String_value>,
    "rule":<String_value>,
    "domain":<String_value>,
    "action":<String_value>,
    "vstype":<Double_value>,
    "hits":<Double_value>,
    "bindhits":<Double_value>,
    "priority":<Double_value>,
    "activepolicy":<Boolean_value>,
    "cspolicytype":<String_value>,
    "logaction":<String_value>,
    "labelname":<String_value>,
    "labeltype":<String_value>,
    "builtin":<String[]_value>,
    "isdefault":<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy/policyname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/crpolicy/policyname_value<String>?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/crpolicy/policyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crpolicy": [ {
      "policyname":<String_value>,
      "rule":<String_value>,
      "domain":<String_value>,
      "action":<String_value>,
      "vstype":<Double_value>,
      "hits":<Double_value>,
      "bindhits":<Double_value>,
      "priority":<Double_value>,
      "activepolicy":<Boolean_value>,
      "cspolicytype":<String_value>,
      "logaction":<String_value>,
      "labelname":<String_value>,
      "labeltype":<String_value>,
      "builtin":<String[]_value>,
      "isdefault":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "crpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

crpolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to crpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the cache redirection policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
crpolicy_crvserver_binding	<crpolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to crpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy_binding/policyname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "crpolicy_binding": [ {
    "policyname":<String_value>,
    "crpolicy_crvserver_binding":<crpolicy_crvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

crpolicy_crvserver_binding

May 20, 2016

Binding object showing the crvserver that can be bound to crpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the cache redirection policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
domain	<String>	Read-write	Domain name.
priority	<Double>	Read-only	priority of bound policy.
hits	<Double>	Read-only	Total number of hits.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
pipolicyhits	<Double>	Read-only	bind hits for PI CR Policy.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
pihits	<Double>	Read-only	Total number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy_crvserver_binding/policyname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crpolicy_crvserver_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crpolicy_crvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crpolicy_crvserver_binding/policyname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crpolicy_crvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crpolicy_crvserver_binding": [ {
    "policyname":<String_value>,
    "domain":<String_value>,
    "priority":<Double_value>,
    "hits":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "pipolicyhits":<Double_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "pihits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crpolicy_crvserver_binding/policyname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"crpolicy_crvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver

May 20, 2016

Configuration for CR virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the cache redirection virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the cache redirection virtual server is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my server? or ?my server?). Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
servicetype	<String>	Read-write	Protocol (type of service) handled by the virtual server. Possible values = HTTP, SSL, NNTP, HDX
ipv46	<String>	Read-write	IPv4 or IPv6 address of the cache redirection virtual server. Usually a public IP address. Clients send connection requests to this IP address. Note: For a transparent cache redirection virtual server, use an asterisk (*) to specify a wildcard virtual server address.
port	<Integer>	Read-write	Port number of the virtual server. Default value: 80 Minimum value = 1 Maximum value = 65534
range	<Double>	Read-write	Number of consecutive IP addresses, starting with the address specified by the IPAddress parameter, to include in a range of addresses assigned to this virtual server. Default value: 1 Minimum value = 1 Maximum value = 254
			Mode of operation for the cache redirection virtual server. Available settings function as follows: * TRANSPARENT - Intercept all traffic flowing to the appliance and apply cache

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cachetype	<String>	Read-write	cache or from the origin server. * FORWARD - Resolve the hostname of the incoming request, by using a DNS server, and forward requests for non-cacheable content to the resolved origin servers. Cacheable requests are sent to the configured cache servers. * REVERSE - Configure reverse proxy caches for specific origin servers. Incoming traffic directed to the reverse proxy can either be served from a cache server or be sent to the origin server with or without modification to the URL. Default value: TRANSPARENT Possible values = TRANSPARENT, REVERSE, FORWARD
redirect	<String>	Read-write	Type of cache server to which to redirect HTTP requests. Available settings function as follows: * CACHE - Direct all requests to the cache. * POLICY - Apply the cache redirection policy to determine whether the request should be directed to the cache or to the origin. * ORIGIN - Direct all requests to the origin server. Default value: POLICY Possible values = CACHE, POLICY, ORIGIN
onpolicymatch	<String>	Read-write	Redirect requests that match the policy to either the cache or the origin server, as specified. Note: For this option to work, you must set the cache redirection type to POLICY. Default value: ORIGIN Possible values = CACHE, ORIGIN
redirecturl	<String>	Read-write	URL of the server to which to redirect traffic if the cache redirection virtual server configured on the NetScaler appliance becomes unavailable. Minimum length = 1 Maximum length = 128
clttimeout	<Double>	Read-write	Time-out value, in seconds, after which to terminate an idle client connection. Minimum value = 0 Maximum value = 31536000
precedence	<String>	Read-write	Type of policy (URL or RULE) that takes precedence on the cache redirection virtual server. Applies only to cache redirection virtual servers that have both URL and RULE based policies. If you specify URL, URL based policies are applied first, in the following order: 1. Domain and exact URL 2. Domain, prefix and suffix 3. Domain and suffix 4. Domain and prefix 5. Domain only 6. Exact URL 7. Prefix and suffix 8. Suffix only 9. Prefix only 10. Default If you specify RULE, the rule based policies are applied before URL based policies are applied. Default value: RULE Possible values = RULE, URL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
arp	<String>	Read-write	Possible values = ON, OFF
ghost	<String>	Read-write	. Possible values = ON, OFF
map	<String>	Read-write	Obsolete. Possible values = ON, OFF
format	<String>	Read-write	. Possible values = ON, OFF
via	<String>	Read-write	Insert a via header in each HTTP request. In the case of a cache miss, the request is redirected from the cache server to the origin server. This header indicates whether the request is being sent from a cache server. Default value: ON Possible values = ON, OFF
cachevserver	<String>	Read-write	Name of the default cache virtual server to which to redirect requests (the default target of the cache redirection virtual server). Minimum length = 1
dnsvservername	<String>	Read-write	Name of the DNS virtual server that resolves domain names arriving at the forward proxy virtual server. Note: This parameter applies only to forward proxy virtual servers, not reverse or transparent. Minimum length = 1
destinationvserver	<String>	Read-write	Destination virtual server for a transparent or forward proxy cache redirection virtual server. Minimum length = 1
domain	<String>	Read-write	Default domain for reverse proxies. Domains are configured to direct an incoming request from a specified source domain to a specified target domain. There can be several configured pairs of source and target domains. You can select one pair to be the default. If the host header or URL of an incoming request does not include a source domain, this option sends the request to the specified target domain. Minimum length = 1
sopersistencetimeout	<Double>	Read-write	Time-out, in minutes, for spillover persistence. Minimum value = 2 Maximum value = 24
sothreshold	<Double>	Read-write	For CONNECTION (or) DYNAMICCONNECTION spillover, the number of connections above which the virtual server enters spillover mode. For BANDWIDTH spillover, the amount of incoming and outgoing traffic (in Kbps) before spillover. For HEALTH spillover, the percentage of active services (by weight) below which spillover occurs. Minimum value = 1
			Reuse TCP connections to the origin server across client connections. Do not set this parameter unless the Service Type parameter is set to HTTP. If you set this parameter to OFF, the possible settings of the Redirect parameter function

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
reuse	<String>	Read-write	* CACHE - TCP connections to the cache servers are not reused. * ORIGIN - TCP connections to the origin servers are not reused. * POLICY - TCP connections to the origin servers are not reused. If you set the Reuse parameter to ON, connections to origin servers and connections to cache servers are reused. Default value: ON Possible values = ON, OFF
state	<String>	Read-write	Initial state of the cache redirection virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
downstateflush	<String>	Read-write	Perform delayed cleanup of connections to this virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
backupvserver	<String>	Read-write	Name of the backup virtual server to which traffic is forwarded if the active server becomes unavailable. Minimum length = 1
disableprimaryondown	<String>	Read-write	Continue sending traffic to a backup virtual server even after the primary virtual server comes UP from the DOWN state. Default value: DISABLED Possible values = ENABLED, DISABLED
l2conn	<String>	Read-write	Use L2 parameters, such as MAC, VLAN, and channel to identify a connection. Possible values = ON, OFF
backendssl	<String>	Read-write	Decides whether the backend connection made by NS to the origin server will be HTTP or SSL. Applicable only for SSL type CR Forward proxy vserver. Default value: DISABLED Possible values = ENABLED, DISABLED
listenpolicy	<String>	Read-write	String specifying the listen policy for the cache redirection virtual server. Can be either an in-line expression or the name of a named expression. Default value: "NONE"
listenpriority	<Double>	Read-write	Priority of the listen policy specified by the Listen Policy parameter. The lower the number, higher the priority. Default value: 101 Minimum value = 0 Maximum value = 100
tcpprofilename	<String>	Read-write	Name of the profile containing TCP configuration information for the cache redirection virtual server. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the profile containing HTTP configuration information for cache redirection virtual server. Minimum length = 1 Maximum length = 127

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum length = 256
srcipexpr	<String>	Read-write	Expression used to extract the source IP addresses from the requests originating from the cache. Can be either an in-line expression or the name of a named expression. Minimum length = 1 Maximum length = 1500
originusip	<String>	Read-write	Use the client?s IP address as the source IP address in requests sent to the origin server. Note: You can enable this parameter to implement fully transparent CR deployment. Possible values = ON, OFF
useportrange	<String>	Read-write	Use a port number from the port range (set by using the set ns param command, or in the Create Virtual Server (Cache Redirection) dialog box) as the source port in the requests sent to the origin server. Default value: OFF Possible values = ON, OFF
appflowlog	<String>	Read-write	Enable logging of AppFlow information. Default value: ENABLED Possible values = ENABLED, DISABLED
netprofile	<String>	Read-write	Name of the network profile containing network configurations for the cache redirection virtual server. Minimum length = 1 Maximum length = 127
icmpvsrresponse	<String>	Read-write	Criterion for responding to PING requests sent to this virtual server. If ACTIVE, respond only if the virtual server is available. If PASSIVE, respond even if the virtual server is not available. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
rhistate	<String>	Read-write	A host route is injected according to the setting on the virtual servers * If set to PASSIVE on all the virtual servers that share the IP address, the appliance always injects the hostroute. * If set to ACTIVE on all the virtual servers that share the IP address, the appliance injects even if one virtual server is UP. * If set to ACTIVE on some virtual servers and PASSIVE on the others, the appliance, injects even if one virtual server set to ACTIVE is UP. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
newname	<String>	Read-write	New name for the cache redirection virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my name? or ?my name?). Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
value	<String>	Read-only	The ssl card status for the transparent ssl cr vserver. Possible values = Certkey not bound, SSL feature disabled
ngname	<String>	Read-only	Nodegroup devno to which this crvserver belongs to.
type	<String>	Read-only	Virtual server type. Possible values = CONTENT, ADDRESS
curstate	<String>	Read-only	The state of the cr vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
status	<Integer>	Read-only	Status.
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
authentication	<String>	Read-only	Authentication. Possible values = ON, OFF
homepage	<String>	Read-only	Home page.
rule	<String>	Read-only	Rule.
policyname	<String>	Read-only	Policies bound to this vserver.
pipolicyhits	<Double>	Read-only	Number of hits.
servicename	<String>	Read-only	Service name.
weight	<Double>	Read-only	Weight for this service.
targetvserver	<String>	Read-only	The CSW target server names.
priority	<Double>	Read-only	The priority for the policy.
somethod	<String>	Read-only	The spillover factor. When the main virtual server reaches this spillover threshold, it will give further traffic to the backupvserver. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sopersistence	<String>	Read-only	The state of spillover persistence.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
lbvserver	<String>	Read-only	The Default target server name. Minimum length = 1
bindpoint	<String>	Read-only	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
invoke	<Boolean>	Read-only	Invoke flag.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"crvserver":{
    "name":<String_value>,
    "td":<Double_value>,
    "servicetype":<String_value>,
    "ipv46":<String_value>,
    "port":<Integer_value>,
    "range":<Double_value>,
    "cachetype":<String_value>,
    "redirect":<String_value>,
    "onpolicymatch":<String_value>,
    "redirecturl":<String_value>,
    "clttimeout":<Double_value>,
    "precedence":<String_value>,
    "arp":<String_value>,
    "ghost":<String_value>,
    "map":<String_value>,
    "format":<String_value>,
    "via":<String_value>,
    "cachevserver":<String_value>,
    "dnsvservername":<String_value>,
    "destinationvserver":<String_value>,
    "domain":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "sothreshold":<Double_value>,
    "reuse":<String_value>,
    "state":<String_value>,
    "downstateflush":<String_value>,
    "backupvserver":<String_value>,
    "disableprimaryondown":<String_value>,
    "l2conn":<String_value>,
    "backendssl":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "comment":<String_value>,
    "srcipexpr":<String_value>,
    "originusip":<String_value>,
    "useportrange":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"crvserver":{
    "name":<String_value>,
    "ipv46":<String_value>,
    "redirect":<String_value>,
    "onpolicymatch":<String_value>,
    "precedence":<String_value>,
    "arp":<String_value>,
    "via":<String_value>,
    "cachevserver":<String_value>,
    "dnsvservername":<String_value>,
    "destinationvserver":<String_value>,
    "domain":<String_value>,
    "reuse":<String_value>,
    "backupvserver":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "disableprimaryondown":<String_value>,
    "redirecturl":<String_value>,
    "clttimeout":<Double_value>,
    "downstateflush":<String_value>,
    "l2conn":<String_value>,
    "backendssl":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "netprofile":<String_value>,
    "comment":<String_value>,
    "srcipexpr":<String_value>,
    "originusip":<String_value>,
    "useportrange":<String_value>,
    "appflowlog":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"crvserver":{
    "name":<String_value>,
    "cachevserver":true,
    "dnsvservername":true,
    "destinationvserver":true,
    "domain":true,
    "backupvserver":true,
    "clttimeout":true,
    "redirecturl":true,
    "l2conn":true,
    "backendssl":true,
    "originusip":true,
    "useportrange":true,
    "srcipexpr":true,
    "tcpprofilename":true,
    "httpprofilename":true,
    "appflowlog":true,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "netprofile":true,
    "icmpvsrresponse":true,
    "redirect":true,
    "onpolicymatch":true,
    "precedence":true,
    "arp":true,
    "via":true,
    "reuse":true,
    "disableprimaryondown":true,
    "downstateflush":true,
    "listenpolicy":true,
    "listenpriority":true,
    "comment":true,
    "rhistate":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"crvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver?**action=disable**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"crvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"crvserver":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/crvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of crvserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/crvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver": [ {
    "name":<String_value>,
    "ip":<String_value>,
    "td":<Double_value>,
    "ipv46":<String_value>,
    "value":<String_value>,
    "port":<Integer_value>,
    "range":<Double_value>,
    "servicetype":<String_value>,
    "ngname":<String_value>,
    "type":<String_value>,
    "curstate":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "status":<Integer_value>,
    "sc":<String_value>,
    "cachetype":<String_value>,
    "redirect":<String_value>,
    "onpolicymatch":<String_value>,
    "precedence":<String_value>,
    "redirecturl":<String_value>,
    "authentication":<String_value>,
    "homepage":<String_value>,
    "dnsvservername":<String_value>,
    "domain":<String_value>,
    "rule":<String_value>,
    "policyname":<String_value>,
    "pipolicyhits":<Double_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "cachevserver":<String_value>,
    "targetvserver":<String_value>,
    "backupvserver":<String_value>,
    "priority":<Double_value>,
    "clttimeout":<Double_value>,
    "somethod":<String_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "sothreshold":<Double_value>,
    "reuse":<String_value>,
    "arp":<String_value>,
    "destinationvserver":<String_value>,
    "via":<String_value>,
    "downstateflush":<String_value>,
    "disableprimaryondown":<String_value>,
    "l2conn":<String_value>,
    "backendssl":<String_value>,
    "comment":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "srcipexpr":<String_value>,
    "originusip":<String_value>,
    "useportrange":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "lbvserver":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>
}]}
```


get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/crvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/crvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver": [ {
      "name":<String_value>,
      "ip":<String_value>,
      "td":<Double_value>,
      "ipv46":<String_value>,
      "value":<String_value>,
      "port":<Integer_value>,
      "range":<Double_value>,
      "servicetype":<String_value>,
      "ngname":<String_value>,
      "type":<String_value>,
      "curstate":<String_value>,
      "status":<Integer_value>,
      "sc":<String_value>,
      "cachetype":<String_value>,
      "redirect":<String_value>,
      "onpolicymatch":<String_value>,
      "precedence":<String_value>,
      "redirecturl":<String_value>,
      "authentication":<String_value>,
      "homepage":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "dnsvservername":<String_value>,
    "domain":<String_value>,
    "rule":<String_value>,
    "policyname":<String_value>,
    "pipolicyhits":<Double_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "cachevserver":<String_value>,
    "targetvserver":<String_value>,
    "backupvserver":<String_value>,
    "priority":<Double_value>,
    "clttimeout":<Double_value>,
    "somethod":<String_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "sothreshold":<Double_value>,
    "reuse":<String_value>,
    "arp":<String_value>,
    "destinationvserver":<String_value>,
    "via":<String_value>,
    "downstateflush":<String_value>,
    "disableprimaryondown":<String_value>,
    "l2conn":<String_value>,
    "backendssl":<String_value>,
    "comment":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "srcipexpr":<String_value>,
    "originusip":<String_value>,
    "useportrange":<String_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "lbvserver":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "crvserver": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_appflowpolicy_binding

May 20, 2016

Binding object showing the appflowpolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke flag.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_appflowpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_appflowpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_appflowpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_appflowpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_appflowpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_appflowpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"crvserver_appflowpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_appfwpolicy_binding

May 20, 2016

Binding object showing the appfwpolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke flag.
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_appfwpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_appfwpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_appfwpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_appfwpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_appfwpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "sc":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_appfwpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"crvserver_appfwpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_binding

May 20, 2016

Binding object which returns the resources bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of a cache redirection virtual server about which to display detailed information. Minimum length = 1
crvserver_cmppolicy_binding	<crvserver_cmppolicy_binding[]>	Read-only	cmppolicy that can be bound to crvserver.
crvserver_lbvserver_binding	<crvserver_lbvserver_binding[]>	Read-only	lbvserver that can be bound to crvserver.
crvserver_rewritepolicy_binding	<crvserver_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to crvserver.
crvserver_cspolicy_binding	<crvserver_cspolicy_binding[]>	Read-only	cspolicy that can be bound to crvserver.
crvserver_appfwpolicy_binding	<crvserver_appfwpolicy_binding[]>	Read-only	appfwpolicy that can be bound to crvserver.
crvserver_crpolicy_binding	<crvserver_crpolicy_binding[]>	Read-only	crpolicy that can be bound to crvserver.
crvserver_spilloverpolicy_binding	<crvserver_spilloverpolicy_binding[]>	Read-only	spilloverpolicy that can be bound to crvserver.
crvserver_filterpolicy_binding	<crvserver_filterpolicy_binding[]>	Read-only	filterpolicy that can be bound to crvserver.
crvserver_icapolicy_binding	<crvserver_icapolicy_binding[]>	Read-only	icapolicy that can be bound to crvserver.
crvserver_appflowpolicy_binding	<crvserver_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to crvserver.
crvserver_responderpolicy_binding	<crvserver_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to crvserver.
crvserver_policymap_binding	<crvserver_policymap_binding[]>	Read-only	policymap that can be bound to crvserver.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
crvserver_feopolicy_binding	<crvserver_feopolicy_binding[]>	Read-only	feopolicy that can be bound to crvserver.
crvserver_cachepolicy_binding	<crvserver_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to crvserver.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "crvserver_binding": [ {
    "name":<String_value>,
    "crvserver_spilloverpolicy_binding":<crvserver_spilloverpolicy_binding[]_value>,
    "crvserver_policymap_binding":<crvserver_policymap_binding[]_value>,
    "crvserver_cachepolicy_binding":<crvserver_cachepolicy_binding[]_value>,
    "crvserver_lbvserver_binding":<crvserver_lbvserver_binding[]_value>,
    "crvserver_appfwpolicy_binding":<crvserver_appfwpolicy_binding[]_value>,
    "crvserver_filterpolicy_binding":<crvserver_filterpolicy_binding[]_value>,
    "crvserver_cmppolicy_binding":<crvserver_cmppolicy_binding[]_value>,
    "crvserver_appflowpolicy_binding":<crvserver_appflowpolicy_binding[]_value>,
    "crvserver_icapolicy_binding":<crvserver_icapolicy_binding[]_value>,
    "crvserver_responderpolicy_binding":<crvserver_responderpolicy_binding[]_value>,
    "crvserver_feopolicy_binding":<crvserver_feopolicy_binding[]_value>,
    "crvserver_cspolicy_binding":<crvserver_cspolicy_binding[]_value>,
    "crvserver_crpolicy_binding":<crvserver_crpolicy_binding[]_value>,
    "crvserver_rewritepolicy_binding":<crvserver_rewritepolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_cachepolicy_binding

May 20, 2016

Binding object showing the cachepolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke flag.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_cachepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_cachepolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_cachepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_cachepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_cachepolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_cachepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"crvserver_cachepolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_cmppolicy_binding

May 20, 2016

Binding object showing the cmppolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke flag.
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
inherited	<String>	Read-only	On State describes that policy bound is inherited from global binding. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_cmppolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_cmppolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_cmppolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_cmppolicy_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_cmppolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "sc":<String_value>,
    "inherited":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_cmppolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{"crvserver_cmppolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_crpolicy_binding

May 20, 2016

Binding object showing the crpolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke flag.
hits	<Double>	Read-only	Number of hits.
pipolicyhits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_crpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_crpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_crpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_crpolicy_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_crpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "hits":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_crpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{"crvserver_crpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_cspolicy_binding

May 20, 2016

Binding object showing the cspolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	The CSW target server names.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke flag.
hits	<Double>	Read-only	Number of hits.
pipolicyhits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_cspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_cspolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotoprioryityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_cspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_cspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_cspolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "hits":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_cspolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"crvserver_cspolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_feopolicy_binding

May 20, 2016

Binding object showing the feopolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label to be invoked.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_feopolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_feopolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_feopolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_feopolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_feopolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_feopolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"crvserver_feopolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_filterpolicy_binding

May 20, 2016

Binding object showing the filterpolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT ~V Evaluate the policy with the next higher priority number. * END ~V End policy evaluation. * USE_INVOCATION_RESULT ~V Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the eval* USE_INVOCATION_RESULT ~V Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the eval uation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy~Rs priority. * The expression evaluates to a priority number that is between the current policy~Rs priority number (say, 30) and the highest priority number (say, 100), b ut does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number incr ements by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
targetvserver	<String>	Read-write	policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label to be invoked.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
inherited	<String>	Read-only	On State describes that policy bound is inherited from global binding. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_filterpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_filterpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_filterpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_filterpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_filterpolicy_binding": [ {
    "priority":<Double_value>,
    "gotoprioryexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "inherited":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_filterpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"crvserver_filterpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_icapolicy_binding

May 20, 2016

Binding object showing the icapolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label to be invoked.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_icapolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_icapolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_icapolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_icapolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_icapolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_icapolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"crvserver_icapolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
lbvserver	<String>	Read-write	The Default target server name. Minimum length = 1
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_lbvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_lbvserver_binding":{
    "name":<String_value>,
    "lbvserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_lbvserver_binding": [ {
    "lbvserver":<String_value>,
    "name":<String_value>,
    "hits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"crvserver_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_policymap_binding

Jul 07, 2017

Binding object showing the policymap that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	An unsigned integer that determines the priority of the policy relative to other policies bound to this cache redirection virtual server. The lower the value, higher the priority. Note: This option is available only when binding content switching, filtering, and compression policies to a cache redirection virtual server.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT ~V Evaluate the policy with the next higher priority number. * END ~V End policy evaluation. * USE_INVOCATION_RESULT ~V Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the eval* USE_INVOCATION_RESULT ~V Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the eval uation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy~Rs priority. * The expression evaluates to a priority number that is between the current policy~Rs priority number (say, 30) and the highest priority number (say, 100), b ut does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number incr ements by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
			For a rewrite policy, the bind point to which to bind the policy. Note: This

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	The CSW target server names.
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label to be invoked.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_policymap_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_policymap_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_policymap_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_policymap_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_policymap_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_policymap_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_policymap_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_policymap_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_policymap_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_policymap_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"crvserver_policymap_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_responderpolicy_binding

May 20, 2016

Binding object showing the responderpolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke flag.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_responderpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_responderpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_responderpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_responderpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_responderpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_responderpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"crvserver_responderpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_rewritepolicy_binding

Jul 07, 2017

Binding object showing the rewritepolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke flag.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_rewritepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_rewritepolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_rewritepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_rewritepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_rewritepolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_rewritepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"crvserver_rewritepolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver_spilloverpolicy_binding

May 20, 2016

Binding object showing the spilloverpolicy that can be bound to crvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority for the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Policies bound to this vserver.
name	<String>	Read-write	Name of the cache redirection virtual server to which to bind the cache redirection policy. Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST
targetvserver	<String>	Read-write	Name of the virtual server to which content is forwarded. Applicable only if the policy is a map policy and the cache redirection virtual server is of type REVERSE. Minimum length = 1
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label to be invoked.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/crvserver_spilloverpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"crvserver_spilloverpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of crvserver_spilloverpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the crvserver_spilloverpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver_spilloverpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/crvserver_spilloverpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"crvserver_spilloverpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Content Switching

May 20, 2016

Content Switching configuration. Content Switching feature that enables you to direct traffic to servers on the basis of content.

Name	Description
csaction	Configuration for Content Switching action
csparameter	Configuration for CS parameter
cspolicy	Configuration for content-switching policy
cspolicy_binding	Binding object showing the resources that can be bound to cspolicy
cspolicy_crvserver_binding	Binding object showing the crvserver that can be bound to cspolicy
cspolicy_cspolicylabel_binding	Binding object showing the cspolicylabel that can be bound to cspolicy
cspolicy_csvserver_binding	Binding object showing the csvserver that can be bound to cspolicy
cspolicylabel	Configuration for CS policy label
cspolicylabel_binding	Binding object showing the resources that can be bound to cspolicylabel
cspolicylabel_cspolicy_binding	Binding object showing the cspolicy that can be bound to cspolicylabel
csvserver	Configuration for CS virtual server
csvserver_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to csvserver
csvserver_appfwpolicy_binding	Binding object showing the appfwpolicy that can be bound to csvserver
csvserver_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to csvserver
csvserver_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to csvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
csvserver_authorizationpolicy_binding	Binding object showing the authorizationpolicy that can be bound to csvserver
csvserver_binding	Binding object showing the resources that can be bound to csvserver
csvserver_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to csvserver
csvserver_cmppolicy_binding	Binding object showing the cmppolicy that can be bound to csvserver
csvserver_cspolicy_binding	Binding object showing the cspolicy that can be bound to csvserver
csvserver_domain_binding	Binding object showing the domain that can be bound to csvserver
csvserver_feopolicy_binding	Binding object showing the feopolicy that can be bound to csvserver
csvserver_filterpolicy_binding	Binding object showing the filterpolicy that can be bound to csvserver
csvserver_gslbvserver_binding	Binding object showing the gslbvserver that can be bound to csvserver
csvserver_lbvserver_binding	Binding object showing the lbvserver that can be bound to csvserver
csvserver_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to csvserver
csvserver_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to csvserver
csvserver_spilloverpolicy_binding	Binding object showing the spilloverpolicy that can be bound to csvserver
csvserver_tmtrafficpolicy_binding	Binding object showing the tmtrafficpolicy that can be bound to csvserver
csvserver_transformpolicy_binding	Binding object showing the transformpolicy that can be bound to csvserver
csvserver_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to csvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csaction

May 20, 2016

Configuration for Content Switching action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the content switching action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the content switching action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my action? or ?my action?).
targetlbvserver	<String>	Read-write	Name of the load balancing virtual server to which the content is switched.
targetvserver	<String>	Read-write	Name of the VPN virtual server to which the content is switched.
targetvserverexpr	<String>	Read-write	Information about this content switching action.
comment	<String>	Read-write	Comments associated with this cs action.
newname	<String>	Read-write	New name for the content switching action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, my name or my name). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<Double>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
builtin	<String[]>	Read-only	. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/csaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csaction":{
    "name":<String_value>,
    "targetlbvserver":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "targetvserver":<String_value>,
    "targetvserverexpr":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/csaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/csaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csaction":{
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "targetvserver":<String_value>,
    "targetvserverexpr":<String_value>,
    "comment":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/csaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csaction":{
    "name":<String_value>,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/csaction?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csaction":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/csaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/csaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/csaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of csaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/csaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/csaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "csaction": [ {
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "targetvserver":<String_value>,
    "targetvserverexpr":<String_value>,
    "hits":<Double_value>,
    "referencecount":<Double_value>,
    "undefhits":<Double_value>,
    "builtin":<String[]_value>,
    "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/csaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/csaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "csaction": [ {
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "targetvserver":<String_value>,
    "targetvserverexpr":<String_value>,
    "hits":<Double_value>,
    "referencecount":<Double_value>,
    "undefhits":<Double_value>,
    "builtin":<String[]_value>,
    "comment":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csparameter

May 20, 2016

Configuration for CS parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
stateupdate	<String>	Read-write	Specifies whether the virtual server checks the attached load balancing server for state information. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/csparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csparameter":{
    "stateupdate":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/csparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csparameter":{
    "stateupdate":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/csparameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csparameter": [ {
      "stateupdate":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cspolicy

May 20, 2016

Configuration for content-switching policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name for the content switching policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Cannot be changed after a policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, my policy or my policy). Minimum length = 1
url	<String>	Read-write	URL string that is matched with the URL of a request. Can contain a wildcard character. Specify the string value in the following format: [[prefix] [*]] [.suffix]. Minimum length = 1 Maximum length = 208
rule	<String>	Read-write	Expression, or name of a named expression, against which traffic is evaluated. Written in the classic or default syntax. Note: Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
domain	<String>	Read-write	The domain name. The string value can range to 63 characters. Minimum length = 1
action	<String>	Read-write	Content switching action that names the target load balancing virtual server to which the traffic is switched.
logaction	<String>	Read-write	The log action associated with the content switching policy.
			The new name of the content switching policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
Name	<Double>	Read-only	Server type.
hits	<Double>	Read-only	Total number of hits.
bindhits	<Double>	Read-only	Total number of hits.
labelname	<String>	Read-only	Name of the label invoked.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
priority	<Double>	Read-only	priority of bound policy.
activepolicy	<Boolean>	Read-only	Indicates whether policy is bound or not.
cspolicytype	<String>	Read-only	Indicates whether policy is PI or not.(used only during display). Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cspolicy":{
    "policyname":<String_value>,
    "url":<String_value>,
    "rule":<String_value>,
    "domain":<String_value>,
    "action":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy/policyname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cspolicy":{
    "policyname":<String_value>,
    "url":<String_value>,
    "rule":<String_value>,
    "domain":<String_value>,
    "action":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cspolicy":{
    "policyname":<String_value>,
    "logaction":true,
    "url":true,
    "rule":true,
    "domain":true,
    "action":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cspolicy":{
    "policyname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cspolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cspolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cspolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/cspolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/cspolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cspolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cspolicy": [ {
      "policyname":<String_value>,
      "url":<String_value>,
      "rule":<String_value>,
      "domain":<String_value>,
      "action":<String_value>,
      "vstype":<Double_value>,
      "hits":<Double_value>,
      "bindhits":<Double_value>,
      "labelname":<String_value>,
      "labeltype":<String_value>,
      "target":<String_value>,
      "priority":<Double_value>,
      "activepolicy":<Boolean_value>,
      "cspolicytype":<String_value>,
      "logaction":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy/policyname_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/cspolicy/policyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/cspolicy/policyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cspolicy": [ {
      "policyname":<String_value>,
      "url":<String_value>,
      "rule":<String_value>,
      "domain":<String_value>,
      "action":<String_value>,
      "vstype":<Double_value>,
      "hits":<Double_value>,
      "bindhits":<Double_value>,
      "labelname":<String_value>,
      "labeltype":<String_value>,
      "target":<String_value>,
      "priority":<Double_value>,
      "activepolicy":<Boolean_value>,
      "cspolicytype":<String_value>,
      "logaction":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "cspolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cspolicylabel

May 20, 2016

Configuration for CS policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name for the policy label. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. The label name must be unique within the list of policy labels for content switching. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, \?my label\? or \?my policylabel\?).
cspolicylabeltype	<String>	Read-write	Protocol supported by the policy label. All policies bound to the policy label must either match the specified protocol or be a subtype of that protocol. Available settings function as follows: * HTTP - Supports policies that process HTTP traffic. Used to access unencrypted Web sites. (The default.) * SSL - Supports policies that process HTTPS/SSL encrypted traffic. Used to access encrypted Web sites. * TCP - Supports policies that process any type of TCP traffic, including HTTP. * SSL_TCP - Supports policies that process SSL-encrypted TCP traffic, including SSL. * UDP - Supports policies that process any type of UDP-based traffic, including DNS. * DNS - Supports policies that process DNS traffic. * ANY - Supports all types of policies except HTTP, SSL, and TCP. * SIP_UDP - Supports policies that process UDP based Session Initiation Protocol (SIP) traffic. SIP initiates, manages, and terminates multimedia communications sessions, and has emerged as the standard for Internet telephony (VoIP). * RTSP - Supports policies that process Real Time Streaming Protocol (RTSP) traffic. RTSP provides delivery of multimedia and other streaming data, such as audio, video, and other types of streamed media. * RADIUS - Supports policies that process Remote Authentication Dial In User Service (RADIUS) traffic. RADIUS supports combined authentication, authorization, and auditing services for network management. * MYSQL - Supports policies that process MYSQL traffic. * MSSQL - Supports policies that process Microsoft SQL traffic. Possible values = HTTP, TCP, RTSP, SSL, SSL_TCP, UDP, DNS, SIP_UDP, SIP_TCP, ANY, RADIUS, RDP, MYSQL, MSSQL, ORACLE, DIAMETER, SSL_DIAMETER, FTP, DNS_TCP, SMPP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
numpol	<Double>	Read-only	number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
policyname	<String>	Read-only	Name of the content switching policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
targetvserver	<String>	Read-only	Name of the virtual server to which to forward requests that match the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = policylabel
invoke_labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cspolicylabel":{
    "labelname":<String_value>,
    "cspolicylabeltype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"cspolicylabel":{
    "labelname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of cspolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cspolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cspolicylabel": [ {
      "labelname":<String_value>,
      "cspolicylabeltype":<String_value>,
      "numpol":<Double_value>,
      "hits":<Double_value>,
      "policyname":<String_value>,
      "priority":<Double_value>,
      "targetvserver":<String_value>,
      "gotopriorityexpression":<String_value>,
      "labeltype":<String_value>,
      "invoke_labelname":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel/labelname_value<String>?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cspolicylabel": [ {
    "labelname":<String_value>,
    "cspolicylabeltype":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "targetvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "cspolicylabel": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cspolicylabel_binding

May 20, 2016

Binding object which returns the resources bound to cspolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the content switching policy label to display.
cspolicylabel_cspolicy_binding	<cspolicylabel_cspolicy_binding[]>	Read-only	cspolicy that can be bound to cspolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "cspolicylabel_binding": [ {
      "labelname":<String_value>,
      "cspolicylabel_cspolicy_binding":<cspolicylabel_cspolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cspolicylabel_cspolicy_binding

May 20, 2016

Binding object showing the cspolicy that can be bound to cspolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the content switching policy.
targetvserver	<String>	Read-write	Name of the virtual server to which to forward requests that match the policy.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = policylabel
labelname	<String>	Read-write	Name of the policy label to which to bind a content switching policy.
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/cspolicylabel_cspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"cspolicylabel_cspolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "targetvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cspolicylabel_cspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cspolicylabel_cspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cspolicylabel_cspolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "targetvserver":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicylabel_cspolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cspolicylabel_cspolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cspolicy_binding

May 20, 2016

Binding object which returns the resources bound to cspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the content switching policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
cspolicy_cspolicylabel_binding	<cspolicy_cspolicylabel_binding[]>	Read-only	cspolicylabel that can be bound to cspolicy.
cspolicy_csvserver_binding	<cspolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to cspolicy.
cspolicy_crvserver_binding	<cspolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to cspolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy_binding/policyname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "cspolicy_binding": [ {
      "policyname":<String_value>,
      "cspolicy_cspolicylabel_binding":<cspolicy_cspolicylabel_binding[]_value>,
      "cspolicy_crvserver_binding":<cspolicy_crvserver_binding[]_value>,
      "cspolicy_csvserver_binding":<cspolicy_csvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cspolicy_crvserver_binding

Jul 07, 2017

Binding object showing the crvserver that can be bound to cspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the content switching policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
domain	<String>	Read-write	The domain name. The string value can range to 63 characters. Minimum length = 1
priority	<Double>	Read-only	priority of bound policy.
hits	<Double>	Read-only	Total number of hits.
url	<String>	Read-only	URL string that is matched with the URL of a request. Can contain a wildcard character. Specify the string value in the following format: [[prefix] [*]] [.suffix]. Minimum length = 1 Maximum length = 208
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy_crvserver_binding/policyname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cspolicy_crvserver_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cspolicy_crvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cspolicy_crvserver_binding/policyname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cspolicy_crvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cspolicy_crvserver_binding": [ {
    "policyname":<String_value>,
    "domain":<String_value>,
    "priority":<Double_value>,
    "hits":<Double_value>,
    "url":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy_crvserver_binding/policyname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cspolicy_crvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cspolicy_cspolicylabel_binding

Jul 07, 2017

Binding object showing the cspolicylabel that can be bound to cspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the content switching policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
domain	<String>	Read-write	The domain name. The string value can range to 63 characters. Minimum length = 1
priority	<Double>	Read-only	priority of bound policy.
hits	<Double>	Read-only	Total number of hits.
url	<String>	Read-only	URL string that is matched with the URL of a request. Can contain a wildcard character. Specify the string value in the following format: [[prefix] [*]] [.suffix]. Minimum length = 1 Maximum length = 208
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy_cspolicylabel_binding/policyname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cspolicy_cspolicylabel_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cspolicy_cspolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cspolicy_cspolicylabel_binding/policyname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cspolicy_cspolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cspolicy_cspolicylabel_binding": [ {
      "policyname":<String_value>,
      "domain":<String_value>,
      "priority":<Double_value>,
      "hits":<Double_value>,
      "url":<String_value>,
      "labeltype":<String_value>,
      "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy_cspolicylabel_binding/policyname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"cspolicy_cspolicylabel_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cspolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to cspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the content switching policy to display. If this parameter is omitted, details of all the policies are displayed. Minimum length = 1
domain	<String>	Read-write	The domain name. The string value can range to 63 characters. Minimum length = 1
priority	<Double>	Read-only	priority of bound policy.
hits	<Double>	Read-only	Total number of hits.
url	<String>	Read-only	URL string that is matched with the URL of a request. Can contain a wildcard character. Specify the string value in the following format: [[prefix] [*]] [.suffix]. Minimum length = 1 Maximum length = 208
pipolicyhits	<Double>	Read-only	bind hits for PI CS Policy.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
pihits	<Double>	Read-only	Total number of hits.
action	<String>	Read-only	Content switching action that names the target load balancing virtual server to which the traffic is switched.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy_csvserver_binding/policyname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/cspolicy_csvserver_binding/policyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of cspolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/cspolicy_csvserver_binding/policyname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the cspolicy_csvserver_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cspolicy_csvserver_binding": [ {
      "policyname":<String_value>,
      "domain":<String_value>,
      "priority":<Double_value>,
      "hits":<Double_value>,
      "url":<String_value>,
      "pipolicyhits":<Double_value>,
      "labeltype":<String_value>,
      "labelname":<String_value>,
      "pihits":<Double_value>,
      "action":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/cspolicy_csvserver_binding/policyname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{"cspolicy_csvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver

Jul 07, 2017

Configuration for CS virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the content switching virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Cannot be changed after the CS virtual server is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, my server or my server). Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
servicetype	<String>	Read-write	Protocol used by the virtual server. Possible values = HTTP, SSL, TCP, FTP, RTSP, SSL_TCP, UDP, DNS, SIP_UDP, SIP_TCP, SIP_SSL, ANY, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, DNS_TCP, ORACLE, SMPP
ipv46	<String>	Read-write	IP address of the content switching virtual server. Minimum length = 1
targettype	<String>	Read-write	Virtual server target type. Possible values = GSLB
dnsrecordtype	<String>	Read-write	. Default value: NSGSLB_IPV4 Possible values = A, AAAA, CNAME, NAPTR
persistenceid	<Double>	Read-write	. Minimum value = 0 Maximum value = 65535
			IP address pattern, in dotted decimal notation, for identifying packets to be accepted by the virtual server. The IP Mask parameter specifies which part of the destination IP address is matched against the pattern. Mutually exclusive with the IP Address parameter.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ippattern	<String>	Read-write	and the IP mask is 255.255.240.0 (a forward mask), the first 20 bits in the destination IP addresses are matched with the first 20 bits in the pattern. The virtual server accepts requests with IP addresses that range from 198.51.96.1 to 198.51.111.254. You can also use a pattern such as 0.0.2.2 and a mask such as 0.0.255.255 (a reverse mask). If a destination IP address matches more than one IP pattern, the pattern with the longest match is selected, and the associated virtual server processes the request. For example, if the virtual servers, vs1 and vs2, have the same IP pattern, 0.0.100.128, but different IP masks of 0.0.255.255 and 0.0.224.255, a destination IP address of 198.51.100.128 has the longest match with the IP pattern of vs1. If a destination IP address matches two or more virtual servers to the same extent, the request is processed by the virtual server whose port number matches the port number in the request.
ipmask	<String>	Read-write	IP mask, in dotted decimal notation, for the IP Pattern parameter. Can have leading or trailing non-zero octets (for example, 255.255.240.0 or 0.0.255.255). Accordingly, the mask specifies whether the first n bits or the last n bits of the destination IP address in a client request are to be matched with the corresponding bits in the IP pattern. The former is called a forward mask. The latter is called a reverse mask.
range	<Double>	Read-write	Number of consecutive IP addresses, starting with the address specified by the IP Address parameter, to include in a range of addresses assigned to this virtual server. Default value: 1 Minimum value = 1 Maximum value = 254
port	<Integer>	Read-write	Port number for content switching virtual server. Minimum value = 1 Range 1 - 65535
state	<String>	Read-write	Initial state of the load balancing virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
stateupdate	<String>	Read-write	Enable state updates for a specific content switching virtual server. By default, the Content Switching virtual server is always UP, regardless of the state of the Load Balancing virtual servers bound to it. This parameter interacts with the global setting as follows: Global Level | Vserver Level | Result ENABLED ENABLED ENABLED ENABLED DISABLED ENABLED DISABLED ENABLED ENABLED DISABLED DISABLED DISABLED If you want to enable state updates for only some content switching virtual servers, be sure to disable the state update parameter. Default value: DISABLED Possible values = ENABLED, DISABLED
cacheable	<String>	Read-write	Use this option to specify whether a virtual server, used for load balancing or content switching, routes requests to the cache redirection virtual server before sending it to the configured servers. Default value: NO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			URL to which traffic is redirected if the virtual server becomes unavailable. The service type of the virtual server should be either HTTP or SSL.
redirecturl	<String>	Read-write	Caution: Make sure that the domain in the URL does not match the domain specified for a content switching policy. If it does, requests are continuously redirected to the unavailable virtual server. Minimum length = 1
clttimeout	<Double>	Read-write	Idle time, in seconds, after which the client connection is terminated. The default values are: 180 seconds for HTTP/SSL-based services. 9000 seconds for other TCP-based services. 120 seconds for DNS-based services. 120 seconds for other UDP-based services. Minimum value = 0 Maximum value = 31536000
precedence	<String>	Read-write	Type of precedence to use for both RULE-based and URL-based policies on the content switching virtual server. With the default (RULE) setting, incoming requests are evaluated against the rule-based content switching policies. If none of the rules match, the URL in the request is evaluated against the URL-based content switching policies. Default value: RULE Possible values = RULE, URL
casesensitive	<String>	Read-write	Consider case in URLs (for policies that use URLs instead of RULES). For example, with the ON setting, the URLs /a/1.html and /A/1.HTML are treated differently and can have different targets (set by content switching policies). With the OFF setting, /a/1.html and /A/1.HTML are switched to the same target. Default value: ON Possible values = ON, OFF
somethod	<String>	Read-write	Type of spillover used to divert traffic to the backup virtual server when the primary virtual server reaches the spillover threshold. Connection spillover is based on the number of connections. Bandwidth spillover is based on the total Kbps of incoming and outgoing traffic. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sopersistence	<String>	Read-write	Maintain source-IP based persistence on primary and backup virtual servers. Default value: DISABLED Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-write	Time-out value, in minutes, for spillover persistence. Default value: 2 Minimum value = 2 Maximum value = 1440
sothreshold	<Double>	Read-write	Depending on the spillover method, the maximum number of connections or the maximum total bandwidth (Kbps) that a virtual server can handle before spillover occurs. Minimum value = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
nobackupaction	<String>	Read-write	Action to be performed if spillover is to take effect, but no backup chain to spillover is usable or exists. Possible values = DROP, ACCEPT, REDIRECT
redirectportrewrite	<String>	Read-write	State of port rewrite while performing HTTP redirect. Default value: DISABLED Possible values = ENABLED, DISABLED
downstateflush	<String>	Read-write	Flush all active transactions associated with a virtual server whose state transitions from UP to DOWN. Do not enable this option for applications that must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED
backupvserver	<String>	Read-write	Name of the backup virtual server that you are configuring. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the backup virtual server is created. You can assign a different backup virtual server or rename the existing virtual server. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks. Minimum length = 1
disableprimaryondown	<String>	Read-write	Continue forwarding the traffic to backup virtual server even after the primary server comes UP from the DOWN state. Default value: DISABLED Possible values = ENABLED, DISABLED
insertvserveripport	<String>	Read-write	Insert the virtual server's VIP address and port number in the request header. Available values function as follows: VIPADDR - Header contains the vserver's IP address and port number without any translation. OFF - The virtual IP and port header insertion option is disabled. V6TOV4MAPPING - Header contains the mapped IPv4 address corresponding to the IPv6 address of the vserver and the port number. An IPv6 address can be mapped to a user-specified IPv4 address using the set ns ip6 command. Possible values = OFF, VIPADDR, V6TOV4MAPPING
vipheader	<String>	Read-write	Name of virtual server IP and port header, for use with the VServer IP Port Insertion parameter. Minimum length = 1
rtspnat	<String>	Read-write	Enable network address translation (NAT) for real-time streaming protocol (RTSP) connections. Default value: OFF Possible values = ON, OFF
authenticationhost	<String>	Read-write	FQDN of the authentication virtual server. The service type of the virtual server should be either HTTP or SSL. Minimum length = 3 Maximum length = 252

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
authentication	<String>	Read-write	Authenticate users who request a connection to the content switching virtual server. Default value: OFF Possible values = ON, OFF
listenpolicy	<String>	Read-write	String specifying the listen policy for the content switching virtual server. Can be either the name of an existing expression or an in-line expression. Default value: "NONE"
listenpriority	<Double>	Read-write	Integer specifying the priority of the listen policy. A higher number specifies a lower priority. If a request matches the listen policies of more than one virtual server the virtual server whose listen policy has the highest priority (the lowest priority number) accepts the request. Default value: 101 Minimum value = 0 Maximum value = 100
authn401	<String>	Read-write	Enable HTTP 401-response based authentication. Default value: OFF Possible values = ON, OFF
authnvsname	<String>	Read-write	Name of authentication virtual server that authenticates the incoming user requests to this content switching virtual server. . Minimum length = 1 Maximum length = 252
push	<String>	Read-write	Process traffic with the push virtual server that is bound to this content switching virtual server (specified by the Push VServer parameter). The service type of the push virtual server should be either HTTP or SSL. Default value: DISABLED Possible values = ENABLED, DISABLED
pushvserver	<String>	Read-write	Name of the load balancing virtual server, of type PUSH or SSL_PUSH, to which the server pushes updates received on the client-facing load balancing virtual server. Minimum length = 1
pushlabel	<String>	Read-write	Expression for extracting the label from the response received from server. This string can be either an existing rule name or an inline expression. The service type of the virtual server should be either HTTP or SSL. Default value: "none"
pushmulticlients	<String>	Read-write	Allow multiple Web 2.0 connections from the same client to connect to the virtual server and expect updates. Default value: NO Possible values = YES, NO
tcpprofilename	<String>	Read-write	Name of the TCP profile containing TCP configuration settings for the virtual server. Minimum length = 1 Maximum length = 127
			Name of the HTTP profile containing HTTP configuration settings for the virtual server. The service type of the virtual server should be either HTTP or

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1 Maximum length = 127
dbprofilename	<String>	Read-write	Name of the DB profile. Minimum length = 1 Maximum length = 127
oracleserverversion	<String>	Read-write	Oracle server version. Default value: 10G Possible values = 10G, 11G
comment	<String>	Read-write	Information about this virtual server.
mssqlserverversion	<String>	Read-write	The version of the MSSQL server. Default value: 2008R2 Possible values = 70, 2000, 2000SP1, 2005, 2008, 2008R2, 2012, 2014
l2conn	<String>	Read-write	Use L2 Parameters to identify a connection. Possible values = ON, OFF
mysqlprotocolversion	<Double>	Read-write	The protocol version returned by the mysql vserver. Default value: 10
mysqlserverversion	<String>	Read-write	The server version string returned by the mysql vserver. Minimum length = 1 Maximum length = 31
mysqlcharacterset	<Double>	Read-write	The character set returned by the mysql vserver. Default value: 8
mysqlservercapabilities	<Double>	Read-write	The server capabilities returned by the mysql vserver. Default value: 41613
appflowlog	<String>	Read-write	Enable logging appflow flow information. Default value: ENABLED Possible values = ENABLED, DISABLED
netprofile	<String>	Read-write	The name of the network profile. Minimum length = 1 Maximum length = 127
icmpvsrresponse	<String>	Read-write	Can be active or passive. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
rhistate	<String>	Read-write	A host route is injected according to the setting on the virtual servers * If set to PASSIVE on all the virtual servers that share the IP address, the appliance always injects the hostroute. * If set to ACTIVE on all the virtual servers that share the IP address, the appliance injects even if one virtual server is UP. * If set to ACTIVE on some virtual servers and PASSIVE on the others, the appliance, injects even if one virtual server set to ACTIVE is UP. Default value: PASSIVE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
authnprofile	<String>	Read-write	Name of the authentication profile to be used when authentication is turned on.
dnsprofilename	<String>	Read-write	Name of the DNS profile to be associated with the VServer. DNS profile properties will applied to the transactions processed by a VServer. This parameter is valid only for DNS and DNS-TCP VServers. Minimum length = 1 Maximum length = 127
domainname	<String>	Read-write	Domain name for which to change the time to live (TTL) and/or backup service IP address. Minimum length = 1
ttl	<Double>	Read-write	. Minimum value = 1
backupip	<String>	Read-write	. Minimum length = 1
cookiedomain	<String>	Read-write	. Minimum length = 1
cookietimeout	<Double>	Read-write	. Minimum value = 0 Maximum value = 1440
sitedomainttl	<Double>	Read-write	. Minimum value = 1
newname	<String>	Read-write	New name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, my name or my name). Minimum length = 1
ip	<String>	Read-only	The IP address of the virtual server.
value	<String>	Read-only	The ssl card status for the transparent ssl cs vserver. Possible values = Certkey not bound, SSL feature disabled
ngname	<String>	Read-only	Nodegroup devno to which this csvserver belongs to.
type	<String>	Read-only	Virtual server type. Possible values = CONTENT, ADDRESS
curstate	<String>	Read-only	The state of the cs vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
status	<Integer>	Read-only	Status.
cachetype	<String>	Read-only	Cache type. Possible values = TRANSPARENT, REVERSE, FORWARD
redirect	<String>	Read-only	Redirect URL string. Possible values = CACHE, POLICY, ORIGIN
homepage	<String>	Read-only	Home page.
dnsvservername	<String>	Read-only	DNS vserver name.
domain	<String>	Read-only	Domain.
policyname	<String>	Read-only	Policies bound to this vserver.
servicename	<String>	Read-only	Service name.
weight	<Double>	Read-only	Weight for this service.
cachevserver	<String>	Read-only	Cache vserver name.
targetvserver	<String>	Read-only	target vserver name.
priority	<Double>	Read-only	Priority for the policy.
url	<String>	Read-only	URL string.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
bindpoint	<String>	Read-only	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
invoke	<Boolean>	Read-only	Invoke flag.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
gt2gb	<String>	Read-only	This argument has no effect. Default value: DISABLED Possible values = ENABLED, DISABLED
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
statechangetimemsec	<Double>	Read-only	Time at which last state change happened. Milliseconds part.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
ruletype	<Double>	Read-only	Rule type.
lbvserver	<String>	Read-only	Name of the default lb vserver bound. Use this param for Default binding only. For Example: bind cs vserver cs1 -lbvserver lb1. Minimum length = 1
targetlbvserver	<String>	Read-only	target vserver name.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csvserver":{
    "name":<String_value>,
    "td":<Double_value>,
    "servicetype":<String_value>,
    "ipv46":<String_value>,
    "targettype":<String_value>,
    "dnsrecordtype":<String_value>,
    "persistenceid":<Double_value>,
    "ippattern":<String_value>,
    "ipmask":<String_value>,
    "range":<Double_value>,
    "port":<Integer_value>,
    "state":<String_value>,
    "stateupdate":<String_value>,
    "cacheable":<String_value>,
    "redirecturl":<String_value>,
    "clttimeout":<Double_value>,
    "precedence":<String_value>,
    "casesensitive":<String_value>,
    "somethod":<String_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "sothreshold":<Double_value>,
    "sobackupaction":<String_value>,
    "redirectportrewrite":<String_value>,
    "downstateflush":<String_value>,
    "backupvserver":<String_value>,
    "disableprimaryondown":<String_value>,
    "insertvserveripport":<String_value>,
    "vipheader":<String_value>,
    "rtspnat":<String_value>,
    "authenticationhost":<String_value>,
    "authentication":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "authn401":<String_value>,
    "authnvsname":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "push":<String_value>,
    "pushvserver":<String_value>,
    "pushlabel":<String_value>,
    "pushmulticlients":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "dbprofilename":<String_value>,
    "oracleserverversion":<String_value>,
    "comment":<String_value>,
    "mssqlserverversion":<String_value>,
    "l2conn":<String_value>,
    "mysqlprotocolversion":<Double_value>,
    "mysqlserverversion":<String_value>,
    "mysqlcharacterset":<Double_value>,
    "mysqlservercapabilities":<Double_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "authnprofile":<String_value>,
    "dnsprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver

HTTP Method: PUT

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csvserver":{
    "name":<String_value>,
    "ipv46":<String_value>,
    "ippattern":<String_value>,
    "ipmask":<String_value>,
    "stateupdate":<String_value>,
    "precedence":<String_value>,
    "casesensitive":<String_value>,
    "backupvserver":<String_value>,
    "redirecturl":<String_value>,
    "cacheable":<String_value>,
    "clttimeout":<Double_value>,
    "somethod":<String_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "sothreshold":<Double_value>,
    "sobackupaction":<String_value>,
    "redirectportrewrite":<String_value>,
    "downstateflush":<String_value>,
    "disableprimaryondown":<String_value>,
    "insertvserveripport":<String_value>,
    "vipheader":<String_value>,
    "rtspnat":<String_value>,
    "authenticationhost":<String_value>,
    "authentication":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "authn401":<String_value>,
    "authnvsname":<String_value>,
    "push":<String_value>,
    "pushvserver":<String_value>,
    "pushlabel":<String_value>,
    "pushmulticlients":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "dbprofilename":<String_value>,
    "comment":<String_value>,
    "l2conn":<String_value>,
    "mssqlserverversion":<String_value>,
    "mysqlprotocolversion":<Double_value>,
    "oracleserverversion":<String_value>,
    "mysqlserverversion":<String_value>,
    "mysqlcharacterset":<Double_value>,
    "mysqlservercapabilities":<Double_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "authnprofile":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "dnsprofilename":<String_value>,
    "dnsrecordtype":<String_value>,
    "persistenceid":<Double_value>,
    "domainname":<String_value>,
    "ttl":<Double_value>,
    "backupip":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cookiedomain":<String_value>,
    "cookietimeout":<Double_value>,
    "sitedomainttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csvserver":{
    "name":<String_value>,
    "casesensitive":true,
    "backupvserver":true,
    "clttimeout":true,
    "redirecturl":true,
    "authn401":true,
    "authentication":true,
    "authenticationhost":true,
    "authnvsname":true,
    "pushvserver":true,
    "pushlabel":true,
    "tcpprofilename":true,
    "httpprofilename":true,
    "dbprofilename":true,
    "l2conn":true,
    "mysqlprotocolversion":true,
    "mysqlserverversion":true,
    "mysqlcharacterset":true,
    "mysqlservercapabilities":true,
    "appflowlog":true,
    "netprofile":true,
    "icmpvsrresponse":true,
    "authnprofile":true,
    "sothreshold":true,
    "dnsprofilename":true,
    "stateupdate":true,
    "precedence":true,
    "cacheable":true,
    "somethod":true,
    "sopersistence":true,
    "sopersistencetimeout":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "sobackupaction":true,
        "redirectportrewrite":true,
        "downstateflush":true,
        "disableprimaryondown":true,
        "insertvserveripport":true,
        "vipheader":true,
        "rtspnat":true,
        "listenpolicy":true,
        "listenpriority":true,
        "push":true,
        "pushmulticlients":true,
        "comment":true,
        "mssqlserverversion":true,
        "oracleserverversion":true,
        "rhistate":true,
        "dnsrecordtype":true,
        "persistenceid":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver?**action=disable**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"csvserver":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/csvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of csvserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/csvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver": [ {
    "name":<String_value>,
    "insertvserveripport":<String_value>,
    "vipheader":<String_value>,
    "ip":<String_value>,
    "td":<Double_value>,
    "ipv46":<String_value>,
    "ippattern":<String_value>,
    "ipmask":<String_value>,
    "value":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

"port":<Integer_value>,
"range":<Double_value>,
"servicetype":<String_value>,
"ngname":<String_value>,
"type":<String_value>,
"curstate":<String_value>,
"sc":<String_value>,
"stateupdate":<String_value>,
"status":<Integer_value>,
"cachetype":<String_value>,
"redirect":<String_value>,
"precedence":<String_value>,
"redirecturl":<String_value>,
"authentication":<String_value>,
"authn401":<String_value>,
"authnvsname":<String_value>,
"casesensitive":<String_value>,
"homepage":<String_value>,
"dnsvservername":<String_value>,
"domain":<String_value>,
"policyname":<String_value>,
"servicename":<String_value>,
"weight":<Double_value>,
"cachevserver":<String_value>,
"targetvserver":<String_value>,
"backupvserver":<String_value>,
"priority":<Double_value>,
"clttimeout":<Double_value>,
"listenpolicy":<String_value>,
"listenpriority":<Double_value>,
"somethod":<String_value>,
"sopersistence":<String_value>,
"sopersistencetimeout":<Double_value>,
"sothreshold":<Double_value>,
"sobackupaction":<String_value>,
"cacheable":<String_value>,
"url":<String_value>,
"gotopriorityexpression":<String_value>,
"redirectportrewrite":<String_value>,
"downstateflush":<String_value>,
"disableprimaryondown":<String_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>,
"gt2gb":<String_value>,
"statechangetimesec":<String_value>,
"statechangetimemsec":<Double_value>,
"tickssincelaststatechange":<Double_value>,
"rtspnat":<String_value>,
"authenticationhost":<String_value>,
"push":<String_value>,
"pushvserver":<String_value>,
"pushlabel":<String_value>,
"pushmulticlients":<String_value>,
"tcpprofilename":<String_value>,
"httpprofilename":<String_value>,
"dbprofilename":<String_value>,
"comment":<String_value>,
"oracleserverversion":<String_value>,
"mssqlserverversion":<String_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "l2conn":<String_value>,
    "ruletype":<Double_value>,
    "mysqlprotocolversion":<Double_value>,
    "mysqlserverversion":<String_value>,
    "mysqlcharacterset":<Double_value>,
    "mysqlservercapabilities":<Double_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "lbvserver":<String_value>,
    "targetlbvserver":<String_value>,
    "authnprofile":<String_value>,
    "dnsprofilename":<String_value>,
    "targettype":<String_value>,
    "domainname":<String_value>,
    "ttl":<Double_value>,
    "backupip":<String_value>,
    "cookiedomain":<String_value>,
    "cookietimeout":<Double_value>,
    "sitedomainttl":<Double_value>,
    "dnsrecordtype":<String_value>,
    "persistenceid":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/csvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/csvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "csvserver": [ {
    "name":<String_value>,
    "insertvserveripport":<String_value>,
    "vipheader":<String_value>,
    "ip":<String_value>,
    "td":<Double_value>,
    "ipv46":<String_value>,
    "ippattern":<String_value>,
    "ipmask":<String_value>,
    "value":<String_value>,
    "port":<Integer_value>,
    "range":<Double_value>,
    "servicetype":<String_value>,
    "ngname":<String_value>,
    "type":<String_value>,
    "curstate":<String_value>,
    "sc":<String_value>,
    "stateupdate":<String_value>,
    "status":<Integer_value>,
    "cachetype":<String_value>,
    "redirect":<String_value>,
    "precedence":<String_value>,
    "redirecturl":<String_value>,
    "authentication":<String_value>,
    "authn401":<String_value>,
    "authnvsname":<String_value>,
    "casesensitive":<String_value>,
    "homepage":<String_value>,
    "dnsvservername":<String_value>,
    "domain":<String_value>,
    "policyname":<String_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "cachevserver":<String_value>,
    "targetvserver":<String_value>,
    "backupvserver":<String_value>,
    "priority":<Double_value>,
    "clttimeout":<Double_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "somethod":<String_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "sothreshold":<Double_value>,
    "sobackupaction":<String_value>,
    "cacheable":<String_value>,
    "url":<String_value>,
    "gotopriorityexpression":<String_value>,
    "redirectportrewrite":<String_value>,
    "downstateflush":<String_value>,
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "disableprimaryondown":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "gt2gb":<String_value>,
    "statechangetimesec":<String_value>,
    "statechangetimemsec":<Double_value>,
    "tickssincelaststatechange":<Double_value>,
    "rtspnat":<String_value>,
    "authenticationhost":<String_value>,
    "push":<String_value>,
    "pushvserver":<String_value>,
    "pushlabel":<String_value>,
    "pushmulticlients":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "dbprofilename":<String_value>,
    "comment":<String_value>,
    "oracleserverversion":<String_value>,
    "mssqlserverversion":<String_value>,
    "l2conn":<String_value>,
    "ruletype":<Double_value>,
    "mysqlprotocolversion":<Double_value>,
    "mysqlserverversion":<String_value>,
    "mysqlcharacterset":<Double_value>,
    "mysqlservercapabilities":<Double_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "lbvserver":<String_value>,
    "targetlbvserver":<String_value>,
    "authnprofile":<String_value>,
    "dnsprofilename":<String_value>,
    "targettype":<String_value>,
    "domainname":<String_value>,
    "ttl":<Double_value>,
    "backupip":<String_value>,
    "cookiedomain":<String_value>,
    "cookietimeout":<Double_value>,
    "sitedomainttl":<Double_value>,
    "dnsrecordtype":<String_value>,
    "persistenceid":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "csvserver": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_appflowpolicy_binding

Jul 07, 2017

Binding object showing the appflowpolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_appflowpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_appflowpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_appflowpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_appflowpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_appflowpolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_appflowpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"csvserver_appflowpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_appfwpolicy_binding

Jul 07, 2017

Binding object showing the appfwpolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_appfwpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_appfwpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_appfwpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_appfwpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_appfwpolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "sc":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_appfwpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"csvserver_appfwpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_auditnslogpolicy_binding

May 20, 2016

Binding object showing the auditnslogpolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT ? Evaluate the policy with the next higher priority number. * END ? End policy evaluation. * USE_INVOCATION_RESULT ? Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy?s priority. * The expression evaluates to a priority number that is between the current policy?s priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_auditnslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_auditnslogpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_auditnslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_auditnslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_auditnslogpolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "gotopriorityexpression":<String_value>,
    "targetlbvserver":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "sc":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_auditnslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"csvserver_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_auditsyslogpolicy_binding

May 20, 2016

Binding object showing the auditsyslogpolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT ? Evaluate the policy with the next higher priority number. * END ? End policy evaluation. * USE_INVOCATION_RESULT ? Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy?s priority. * The expression evaluates to a priority number that is between the current policy?s priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_auditsyslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_auditsyslogpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_auditsyslogpolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "gotopriorityexpression":<String_value>,
    "targetlbvserver":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "sc":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_auditsyslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"csvserver_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_authorizationpolicy_binding

May 20, 2016

Binding object showing the authorizationpolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_authorizationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_authorizationpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_authorizationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_authorizationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_authorizationpolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "sc":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_authorizationpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"csvserver_authorizationpolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_binding

Jul 07, 2017

Binding object which returns the resources bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of a content switching virtual server for which to display information, including the policies bound to the virtual server. To display a list of all configured Content Switching virtual servers, do not specify a value for this parameter. Minimum length = 1
csvserver_spilloverpolicy_binding	<csvserver_spilloverpolicy_binding[]>	Read-only	spilloverpolicy that can be bound to csvserver.
csvserver_auditnslogpolicy_binding	<csvserver_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to csvserver.
csvserver_domain_binding	<csvserver_domain_binding[]>	Read-only	domain that can be bound to csvserver.
csvserver_filterpolicy_binding	<csvserver_filterpolicy_binding[]>	Read-only	filterpolicy that can be bound to csvserver.
csvserver_cmppolicy_binding	<csvserver_cmppolicy_binding[]>	Read-only	cmppolicy that can be bound to csvserver.
csvserver_lbvserver_binding	<csvserver_lbvserver_binding[]>	Read-only	lbvserver that can be bound to csvserver.
csvserver_appflowpolicy_binding	<csvserver_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to csvserver.
csvserver_responderpolicy_binding	<csvserver_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to csvserver.
csvserver_transformpolicy_binding	<csvserver_transformpolicy_binding[]>	Read-only	transformpolicy that can be bound to csvserver.
csvserver_vpnvserver_binding	<csvserver_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to csvserver.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			csvserver.
csvserver_authorizationpolicy_binding	<csvserver_authorizationpolicy_binding[]>	Read-only	authorizationpolicy that can be bound to csvserver.
csvserver_rewritepolicy_binding	<csvserver_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to csvserver.
csvserver_cachepolicy_binding	<csvserver_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to csvserver.
csvserver_cspolicy_binding	<csvserver_cspolicy_binding[]>	Read-only	cspolicy that can be bound to csvserver.
csvserver_gslbvserver_binding	<csvserver_gslbvserver_binding[]>	Read-only	gslbvserver that can be bound to csvserver.
csvserver_auditsyslogpolicy_binding	<csvserver_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to csvserver.
csvserver_tmtrafficpolicy_binding	<csvserver_tmtrafficpolicy_binding[]>	Read-only	tmtrafficpolicy that can be bound to csvserver.
csvserver_appfwpolicy_binding	<csvserver_appfwpolicy_binding[]>	Read-only	appfwpolicy that can be bound to csvserver.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "csvserver_binding": [ {
    "name":<String_value>,
    "csvserver_auditnslogpolicy_binding":<csvserver_auditnslogpolicy_binding[]_value>,
    "csvserver_tmtrafficpolicy_binding":<csvserver_tmtrafficpolicy_binding[]_value>,
    "csvserver_lbvserver_binding":<csvserver_lbvserver_binding[]_value>,
    "csvserver_responderpolicy_binding":<csvserver_responderpolicy_binding[]_value>,
    "csvserver_gslbvserver_binding":<csvserver_gslbvserver_binding[]_value>,
    "csvserver_cachepolicy_binding":<csvserver_cachepolicy_binding[]_value>,
    "csvserver_filterpolicy_binding":<csvserver_filterpolicy_binding[]_value>,
    "csvserver_transformpolicy_binding":<csvserver_transformpolicy_binding[]_value>,
    "csvserver_appflowpolicy_binding":<csvserver_appflowpolicy_binding[]_value>,
    "csvserver_appfwpolicy_binding":<csvserver_appfwpolicy_binding[]_value>,
    "csvserver_authorizationpolicy_binding":<csvserver_authorizationpolicy_binding[]_value>,
    "csvserver_auditsyslogpolicy_binding":<csvserver_auditsyslogpolicy_binding[]_value>,
    "csvserver_cmppolicy_binding":<csvserver_cmppolicy_binding[]_value>,
    "csvserver_rewritepolicy_binding":<csvserver_rewritepolicy_binding[]_value>,
    "csvserver_spilloverpolicy_binding":<csvserver_spilloverpolicy_binding[]_value>,
    "csvserver_cspolicy_binding":<csvserver_cspolicy_binding[]_value>,
    "csvserver_feopolicy_binding":<csvserver_feopolicy_binding[]_value>,
    "csvserver_vpnvserver_binding":<csvserver_vpnvserver_binding[]_value>,
    "csvserver_domain_binding":<csvserver_domain_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_cachepolicy_binding

Jul 07, 2017

Binding object showing the cachepolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_cachepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_cachepolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_cachepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_cachepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_cachepolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_cachepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"csvserver_cachepolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_cmppolicy_binding

Jul 07, 2017

Binding object showing the cmppolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_cmppolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_cmppolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotoprioityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_cmppolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_cmppolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_cmppolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "sc":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_cmppolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"csvserver_cmppolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_cspolicy_binding

May 20, 2016

Binding object showing the cspolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
targetlbvserver	<String>	Read-write	target vserver name.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
rule	<String>	Read-only	Rule.
hits	<Double>	Read-only	Number of hits.
pipolicyhits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_cspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_cspolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_cspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_cspolicy_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_cspolicy_binding": [ {
      "priority":<Double_value>,
      "bindpoint":<String_value>,
      "policyname":<String_value>,
      "labelname":<String_value>,
      "name":<String_value>,
      "gotopriorityexpression":<String_value>,
      "targetlbvserver":<String_value>,
      "invoke":<Boolean_value>,
      "labeltype":<String_value>,
      "rule":<String_value>,
      "hits":<Double_value>,
      "pipolicyhits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_cspolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{"csvserver_cspolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_domain_binding

May 20, 2016

Binding object showing the domain that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
backupip	<String>	Read-write	. Minimum length = 1
ttl	<Double>	Read-write	. Minimum value = 1
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
domainname	<String>	Read-write	Domain name for which to change the time to live (TTL) and/or backup service IP address. Minimum length = 1
sitedomainttl	<Double>	Read-write	. Minimum value = 1
cookiedomain	<String>	Read-write	. Minimum length = 1
cookietimeout	<Double>	Read-write	. Minimum value = 0 Maximum value = 1440
appflowlog	<String>	Read-only	Enable logging appflow flow information. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_domain_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_domain_binding":{
    "name":<String_value>,
    "domainname":<String_value>,
    "ttl":<Double_value>,
    "backupip":<String_value>,
    "cookiedomain":<String_value>,
    "cookietimeout":<Double_value>,
    "sitedomainttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_domain_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_domain_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_domain_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_domain_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_domain_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_domain_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_domain_binding": [ {
    "backupip":<String_value>,
    "ttl":<Double_value>,
    "name":<String_value>,
    "domainname":<String_value>,
    "sitedomainttl":<Double_value>,
    "cookiedomain":<String_value>,
    "cookietimeout":<Double_value>,
    "appflowlog":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_domain_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"csvserver_domain_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_feopolicy_binding

May 20, 2016

Binding object showing the feopolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_feopolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_feopolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_feopolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_feopolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_feopolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_feopolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"csvserver_feopolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_filterpolicy_binding

May 20, 2016

Binding object showing the filterpolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT ? Evaluate the policy with the next higher priority number. * END ? End policy evaluation. * USE_INVOCATION_RESULT ? Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy?s priority. * The expression evaluates to a priority number that is between the current policy?s priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
sc	<String>	Read-only	The state of SureConnect the specified virtual server. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_filterpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_filterpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_filterpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_filterpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_filterpolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "gotopriorityexpression":<String_value>,
    "targetlbvserver":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "sc":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_filterpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"csvserver_filterpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_gslbvserver_binding

Jul 07, 2017

Binding object showing the gslbvserver that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<String>	Read-write	Name of the default gslb or vpn vserver bound to CS vserver of type GSLB/VPN. For Example: bind cs vserver cs1 -vserver gslb1 or bind cs vserver cs1 -vserver vpn1. Minimum length = 1
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_gslbvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_gslbvserver_binding":{
    "name":<String_value>,
    "vserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_gslbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_gslbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_gslbvserver_binding": [ {
    "vserver":<String_value>,
    "name":<String_value>,
    "hits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_gslbvserver_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"csvserver_gslbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_lbvserver_binding

Jul 07, 2017

Binding object showing the lbvserver that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetvserver	<String>	Read-write	The virtual server name (created with the add lb vserver command) to which content will be switched. Minimum length = 1
lbvserver	<String>	Read-write	Name of the default lb vserver bound. Use this param for Default binding only. For Example: bind cs vserver cs1 -lbvserver lb1. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_lbvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_lbvserver_binding":{
    "name":<String_value>,
    "lbvserver":<String_value>,
    "targetvserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_lbvserver_binding": [ {
    "name":<String_value>,
    "targetvserver":<String_value>,
    "lbvserver":<String_value>,
    "hits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"csvserver_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_responderpolicy_binding

May 20, 2016

Binding object showing the responderpolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_responderpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_responderpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_responderpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_responderpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_responderpolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_responderpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"csvserver_responderpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_rewritepolicy_binding

May 20, 2016

Binding object showing the rewritepolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_rewritepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_rewritepolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_rewritepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_rewritepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_rewritepolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_rewritepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"csvserver_rewritepolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_spilloverpolicy_binding

May 20, 2016

Binding object showing the spilloverpolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_spilloverpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_spilloverpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_spilloverpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_spilloverpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_spilloverpolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_spilloverpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"csvserver_spilloverpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_tmtrafficpolicy_binding

May 20, 2016

Binding object showing the tmtrafficpolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	For a rewrite policy, the bind point to which to bind the policy. Note: This parameter applies only to rewrite policies, because content switching policies are evaluated only at request time. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label to be invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT ? Evaluate the policy with the next higher priority number. * END ? End policy evaluation. * USE_INVOCATION_RESULT ? Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy?s priority. * The expression evaluates to a priority number that is between the current policy?s priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
invoke	<Boolean>	Read-write	Invoke a policy label if this policy's rule evaluates to TRUE (valid only for default-syntax policies such as application firewall, transform, integrated cache, rewrite, responder, and content switching).
labeltype	<String>	Read-write	Type of label to be invoked. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_tmtrafficpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_tmtrafficpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_tmtrafficpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_tmtrafficpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_tmtrafficpolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "gotopriorityexpression":<String_value>,
    "targetlbvserver":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

---

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_tmtrafficpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"csvserver_tmtrafficpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_transformpolicy_binding

May 20, 2016

Binding object showing the transformpolicy that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority for the policy.
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
policyname	<String>	Read-write	Policies bound to this vserver.
labelname	<String>	Read-write	Name of the label invoked.
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
targetlbvserver	<String>	Read-write	Name of the Load Balancing virtual server to which the content is switched, if policy rule is evaluated to be TRUE. Example: bind cs vs cs1 -policyname pol1 -priority 101 -targetLBVserver lb1 Note: Use this parameter only in case of Content Switching policy bind operations to a CS vserver. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_transformpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_transformpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "targetlbvserver":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_transformpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_transformpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_transformpolicy_binding": [ {
    "priority":<Double_value>,
    "bindpoint":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "name":<String_value>,
    "targetlbvserver":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_transformpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"csvserver_transformpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to csvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<String>	Read-write	Name of the default gslb or vpn vserver bound to CS vserver of type GSLB/VPN. For Example: bind cs vserver cs1 -vserver gslb1 or bind cs vserver cs1 -vserver vpn1. Minimum length = 1
name	<String>	Read-write	Name of the content switching virtual server to which the content switching policy applies. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/csvserver_vpnvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"csvserver_vpnvserver_binding":{
    "name":<String_value>,
    "vserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of csvserver_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the csvserver_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver_vpnvserver_binding": [ {
    "vserver":<String_value>,
    "name":<String_value>,
    "hits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/csvserver_vpnvserver_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"csvserver_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Db

May 20, 2016

All the commands associated with database user

Name	Description
dbdbprofile	Configuration for DB profile
dbuser	Configuration for DB user

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dbdbprofile

May 20, 2016

Configuration for DB profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the database profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Cannot be changed after the profile is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my profile" or 'my profile'). . Minimum length = 1 Maximum length = 127
interpretquery	<String>	Read-write	If ENABLED, inspect the query and update the connection information, if required. If DISABLED, forward the query to the server. Default value: YES Possible values = YES, NO
stickiness	<String>	Read-write	If the queries are related to each other, forward to the same backend server. Default value: NO Possible values = YES, NO
kcdaccount	<String>	Read-write	Name of the KCD account that is used for Windows authentication. Minimum length = 1 Maximum length = 127
conmultiplex	<String>	Read-write	Use the same server-side connection for multiple client-side requests. Default is enabled. Default value: ENABLED Possible values = ENABLED, DISABLED
enablecachingconmuxoff	<String>	Read-write	Enable caching when connection multiplexing is OFF. Default value: DISABLED Possible values = ENABLED, DISABLED
refcnt	<Double>	Read-only	Profile Reference Count.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dbdbprofile":{
    "name":<String_value>,
    "interpretquery":<String_value>,
    "stickiness":<String_value>,
    "kcdaccount":<String_value>,
    "conmultiplex":<String_value>,
    "enablecachingconmuxoff":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dbdbprofile":{
    "name":<String_value>,
    "interpretquery":<String_value>,
    "stickiness":<String_value>,
    "kcdaccount":<String_value>,
    "conmultiplex":<String_value>,
    "enablecachingconmuxoff":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dbdbprofile":{
    "name":<String_value>,
    "interpretquery":true,
    "stickiness":true,
    "kcdaccount":true,
    "conmultiplex":true,
    "enablecachingconmuxoff":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dbdbprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dbdbprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dbdbprofile": [ {
      "name":<String_value>,
      "interpretquery":<String_value>,
      "stickiness":<String_value>,
      "kcdaccount":<String_value>,
      "conmultiplex":<String_value>,
      "refcnt":<Double_value>,
      "enablecachingconmuxoff":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile/name_value<String>?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dbdbprofile": [ {
    "name":<String_value>,
    "interpretquery":<String_value>,
    "stickiness":<String_value>,
    "kcdaccount":<String_value>,
    "conmultiplex":<String_value>,
    "refcnt":<Double_value>,
    "enablecachingconmuxoff":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dbdbprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{ "dbdbprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dbuser

May 20, 2016

Configuration for DB user resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	Name of the database user. Must be the same as the user name specified in the database. Minimum length = 1
password	<String>	Read-write	Password for logging on to the database. Must be the same as the password specified in the database. Minimum length = 1
loggedin	<Boolean>	Read-write	Display the names of all database users currently logged on to the NetScaler appliance.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dbuser

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dbuser":{
    "username":<String_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dbuser/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/dbuser

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dbuser":{
    "username":<String_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dbuser

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dbuser?**args=username:<String_value>,loggedin:<Boolean_value>**

Use this query-parameter to get dbuser resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dbuser?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dbuser?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dbuser resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dbuser?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dbuser?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dbuser resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dbuser": [ {
username:<String_value>,loggedin:<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dbuser/username_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dbuser/username_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dbuser/username_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dbuser": [ {
username:<String_value>,loggedin:<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dbuser?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dbuser": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Domain Name Service

May 20, 2016

Domain Name Service(DNS) configuration.

Name	Description
dnsaaaarec	Configuration for IPv6 address type record
dnsaction	Configuration for DNS action
dnsaction64	Configuration for dns64 action
dnsaddrec	Configuration for address type record
dnscnamerec	Configuration for CNAME record
dnsglobal_binding	Binding object showing the resources that can be bound to dnsglobal
dnsglobal_dnspolicy_binding	Binding object showing the dnspolicy that can be bound to dnsglobal
dnskey	
dnsmxrec	Configuration for MX record
dnsnameserver	Configuration for name server
dnsnaptrrec	Configuration for NAPTR record
dnsnsecrec	
dnsnsrec	Configuration for name server record
dnsparameter	Configuration for DNS parameter
dnspolicy	Configuration for DNS policy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicy64	
Name	**Description**
dnspolicy64_binding	Binding object showing the resources that can be bound to dnspolicy64
dnspolicy64_lbvserver_binding	Binding object showing the lbvserver that can be bound to dnspolicy64
dnspolicy_binding	Binding object showing the resources that can be bound to dnspolicy
dnspolicy_dnsglobal_binding	Binding object showing the dnsglobal that can be bound to dnspolicy
dnspolicy_dnspolicylabel_binding	Binding object showing the dnspolicylabel that can be bound to dnspolicy
dnspolicylabel	Configuration for dns policy label
dnspolicylabel_binding	Binding object showing the resources that can be bound to dnspolicylabel
dnspolicylabel_dnspolicy_binding	Binding object showing the dnspolicy that can be bound to dnspolicylabel
dnspolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to dnspolicylabel
dnsprofile	Configuration for DNS profile
dnsproxyrecords	Configuration for proxy record
dnsptrrec	Configuration for PTR record
dnssoarec	Configuration for SOA record
dnssrvrec	Configuration for server record
dnssuffix	Configuration for DNS suffix
dnstxtrec	Configuration for TXT record
dnsview	Configuration for DNS view
dnsview_binding	Binding object showing the resources that can be bound to dnsview

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsview_dnspolicy_binding	
Name	**Description**
dnsview_gslbservice_binding	Binding object showing the gslbservice that can be bound to dnsview
dnszone	Configuration for DNS zone
dnszone_binding	Binding object showing the resources that can be bound to dnszone
dnszone_dnskey_binding	Binding object showing the dnskey that can be bound to dnszone
dnszone_domain_binding	Binding object showing the domain that can be bound to dnszone

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsaaaarec

May 20, 2016

Configuration for IPv6 address type record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
hostname	<String>	Read-write	Domain name. Minimum length = 1
ipv6address	<String>	Read-write	One or more IPv6 addresses to assign to the domain name. Minimum length = 1
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the NetScaler appliance uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
vservername	<String>	Read-only	Vitual server name.
authtype	<String>	Read-only	Authentication type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaaaarec

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsaaaarec":{
    "hostname":<String_value>,
    "ipv6address":<String_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaaaarec/hostname_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaaaarec

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnsaaaarec?**args=hostname:<String_value>,ipv6address:<String_value>,type:<String_value>**

Use this query-parameter to get dnsaaaarec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsaaaarec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsaaaarec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsaaaarec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsaaaarec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsaaaarec?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsaaaarec resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsaaaarec": [ {
hostname:<String_value>,ipv6address:<String_value>,type:<String_value>      "ttl":<Double_value>,
      "vservername":<String_value>,
      "authtype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaaaarec?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsaaaarec": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsaction

Jul 07, 2017

Configuration for DNS action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
actionname	<String>	Read-write	Name of the dns action.
actiontype	<String>	Read-write	The type of DNS action that is being configured. Possible values = ViewName, GslbPrefLoc, noop, Drop, Cache_Bypass, Rewrite_Response
ipaddress	<String[]>	Read-write	List of IP address to be returned in case of rewrite_response actiontype. They can be of IPV4 or IPV6 type. In case of set command We will remove all the IP address previously present in the action and will add new once given in set dns action command.
ttl	<Double>	Read-write	Time to live, in seconds. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
viewname	<String>	Read-write	The view name that must be used for the given action.
preferredloclist	<String[]>	Read-write	The location list in priority order used for the given action. Minimum length = 1
dnsprofilename	<String>	Read-write	Name of the DNS profile to be associated with the transaction for which the action is chosen. Minimum length = 1 Maximum length = 127
drop	<String>	Read-only	The dns packet must be dropped. Possible values = YES, NO
cachebypass	<String>	Read-only	By pass dns cache for this. Possible values = YES, NO
builtin	<String[]>	Read-only	Flag to determine whether DNS action is default or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsaction":{
    "actionname":<String_value>,
    "actiontype":<String_value>,
    "ipaddress":<String[]_value>,
    "ttl":<Double_value>,
    "viewname":<String_value>,
    "preferredloclist":<String[]_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "dnsprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction/actionname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsaction":{
    "actionname":<String_value>,
    "ipaddress":<String[]_value>,
    "ttl":<Double_value>,
    "viewname":<String_value>,
    "preferredloclist":<String[]_value>,
    "dnsprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsaction":{
    "actionname":<String_value>,
    "ttl":true,
    "dnsprofilename":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/dnsaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsaction": [ {
      "actionname":<String_value>,
      "actiontype":<String_value>,
      "ttl":<Double_value>,
      "ipaddress":<String[]_value>,
      "viewname":<String_value>,
      "preferredloclist":<String[]_value>,
      "drop":<String_value>,
      "cachebypass":<String_value>,
      "dnsprofilename":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction/actionname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsaction/actionname_value<String>?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsaction/actionname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsaction": [ {
      "actionname":<String_value>,
      "actiontype":<String_value>,
      "ttl":<Double_value>,
      "ipaddress":<String[]_value>,
      "viewname":<String_value>,
      "preferredloclist":<String[]_value>,
      "drop":<String_value>,
      "cachebypass":<String_value>,
      "dnsprofilename":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "dnsaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsaction64

Jul 07, 2017

Configuration for dns64 action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
actionname	<String>	Read-write	Name of the dns64 action.
prefix	<String>	Read-write	The dns64 prefix to be used if the after evaluating the rules.
mappedrule	<String>	Read-write	The expression to select the criteria for ipv4 addresses to be used for synthesis. Only if the mappedrule is evaluated to true the corresponding ipv4 address is used for synthesis using respective prefix, otherwise the A RR is discarded.
excluderule	<String>	Read-write	The expression to select the criteria for eliminating the corresponding ipv6 addresses from the response.
builtin	<String[]>	Read-only	Flag to determine whether dna64action is default or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction64

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsaction64":{
    "actionname":<String_value>,
    "prefix":<String_value>,
    "mappedrule":<String_value>,
    "excluderule":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction64/actionname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction64

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsaction64":{
    "actionname":<String_value>,
    "prefix":<String_value>,
    "mappedrule":<String_value>,
    "excluderule":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction64?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsaction64":{
    "actionname":<String_value>,
    "prefix":true,
    "mappedrule":true,
    "excluderule":true
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction64

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsaction64?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsaction64?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsaction64 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsaction64?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsaction64?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsaction64 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsaction64": [ {
      "actionname":<String_value>,
      "prefix":<String_value>,
      "mappedrule":<String_value>,
      "excluderule":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction64/actionname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsaction64/actionname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsaction64/actionname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "dnsaction64": [ {
      "actionname":<String_value>,
      "prefix":<String_value>,
      "mappedrule":<String_value>,
      "excluderule":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaction64?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsaction64": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsaddrec

May 20, 2016

Configuration for address type record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
hostname	<String>	Read-write	Domain name. Minimum length = 1
ipaddress	<String>	Read-write	One or more IPv4 addresses to assign to the domain name. Minimum length = 1
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the NetScaler appliance uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
type	<String>	Read-write	The address record type. The type can take 3 values: ADNS - If this is specified, all of the authoritative address records will be displayed. PROXY - If this is specified, all of the proxy address records will be displayed. ALL - If this is specified, all of the address records will be displayed. Possible values = ALL, ADNS, PROXY
vservername	<String>	Read-only	Vitual server name.
authtype	<String>	Read-only	Authentication type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsaddrec":{
    "hostname":<String_value>,
    "ipaddress":<String_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec/hostname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec?**args=hostname:<String_value>,type:<String_value>**

Use this query-parameter to get dnsaddrec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsaddrec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec?**pagesize=#no&pageno=#no**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the dnsaddrec resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsaddrec": [ {
hostname:<String_value>,type:<String_value>      "ipaddress":<String_value>,
      "ttl":<Double_value>,
      "vservername":<String_value>,
      "authtype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec/hostname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec/hostname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec/hostname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsaddrec": [ {
hostname:<String_value>,type:<String_value>      "ipaddress":<String_value>,
      "ttl":<Double_value>,
      "vservername":<String_value>,
      "authtype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsaddrec?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsaddrec": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnscnamerec

May 20, 2016

Configuration for CNAME record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
aliasname	<String>	Read-write	Alias for the canonical domain name. Minimum length = 1
canonicalname	<String>	Read-write	Canonical domain name. Minimum length = 1
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the NetScaler appliance uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Default value: ADNS Possible values = ALL, ADNS, PROXY
vservername	<String>	Read-only	GSLB Vitual server name to which this domain is bound.
authtype	<String>	Read-only	Record type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnscnamerec":{
    "aliasname":<String_value>,
    "canonicalname":<String_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec/aliasname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec?**args=aliasname:<String_value>,type:<String_value>**

Use this query-parameter to get dnscnamerec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnscnamerec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec?**pagesize=#no&pageno=#no**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the dnscnamerec resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnscnamerec": [ {
aliasname:<String_value>,type:<String_value>      "canonicalname":<String_value>,
      "ttl":<Double_value>,
      "vservername":<String_value>,
      "authtype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec/aliasname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec/aliasname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec/aliasname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnscnamerec": [ {
aliasname:<String_value>,type:<String_value>      "canonicalname":<String_value>,
      "ttl":<Double_value>,
      "vservername":<String_value>,
      "authtype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnscnamerec?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnscnamerec": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsglobal_binding

May 20, 2016

Binding object which returns the resources bound to dnsglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
dnsglobal_dnspolicy_binding	<dnsglobal_dnspolicy_binding[]>	Read-only	dnspolicy that can be bound to dnsglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "dnsglobal_binding": [ {
    "dnsglobal_dnspolicy_binding":<dnsglobal_dnspolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsglobal_dnspolicy_binding

May 20, 2016

Binding object showing the dnspolicy that can be bound to dnsglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy with which it is bound. Maximum allowed priority should be less than 65535.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the dns policy.
labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label. Minimum length = 1
invoke	<Boolean>	Read-write	Invoke flag.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
type	<String>	Read-write	Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = policylabel
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
flowtype	<Double>	Read-only	flowtype of the bound rewrite policy.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/dnsglobal_dnspolicy_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"dnsglobal_dnspolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsglobal_dnspolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsglobal_dnspolicy_binding

Query-parameters:

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/dnsglobal_dnspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnsglobal_dnspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsglobal_dnspolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsglobal_dnspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsglobal_dnspolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "type":<String_value>,
    "labeltype":<String_value>,
    "numpol":<Double_value>,
    "flowtype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsglobal_dnspolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "dnsglobal_dnspolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnskey

May 20, 2016

Configuration for 0 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
keyname	<String>	Read-write	Name of the public-private key pair to publish in the zone. Minimum length = 1
publickey	<String>	Read-write	File name of the public key.
privatekey	<String>	Read-write	File name of the private key.
expires	<Double>	Read-write	Time period for which to consider the key valid, after the key is used to sign a zone. Default value: 120 Minimum value = 1 Maximum value = 32767
units1	<String>	Read-write	Units for the expiry period. Default value: DAYS Possible values = MINUTES, HOURS, DAYS
notificationperiod	<Double>	Read-write	Time at which to generate notification of key expiration, specified as number of days, hours, or minutes before expiry. Must be less than the expiry period. The notification is an SNMP trap sent to an SNMP manager. To enable the appliance to send the trap, enable the DNSKEY-EXPIRY SNMP alarm. Default value: 7 Minimum value = 1 Maximum value = 32767
units2	<String>	Read-write	Units for the notification period. Default value: DAYS Possible values = MINUTES, HOURS, DAYS
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the DNSKEY resource record created in the zone. TTL is the time for which the record must be cached by the DNS proxies. If the TTL is not specified, either the DNS zone's minimum TTL or the default value of 3600 is used. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
			Name of the zone for which to create a key.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
keytype	<String>	Read-write	Type of key to create. Default value: NS_DNSKEY_ZSK Possible values = KSK, KeySigningKey, ZSK, ZoneSigningKey
algorithm	<String>	Read-write	Algorithm to generate for zone signing. Default value: NS_DNSKEYALGO_RSASHA1 Possible values = RSASHA1
keysize	<Double>	Read-write	Size of the key, in bits. Default value: 512
filenameprefix	<String>	Read-write	Common prefix for the names of the generated public and private key files and the Delegation Signer (DS) resource record. During key generation, the .key, .private, and .ds suffixes are appended automatically to the file name prefix to produce the names of the public key, the private key, and the DS record, respectively.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | CREATE | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnskey

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnskey":{
    "keyname":<String_value>,
    "publickey":<String_value>,
    "privatekey":<String_value>,
    "expires":<Double_value>,
    "units1":<String_value>,
    "notificationperiod":<Double_value>,
    "units2":<String_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

create

URL: http://<netscaler-ip-address>/nitro/v1/config/dnskey?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnskey":{
    "zonename":<String_value>,
    "keytype":<String_value>,
    "algorithm":<String_value>,
    "keysize":<Double_value>,
    "filenameprefix":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnskey

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnskey":{
    "keyname":<String_value>,
    "expires":<Double_value>,
    "units1":<String_value>,
    "notificationperiod":<Double_value>,
    "units2":<String_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnskey?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnskey":{
    "keyname":<String_value>,
    "expires":true,
    "units1":true,
    "notificationperiod":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "units2":true,
    "ttl":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnskey/keyname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnskey

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnskey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnskey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnskey resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnskey?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnskey?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnskey resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnskey": [ {
      "keyname":<String_value>,
      "publickey":<String_value>,
      "privatekey":<String_value>,
      "expires":<Double_value>,
      "units1":<String_value>,
      "notificationperiod":<Double_value>,
      "units2":<String_value>,
      "ttl":<Double_value>,
      "zonename":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnskey/keyname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnskey/keyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/dnskey/keyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnskey": [ {
      "keyname":<String_value>,
      "publickey":<String_value>,
      "privatekey":<String_value>,
      "expires":<Double_value>,
      "units1":<String_value>,
      "notificationperiod":<Double_value>,
      "units2":<String_value>,
      "ttl":<Double_value>,
      "zonename":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnskey?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnskey": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsmxrec

May 20, 2016

Configuration for MX record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
domain	<String>	Read-write	Domain name for which to add the MX record. Minimum length = 1
mx	<String>	Read-write	Host name of the mail exchange server. Minimum length = 1
pref	<Double>	Read-write	Priority number to assign to the mail exchange server. A domain name can have multiple mail servers, with a priority number assigned to each server. The lower the priority number, the higher the mail server's priority. When other mail servers have to deliver mail to the specified domain, they begin with the mail server with the lowest priority number, and use other configured mail servers, in priority order, as backups. Minimum value = 0 Maximum value = 65535
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the NetScaler appliance uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Default value: ADNS Possible values = ALL, ADNS, PROXY
authtype	<String>	Read-only	Record type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsmxrec":{
    "domain":<String_value>,
    "mx":<String_value>,
    "pref":<Double_value>,
    "ttl":<Double_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec/domain_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsmxrec":{
    "domain":<String_value>,
    "mx":<String_value>,
    "pref":<Double_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsmxrec":{
    "domain":<String_value>,
    "mx":<String_value>,
    "ttl":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec?**args=domain:<String_value>,type:<String_value>**

Use this query-parameter to get dnsmxrec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsmxrec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsmxrec resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsmxrec": [ {
domain:<String_value>,type:<String_value>      "mx":<String_value>,
      "pref":<Double_value>,
      "ttl":<Double_value>,
      "authtype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec/domain_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec/domain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec/domain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsmxrec": [ {
domain:<String_value>,type:<String_value>      "mx":<String_value>,
      "pref":<Double_value>,
      "ttl":<Double_value>,
      "authtype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsmxrec?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "dnsmxrec": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsnameserver

May 20, 2016

Configuration for name server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ip	<String>	Read-write	IP address of an external name server or, if the Local parameter is set, IP address of a local DNS server (LDNS). Minimum length = 1
dnsvservername	<String>	Read-write	Name of a DNS virtual server. Overrides any IP address-based name servers configured on the NetScaler appliance. Minimum length = 1
local	<Boolean>	Read-write	Mark the IP address as one that belongs to a local recursive DNS server on the NetScaler appliance. The appliance recursively resolves queries received on an IP address that is marked as being local. For recursive resolution to work, the global DNS parameter, Recursion, must also be set. If no name server is marked as being local, the appliance functions as a stub resolver and load balances the name servers.
state	<String>	Read-write	Administrative state of the name server. Default value: ENABLED Possible values = ENABLED, DISABLED
type	<String>	Read-write	Protocol used by the name server. UDP_TCP is not valid if the name server is a DNS virtual server configured on the appliance. Default value: UDP Possible values = UDP, TCP, UDP_TCP
dnsprofilename	<String>	Read-write	Name of the DNS profile to be associated with the name server. Minimum length = 1
servicename	<String>	Read-only	The name of the dns vserver.
port	<Integer>	Read-only	Port of the service. Range 1 - 65535
nameserverstate	<String>	Read-only	State of the server. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
clmonview	<Double>	Read-only	Tells the view id by which state of the service is updated.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | ENABLE | DISABLE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"dnsnameserver":{
    "ip":<String_value>,
    "dnsvservername":<String_value>,
    "local":<Boolean_value>,
    "state":<String_value>,
    "type":<String_value>,
    "dnsprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsnameserver":{
    "ip":<String_value>,
    "dnsprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"dnsnameserver":{
    "ip":<String_value>,
    "dnsprofilename":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver/ip_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsnameserver":{
    "ip":<String_value>,
    "dnsvservername":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsnameserver":{
    "ip":<String_value>,
    "dnsvservername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsnameserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsnameserver resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsnameserver": [ {
      "ip":<String_value>,
      "dnsvservername":<String_value>,
      "servicename":<String_value>,
      "port":<Integer_value>,
      "type":<String_value>,
      "state":<String_value>,
      "nameserverstate":<String_value>,
      "local":<Boolean_value>,
      "clmonowner":<Double_value>,
      "clmonview":<Double_value>,
      "dnsprofilename":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnameserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "dnsnameserver": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsnaptrrec

May 20, 2016

Configuration for NAPTR record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
domain	<String>	Read-write	Name of the domain for the NAPTR record. Minimum length = 1
order	<Double>	Read-write	An integer specifying the order in which the NAPTR records MUST be processed in order to accurately represent the ordered list of Rules. The ordering is from lowest to highest. Minimum value = 0 Maximum value = 65535
preference	<Double>	Read-write	An integer specifying the preference of this NAPTR among NAPTR records having same order. lower the number, higher the preference. Minimum value = 0 Maximum value = 65535
flags	<String>	Read-write	flags for this NAPTR. Maximum length = 255
services	<String>	Read-write	Service Parameters applicable to this delegation path. Maximum length = 255
regexp	<String>	Read-write	The regular expression, that specifies the substitution expression for this NAPTR. Maximum length = 255
replacement	<String>	Read-write	The replacement domain name for this NAPTR. Maximum length = 255
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the NetScaler appliance uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
			Unique, internally generated record ID. View the details of the naptr record to obtain its record ID. Records can be removed by either specifying the domain name and record id OR by specifying

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			command. Minimum value = 1 Maximum value = 65535
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Default value: ADNS Possible values = ALL, ADNS, PROXY
authtype	<String>	Read-only	Authentication type. Possible values = ALL, ADNS, PROXY
vservername	<String>	Read-only	Virtual server name.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsnaptrrec":{
    "domain":<String_value>,
    "order":<Double_value>,
    "preference":<Double_value>,
    "flags":<String_value>,
    "services":<String_value>,
    "regexp":<String_value>,
    "replacement":<String_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec/domain_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec?**args=domain:<String_value>,type:<String_value>**

Use this query-parameter to get dnsnaptrrec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsnaptrrec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsnaptrrec resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "dnsnaptrrec": [ {
domain:<String_value>,type:<String_value>      "order":<Double_value>,
      "preference":<Double_value>,
      "flags":<String_value>,
      "services":<String_value>,
      "regexp":<String_value>,
      "replacement":<String_value>,
      "ttl":<Double_value>,
      "recordid":<Double_value>,
      "authtype":<String_value>,
      "vservername":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec/domain_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec/domain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec/domain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsnaptrrec": [ {
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
domain:<String_value>,type:<String_value>      "order":<Double_value>,
      "preference":<Double_value>,
      "flags":<String_value>,
      "services":<String_value>,
      "regexp":<String_value>,
      "replacement":<String_value>,
      "ttl":<Double_value>,
      "recordid":<Double_value>,
      "authtype":<String_value>,
      "vservername":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnaptrrec?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsnaptrrec": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsnsecrec

May 20, 2016

Configuration for 0 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
hostname	<String>	Read-write	Name of the domain. Minimum length = 1
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
nextnsec	<String>	Read-only	Next nsec record in the chain.
nextrecs	<String[]>	Read-only	An array of record types associated with the nsec record. Possible values = A, NS, CNAME, SOA, MX, AAAA, SRV, RRSIG, NSEC, DNSKEY, PTR, TXT, NAPTR
ttl	<Double>	Read-only	Time to Live (TTL), in seconds, for the record.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnsecrec

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnsnsecrec?**args=hostname:<String_value>,type:<String_value>**

Use this query-parameter to get dnsnsecrec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsnsecrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsnsecrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsnsecrec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsnsecrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsnsecrec?**pagesize=#no&pageno=#no**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the dnsnsecrec resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsnsecrec": [ {
hostname:<String_value>,type:<String_value>      "nextnsec":<String_value>,
      "nsecbitarray":<String[]_value>,
      "nextrecs":<String[]_value>,
      "ttl":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnsecrec/hostname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsnsecrec/hostname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsnsecrec/hostname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsnsecrec": [ {
hostname:<String_value>,type:<String_value>      "nextnsec":<String_value>,
      "nsecbitarray":<String[]_value>,
      "nextrecs":<String[]_value>,
      "ttl":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnsecrec?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsnsecrec": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsnsrec

May 20, 2016

Configuration for name server record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
domain	<String>	Read-write	Domain name. Minimum length = 1
nameserver	<String>	Read-write	Host name of the name server to add to the domain. Minimum length = 1
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the NetScaler appliance uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
authtype	<String>	Read-only	Record type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsnsrec":{
    "domain":<String_value>,
    "nameserver":<String_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec/domain_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec?**args=domain:<String_value>,type:<String_value>**

Use this query-parameter to get dnsnsrec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsnsrec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsnsrec resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsnsrec": [ {
domain:<String_value>,type:<String_value>      "nameserver":<String_value>,
      "ttl":<Double_value>,
      "authtype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec/domain_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec/domain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec/domain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsnsrec": [ {
domain:<String_value>,type:<String_value>      "nameserver":<String_value>,
      "ttl":<Double_value>,
      "authtype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsnsrec?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsnsrec": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsparameter

Jul 07, 2017

Configuration for DNS parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
retries	<Double>	Read-write	Maximum number of retry attempts when no response is received for a query sent to a name server. Applies to end resolver and forwarder configurations. Default value: 5 Minimum value = 1 Maximum value = 5
minttl	<Double>	Read-write	Minimum permissible time to live (TTL) for all records cached in the DNS cache by DNS proxy, end resolver, and forwarder configurations. If the TTL of a record that is to be cached is lower than the value configured for minTTL, the TTL of the record is set to the value of minTTL before caching. When you modify this setting, the new value is applied only to those records that are cached after the modification. The TTL values of existing records are not changed. Minimum value = 0 Maximum value = 604800
maxttl	<Double>	Read-write	Maximum time to live (TTL) for all records cached in the DNS cache by DNS proxy, end resolver, and forwarder configurations. If the TTL of a record that is to be cached is higher than the value configured for maxTTL, the TTL of the record is set to the value of maxTTL before caching. When you modify this setting, the new value is applied only to those records that are cached after the modification. The TTL values of existing records are not changed. Default value: 604800 Minimum value = 1 Maximum value = 604800
cacherecords	<String>	Read-write	Cache resource records in the DNS cache. Applies to resource records obtained through proxy configurations only. End resolver and forwarder configurations always cache records in the DNS cache, and you cannot disable this behavior. When you disable record caching, the appliance stops caching server responses. However, cached records are not flushed. The appliance does not serve requests from the cache until record caching is enabled again. Default value: YES Possible values = YES, NO
namelookuppriority	<String>	Read-write	Type of lookup (DNS or WINS) to attempt first. If the first-priority lookup fails, the second-priority lookup is attempted. Used only by the SSL VPN feature. Default value: WINS Possible values = WINS, DNS

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
recursion	<String>	Read-write	not hosted on the NetScaler appliance. Also resolve queries recursively when the external name servers configured on the appliance (for a forwarder configuration) are unavailable. When external name servers are unavailable, the appliance queries a root server and resolves the request recursively, as it does for an end resolver configuration. Default value: DISABLED Possible values = ENABLED, DISABLED
resolutionorder	<String>	Read-write	Type of DNS queries (A, AAAA, or both) to generate during the routine functioning of certain NetScaler features, such as SSL VPN, cache redirection, and the integrated cache. The queries are sent to the external name servers that are configured for the forwarder function. If you specify both query types, you can also specify the order. Available settings function as follows: * OnlyAQuery. Send queries for IPv4 address records (A records) only. * OnlyAAAAQuery. Send queries for IPv6 address records (AAAA records) instead of queries for IPv4 address records (A records). * AThenAAAAQuery. Send a query for an A record, and then send a query for an AAAA record if the query for the A record results in a NODATA response from the name server. * AAAAThenAQuery. Send a query for an AAAA record, and then send a query for an A record if the query for the AAAA record results in a NODATA response from the name server. Default value: OnlyAQuery Possible values = OnlyAQuery, OnlyAAAAQuery, AThenAAAAQuery, AAAAThenAQuery
dnssec	<String>	Read-write	Enable or disable the Domain Name System Security Extensions (DNSSEC) feature on the appliance. Note: Even when the DNSSEC feature is enabled, forwarder configurations (used by internal NetScaler features such as SSL VPN and Cache Redirection for name resolution) do not support the DNSSEC OK (DO) bit in the EDNS0 OPT header. Default value: ENABLED Possible values = ENABLED, DISABLED
maxpipeline	<Double>	Read-write	Maximum number of concurrent DNS requests to allow on a single client connection, which is identified by the <clientip:port>-<vserver ip:port> tuple. A value of 0 (zero) applies no limit to the number of concurrent DNS requests allowed on a single client connection.
dnsrootreferral	<String>	Read-write	Send a root referral if a client queries a domain name that is unrelated to the domains configured/cached on the NetScaler appliance. If the setting is disabled, the appliance sends a blank response instead of a root referral. Applicable to domains for which the appliance is authoritative. Disable the parameter when the appliance is under attack from a client that is sending a flood of queries for unrelated domains. Default value: DISABLED Possible values = ENABLED, DISABLED
dns64timeout	<Double>	Read-write	While doing DNS64 resolution, this parameter specifies the time to wait before sending an A query if no response is received from backend DNS server for AAAA query. Minimum value = 0 Maximum value = 10000

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsparameter":{
    "retries":<Double_value>,
    "minttl":<Double_value>,
    "maxttl":<Double_value>,
    "cacherecords":<String_value>,
    "namelookuppriority":<String_value>,
    "recursion":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "resolutionorder":<String_value>,
    "dnssec":<String_value>,
    "maxpipeline":<Double_value>,
    "dnsrootreferral":<String_value>,
    "dns64timeout":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsparameter":{
    "retries":true,
    "minttl":true,
    "maxttl":true,
    "cacherecords":true,
    "namelookuppriority":true,
    "recursion":true,
    "resolutionorder":true,
    "dnssec":true,
    "maxpipeline":true,
    "dnsrootreferral":true,
    "dns64timeout":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsparameter

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsparameter": [ {
      "retries":<Double_value>,
      "minttl":<Double_value>,
      "maxttl":<Double_value>,
      "namelookuppriority":<String_value>,
      "cacherecords":<String_value>,
      "recursion":<String_value>,
      "resolutionorder":<String_value>,
      "dnssec":<String_value>,
      "maxpipeline":<Double_value>,
      "dnsrootreferral":<String_value>,
      "dns64timeout":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicy

May 20, 2016

Configuration for DNS policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the DNS policy.
rule	<String>	Read-write	Expression against which DNS traffic is evaluated. Written in the default syntax. Note: * On the command line interface, if the expression includes blank spaces, the entire expression must be enclosed in double quotation marks. * If the expression itself includes double quotation marks, you must escape the quotations by using the character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' Example: CLIENT.UDP.DNS.DOMAIN.EQ("domainname").
viewname	<String>	Read-write	The view name that must be used for the given policy.
preferredlocation	<String>	Read-write	The location used for the given policy. This is deprecated attribute. Please use -prefLocList.
preferredloclist	<String[]>	Read-write	The location list in priority order used for the given policy. Minimum length = 1
drop	<String>	Read-write	The dns packet must be dropped. Possible values = YES, NO
cachebypass	<String>	Read-write	By pass dns cache for this. Possible values = YES, NO
actionname	<String>	Read-write	Name of the DNS action to perform when the rule evaluates to TRUE. The built in actions function as follows: * dns_default_act_Drop. Drop the DNS request. * dns_default_act_Cachebypass. Bypass the DNS cache and forward the request to the name server. You can create custom actions by using the add dns action command in the CLI or the DNS > Actions > Create DNS Action dialog box in the NetScaler configuration utility.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
logaction	<String>	Read-write	Name of the messagelog action to use for requests that match this policy.
hits	<Double>	Read-only	The number of times the policy has been hit.
undefhits	<Double>	Read-only	Number of Undef hits.
description	<String>	Read-only	Description of the policy.
builtin	<String[]>	Read-only	Flag to determine whether DNS policy is default or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnspolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "viewname":<String_value>,
    "preferredlocation":<String_value>,
    "preferredloclist":<String[]_value>,
    "drop":<String_value>,
    "cachebypass":<String_value>,
    "actionname":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnspolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "viewname":<String_value>,
    "preferredlocation":<String_value>,
    "preferredloclist":<String[]_value>,
    "drop":<String_value>,
    "cachebypass":<String_value>,
    "actionname":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnspolicy":{
    "name":<String_value>,
    "logaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnspolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnspolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "viewname":<String_value>,
      "preferredlocation":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "preferredloclist":<String[]_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "drop":<String_value>,
    "actionname":<String_value>,
    "cachebypass":<String_value>,
    "description":<String_value>,
    "logaction":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "viewname":<String_value>,
    "preferredlocation":<String_value>,
    "preferredloclist":<String[]_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "drop":<String_value>,
    "actionname":<String_value>,
    "cachebypass":<String_value>,
    "description":<String_value>,
    "logaction":<String_value>,
    "builtin":<String[]_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicy64

Jul 07, 2017

Configuration for dns64 policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the DNS64 policy.
rule	<String>	Read-write	Expression against which DNS traffic is evaluated. Written in the default syntax. Note: * On the command line interface, if the expression includes blank spaces, the entire expression must be enclosed in double quotation marks. * If the expression itself includes double quotation marks, you must escape the quotations by using the character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' Example: CLIENT.IP.SRC.IN_SUBENT(23.34.0.0/16).
action	<String>	Read-write	Name of the DNS64 action to perform when the rule evaluates to TRUE. The built in actions function as follows: * A default dns64 action with prefix <default prefix> and mapped and exclude are any You can create custom actions by using the add dns action command in the CLI or the DNS64 > Actions > Create DNS64 Action dialog box in the NetScaler configuration utility.
hits	<Double>	Read-only	The number of times the policy has been hit.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
undefhits	<Double>	Read-only	Number of Undef hits.
description	<String>	Read-only	Description of the policy.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

Name	Data Type	Permissions	Description

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnspolicy64":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnspolicy64":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnspolicy64 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnspolicy64 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicy64": [ {
    "name":<String_value>,
    "rule":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "hits":<Double_value>,
    "action":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "undefhits":<Double_value>,
    "description":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicy64": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "hits":<Double_value>,
    "action":<String_value>,
    "labeltype":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "labelname":<String_value>,
    "undefhits":<Double_value>,
    "description":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicy64": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicy64_binding

May 20, 2016

Binding object which returns the resources bound to dnspolicy64.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the DNS64 policy.
dnspolicy64_lbvserver_binding	<dnspolicy64_lbvserver_binding[]>	Read-only	lbvserver that can be bound to dnspolicy64.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "dnspolicy64_binding": [ {
      "name":<String_value>,
      "dnspolicy64_lbvserver_binding":<dnspolicy64_lbvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicy64_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to dnspolicy64.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the DNS64 policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnspolicy64_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnspolicy64_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicy64_lbvserver_binding": [ {
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy64_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"dnspolicy64_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicylabel

May 20, 2016

Configuration for dns policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the dns policy label.
transform	<String>	Read-write	The type of transformations allowed by the policies bound to the label. Possible values = dns_req, dns_res
newname	<String>	Read-write	The new name of the dns policylabel. Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = policylabel
invoke_labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
flowtype	<Double>	Read-only	Flowtype of the bound dns policy.
description	<String>	Read-only	Description of the policylabel.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default dns policylabel.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnspolicylabel":{
    "labelname":<String_value>,
    "transform":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnspolicylabel":{
    "labelname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnspolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnspolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicylabel": [ {
    "labelname":<String_value>,
    "transform":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "flowtype":<Double_value>,
    "description":<String_value>,
    "isdefault":<Boolean_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicylabel": [ {
    "labelname":<String_value>,
    "transform":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "flowtype":<Double_value>,
    "description":<String_value>,
    "isdefault":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicylabel": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsprofile

May 20, 2016

Configuration for DNS profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
dnsprofilename	<String>	Read-write	Name of the DNS profile. Minimum length = 1 Maximum length = 127
dnsquerylogging	<String>	Read-write	DNS query logging; if enabled, DNS query information such as DNS query id, DNS query flags , DNS domain name and DNS query type will be logged. Default value: DISABLED Possible values = ENABLED, DISABLED
dnsanswerseclogging	<String>	Read-write	DNS answer section; if enabled, answer section in the response will be logged. Default value: DISABLED Possible values = ENABLED, DISABLED
dnsextendedlogging	<String>	Read-write	DNS extended logging; if enabled, authority and additional section in the response will be logged. Default value: DISABLED Possible values = ENABLED, DISABLED
dnserrorlogging	<String>	Read-write	DNS error logging; if enabled, whenever error is encountered in DNS module reason for the error will be logged. Default value: DISABLED Possible values = ENABLED, DISABLED
cacherecords	<String>	Read-write	Cache resource records in the DNS cache. Applies to resource records obtained through proxy configurations only. End resolver and forwarder configurations always cache records in the DNS cache, and you cannot disable this behavior. When you disable record caching, the appliance stops caching server responses. However, cached records are not flushed. The appliance does not serve requests from the cache until record caching is enabled again. Default value: ENABLED Possible values = ENABLED, DISABLED
cachenegativeresponses	<String>	Read-write	Cache negative responses in the DNS cache. When disabled, the appliance stops caching negative responses except referral records. This applies to all configurations - proxy, end resolver, and forwarder. However, cached responses are not flushed. The appliance does not serve negative responses from the cache until this parameter is enabled again.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Possible values = ENABLED, DISABLED
Name	**Data Type**	**Permissions**	**Description**
referencecount	<Double>	Read-only	Number of entities using this profile.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"dnsprofile":{
    "dnsprofilename":<String_value>,
    "dnsquerylogging":<String_value>,
    "dnsanswerseclogging":<String_value>,
    "dnsextendedlogging":<String_value>,
    "dnserrorlogging":<String_value>,
    "cacherecords":<String_value>,
    "cachenegativeresponses":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsprofile":{
    "dnsprofilename":<String_value>,
    "dnsquerylogging":<String_value>,
    "dnsanswerseclogging":<String_value>,
    "dnsextendedlogging":<String_value>,
    "dnserrorlogging":<String_value>,
    "cacherecords":<String_value>,
    "cachenegativeresponses":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsprofile?**action=unset**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsprofile":{
    "dnsprofilename":<String_value>,
    "dnsquerylogging":true,
    "dnsanswerseclogging":true,
    "dnsextendedlogging":true,
    "dnserrorlogging":true,
    "cacherecords":true,
    "cachenegativeresponses":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsprofile/dnsprofilename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsprofile?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsprofile": [ {
    "dnsprofilename":<String_value>,
    "dnsquerylogging":<String_value>,
    "dnsanswerseclogging":<String_value>,
    "dnsextendedlogging":<String_value>,
    "dnserrorlogging":<String_value>,
    "cacherecords":<String_value>,
    "cachenegativeresponses":<String_value>,
    "referencecount":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsprofile/dnsprofilename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsprofile/dnsprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsprofile/dnsprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsprofile": [ {
      "dnsprofilename":<String_value>,
      "dnsquerylogging":<String_value>,
      "dnsanswerseclogging":<String_value>,
      "dnsextendedlogging":<String_value>,
      "dnserrorlogging":<String_value>,
      "cacherecords":<String_value>,
      "cachenegativeresponses":<String_value>,
      "referencecount":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsprofile?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "dnsprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicylabel_binding

Jul 07, 2017

Binding object which returns the resources bound to dnspolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the dns policy label.
dnspolicylabel_policybinding_binding	<dnspolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to dnspolicylabel.
dnspolicylabel_dnspolicy_binding	<dnspolicylabel_dnspolicy_binding[]>	Read-only	dnspolicy that can be bound to dnspolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "dnspolicylabel_binding": [ {
    "labelname":<String_value>,
    "dnspolicylabel_policybinding_binding":<dnspolicylabel_policybinding_binding[]_value>,
    "dnspolicylabel_dnspolicy_binding":<dnspolicylabel_dnspolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicylabel_dnspolicy_binding

May 20, 2016

Binding object showing the dnspolicy that can be bound to dnspolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	The dns policy name.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = policylabel
labelname	<String>	Read-write	Name of the dns policy label.
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/dnspolicylabel_dnspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"dnspolicylabel_dnspolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnspolicylabel_dnspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnspolicylabel_dnspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "dnspolicylabel_dnspolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel_dnspolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"dnspolicylabel_dnspolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicylabel_policybinding_binding

Jul 07, 2017

Binding object showing the policybinding that can be bound to dnspolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	The dns policy name.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = policylabel
labelname	<String>	Read-write	Name of the dns policy label.
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then

© 1999-2017 Citrix Systems, Inc. All rights reserved.

using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnspolicylabel_policybinding_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnspolicylabel_policybinding_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicylabel_policybinding_binding": [ {
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "policyname":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"dnspolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicy_binding

May 20, 2016

Binding object which returns the resources bound to dnspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the DNS policy.
dnspolicy_dnspolicylabel_binding	<dnspolicy_dnspolicylabel_binding[]>	Read-only	dnspolicylabel that can be bound to dnspolicy.
dnspolicy_dnsglobal_binding	<dnspolicy_dnsglobal_binding[]>	Read-only	dnsglobal that can be bound to dnspolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "dnspolicy_binding": [ {
    "name":<String_value>,
    "dnspolicy_dnsglobal_binding":<dnspolicy_dnsglobal_binding[]_value>,
    "dnspolicy_dnspolicylabel_binding":<dnspolicy_dnspolicylabel_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicy_dnsglobal_binding

May 20, 2016

Binding object showing the dnsglobal that can be bound to dnspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the DNS policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy_dnsglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy_dnsglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnspolicy_dnsglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy_dnsglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnspolicy_dnsglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicy_dnsglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy_dnsglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"dnspolicy_dnsglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicy_dnspolicylabel_binding

May 20, 2016

Binding object showing the dnspolicylabel that can be bound to dnspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the DNS policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy_dnspolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy_dnspolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnspolicy_dnspolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnspolicy_dnspolicylabel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnspolicy_dnspolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicy_dnspolicylabel_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnspolicy_dnspolicylabel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"dnspolicy_dnspolicylabel_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsproxyrecords

May 20, 2016

Configuration for proxy record resource.

Operations (click to see Properties)

FLUSH

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

flush

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsproxyrecords?**action=flush**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"dnsproxyrecords":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsptrrec

Jul 07, 2017

Configuration for PTR record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
reversedomain	<String>	Read-write	Reversed domain name representation of the IPv4 or IPv6 address for which to create the PTR record. Use the "in-addr.arpa." suffix for IPv4 addresses and the "ip6.arpa." suffix for IPv6 addresses. Minimum length = 1
domain	<String>	Read-write	Domain name for which to configure reverse mapping. Minimum length = 1
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the NetScaler appliance uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
authtype	<String>	Read-only	Authentication type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsptrrec":{
    "reversedomain":<String_value>,
    "domain":<String_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec/reversedomain_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec?**args=reversedomain:<String_value>,type:<String_value>**

Use this query-parameter to get dnsptrrec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsptrrec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsptrrec resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsptrrec": [ {
reversedomain:<String_value>,type:<String_value>      "domain":<String_value>,
      "ttl":<Double_value>,
      "authtype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec/reversedomain_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec/reversedomain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec/reversedomain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsptrrec": [ {
reversedomain:<String_value>,type:<String_value>      "domain":<String_value>,
      "ttl":<Double_value>,
      "authtype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsptrrec?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsptrrec": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnssoarec

Jul 07, 2017

Configuration for SOA record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
domain	<String>	Read-write	Domain name for which to add the SOA record. Minimum length = 1
originserver	<String>	Read-write	Domain name of the name server that responds authoritatively for the domain. Minimum length = 1
contact	<String>	Read-write	Email address of the contact to whom domain issues can be addressed. In the email address, replace the @ sign with a period (.). For example, enter domainadmin.example.com instead of domainadmin@example.com. Minimum length = 1
serial	<Double>	Read-write	The secondary server uses this parameter to determine whether it requires a zone transfer from the primary server. Default value: 100 Minimum value = 0 Maximum value = 4294967294
refresh	<Double>	Read-write	Time, in seconds, for which a secondary server must wait between successive checks on the value of the serial number. Default value: 3600 Minimum value = 0 Maximum value = 4294967294
retry	<Double>	Read-write	Time, in seconds, between retries if a secondary server's attempt to contact the primary server for a zone refresh fails. Default value: 3 Minimum value = 0 Maximum value = 4294967294
expire	<Double>	Read-write	Time, in seconds, after which the zone data on a secondary name server can no longer be considered authoritative because all refresh and retry attempts made during the period have failed. After the expiry period, the secondary server stops serving the zone. Typically one week. Not used by the primary server. Default value: 3600 Minimum value = 0 Maximum value = 4294967294
			Default time to live (TTL) for all records in the zone. Can be overridden for individual records.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 0 Maximum value = 2147483647
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the NetScaler appliance uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY
authtype	<String>	Read-only	Record type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssoarec

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnssoarec":{
    "domain":<String_value>,
    "originserver":<String_value>,
    "contact":<String_value>,
    "serial":<Double_value>,
    "refresh":<Double_value>,
    "retry":<Double_value>,
    "expire":<Double_value>,
    "minimum":<Double_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssoarec/domain_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssoarec

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnssoarec":{
    "domain":<String_value>,
    "originserver":<String_value>,
    "contact":<String_value>,
    "serial":<Double_value>,
    "refresh":<Double_value>,
    "retry":<Double_value>,
    "expire":<Double_value>,
    "minimum":<Double_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssoarec?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnssoarec":{
    "domain":<String_value>,
    "serial":true,
    "refresh":true,
    "retry":true,
    "expire":true,
    "minimum":true,
    "ttl":true
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssoarec

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnssoarec?**args=domain:<String_value>,type:<String_value>**

Use this query-parameter to get dnssoarec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnssoarec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnssoarec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnssoarec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnssoarec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnssoarec?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnssoarec resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnssoarec": [ {
domain:<String_value>,type:<String_value>      "originserver":<String_value>,
      "contact":<String_value>,
      "serial":<Double_value>,
      "refresh":<Double_value>,
      "retry":<Double_value>,
      "expire":<Double_value>,
      "minimum":<Double_value>,
      "ttl":<Double_value>,
      "authtype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssoarec/domain_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnssoarec/domain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnssoarec/domain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnssoarec": [ {
domain:<String_value>,type:<String_value>      "originserver":<String_value>,
      "contact":<String_value>,
      "serial":<Double_value>,
      "refresh":<Double_value>,
      "retry":<Double_value>,
      "expire":<Double_value>,
      "minimum":<Double_value>,
      "ttl":<Double_value>,
      "authtype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssoarec?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnssoarec": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnssrvrec

May 20, 2016

Configuration for server record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
domain	<String>	Read-write	Domain name, which, by convention, is prefixed by the symbolic name of the desired service and the symbolic name of the desired protocol, each with an underscore (_) prepended. For example, if an SRV-aware client wants to discover a SIP service that is provided over UDP, in the domain example.com, the client performs a lookup for _sip._udp.example.com. Minimum length = 1
target	<String>	Read-write	Target host for the specified service.
priority	<Double>	Read-write	Integer specifying the priority of the target host. The lower the number, the higher the priority. If multiple target hosts have the same priority, selection is based on the Weight parameter. Minimum value = 0 Maximum value = 65535
weight	<Double>	Read-write	Weight for the target host. Aids host selection when two or more hosts have the same priority. A larger number indicates greater weight. Minimum value = 0 Maximum value = 65535
port	<Double>	Read-write	Port on which the target host listens for client requests. Minimum value = 0 Maximum value = 65535
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the NetScaler appliance uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Possible values = ALL, ADNS, PROXY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
authtype	<String>	Read-only	Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"dnssrvrec":{
    "domain":<String_value>,
    "target":<String_value>,
    "priority":<Double_value>,
    "weight":<Double_value>,
    "port":<Double_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec/domain_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnssrvrec":{
    "domain":<String_value>,
    "target":<String_value>,
    "priority":<Double_value>,
    "weight":<Double_value>,
    "port":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnssrvrec":{
    "domain":<String_value>,
    "target":<String_value>,
    "ttl":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec?**args=domain:<String_value>,target:<String_value>,type:<String_value>**

Use this query-parameter to get dnssrvrec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnssrvrec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnssrvrec resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnssrvrec": [ {
domain:<String_value>,target:<String_value>,type:<String_value>      "priority":<Double_value>,
      "weight":<Double_value>,
      "port":<Double_value>,
      "ttl":<Double_value>,
      "authtype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssrvrec?**count=yes**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "dnssrvrec": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnssuffix

May 20, 2016

Configuration for DNS suffix resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
Dnssuffix	<String>	Read-write	Suffix to be appended when resolving domain names that are not fully qualified. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssuffix

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnssuffix":{
    "Dnssuffix":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssuffix/Dnssuffix_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssuffix

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnssuffix?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnssuffix?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnssuffix resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnssuffix?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnssuffix?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnssuffix resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnssuffix": [ {
    "Dnssuffix":<String_value>,

}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssuffix/Dnssuffix_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnssuffix/Dnssuffix_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnssuffix/Dnssuffix_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnssuffix": [ {
      "Dnssuffix":<String_value>,

}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnssuffix?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "dnssuffix": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnstxtrec

May 20, 2016

Configuration for TXT record resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
domain	<String>	Read-write	Name of the domain for the TXT record. Minimum length = 1
String	<String[]>	Read-write	Information to store in the TXT resource record. Enclose the string in single or double quotation marks. A TXT resource record can contain up to six strings, each of which can contain up to 255 characters. If you want to add a string of more than 255 characters, evaluate whether splitting it into two or more smaller strings, subject to the six-string limit, works for you. Maximum length = 255
ttl	<Double>	Read-write	Time to Live (TTL), in seconds, for the record. TTL is the time for which the record must be cached by DNS proxies. The specified TTL is applied to all the resource records that are of the same record type and belong to the specified domain name. For example, if you add an address record, with a TTL of 36000, to the domain name example.com, the TTLs of all the address records of example.com are changed to 36000. If the TTL is not specified, the NetScaler appliance uses either the DNS zone's minimum TTL or, if the SOA record is not available on the appliance, the default value of 3600. Default value: 3600 Minimum value = 0 Maximum value = 2147483647
recordid	<Double>	Read-write	Unique, internally generated record ID. View the details of the TXT record to obtain its record ID. Mutually exclusive with the string parameter. Minimum value = 1 Maximum value = 65535
type	<String>	Read-write	Type of records to display. Available settings function as follows: * ADNS - Display all authoritative address records. * PROXY - Display all proxy address records. * ALL - Display all address records. Default value: ADNS Possible values = ALL, ADNS, PROXY
authtype	<String>	Read-only	Authentication type. Possible values = ALL, ADNS, PROXY
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnstxtrec":{
    "domain":<String_value>,
    "String":<String[]_value>,
    "ttl":<Double_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec/domain_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec?**args=domain:<String_value>,type:<String_value>**

Use this query-parameter to get dnstxtrec resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnstxtrec resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnstxtrec resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnstxtrec": [ {
domain:<String_value>,type:<String_value>      "String":<String[]_value>,
      "ttl":<Double_value>,
      "recordid":<Double_value>,
      "authtype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec/domain_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec/domain_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec/domain_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnstxtrec": [ {
domain:<String_value>,type:<String_value>      "String":<String[]_value>,
      "ttl":<Double_value>,
      "recordid":<Double_value>,
      "authtype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnstxtrec?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{ "dnstxtrec": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsview

May 20, 2016

Configuration for DNS view resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
viewname	<String>	Read-write	Name for the DNS view. Minimum length = 1
flags	<Double>	Read-only	Flags controlling display.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsview

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnsview":{
    "viewname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsview/viewname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsview

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsview?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsview?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dnsview resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsview?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsview?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsview resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsview": [ {
      "viewname":<String_value>,
      "servicename":<String_value>,
      "flags":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsview/viewname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnsview/viewname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnsview/viewname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsview": [ {
      "viewname":<String_value>,
      "servicename":<String_value>,
      "flags":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsview?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "dnsview": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsview_binding

May 20, 2016

Binding object which returns the resources bound to dnsview.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
viewname	<String>	Read-write	Name of the view to display. Minimum length = 1
dnsview_gslbservice_binding	<dnsview_gslbservice_binding[]>	Read-only	gslbservice that can be bound to dnsview.
dnsview_dnspolicy_binding	<dnsview_dnspolicy_binding[]>	Read-only	dnspolicy that can be bound to dnsview.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsview_binding/viewname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "dnsview_binding": [ {
    "viewname":<String_value>,
    "dnsview_dnspolicy_binding":<dnsview_dnspolicy_binding[]_value>,
    "dnsview_gslbservice_binding":<dnsview_gslbservice_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsview_dnspolicy_binding

May 20, 2016

Binding object showing the dnspolicy that can be bound to dnsview.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
viewname	<String>	Read-write	Name of the view to display. Minimum length = 1
dnspolicyname	<String>	Read-write	dnspolicy name of this view.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsview_dnspolicy_binding/viewname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsview_dnspolicy_binding/viewname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnsview_dnspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsview_dnspolicy_binding/viewname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsview_dnspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsview_dnspolicy_binding": [ {
      "viewname":<String_value>,
      "dnspolicyname":<String_value>
}]}
```

---

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsview_dnspolicy_binding/viewname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"dnsview_dnspolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsview_gslbservice_binding

May 20, 2016

Binding object showing the gslbservice that can be bound to dnsview.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
viewname	<String>	Read-write	Name of the view to display. Minimum length = 1
gslbservicename	<String>	Read-write	Service name of the service using this view.
ipaddress	<String>	Read-only	IP of the service corresponding to the given view.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsview_gslbservice_binding/viewname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/dnsview_gslbservice_binding/viewname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnsview_gslbservice_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnsview_gslbservice_binding/viewname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnsview_gslbservice_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsview_gslbservice_binding": [ {
      "viewname":<String_value>,
      "gslbservicename":<String_value>,
      "ipaddress":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnsview_gslbservice_binding/viewname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"dnsview_gslbservice_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnszone

May 20, 2016

Configuration for DNS zone resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
zonename	<String>	Read-write	Name of the zone to create. Minimum length = 1
proxymode	<String>	Read-write	Deploy the zone in proxy mode. Enable in the following scenarios: * The load balanced DNS servers are authoritative for the zone and all resource records that are part of the zone. * The load balanced DNS servers are authoritative for the zone, but the NetScaler appliance owns a subset of the resource records that belong to the zone (partial zone ownership configuration). Typically seen in global server load balancing (GSLB) configurations, in which the appliance responds authoritatively to queries for GSLB domain names but forwards queries for other domain names in the zone to the load balanced servers. In either scenario, do not create the zone's Start of Authority (SOA) and name server (NS) resource records on the appliance. Disable if the appliance is authoritative for the zone, but make sure that you have created the SOA and NS records on the appliance before you create the zone. Default value: ENABLED Possible values = YES, NO
dnssecoffload	<String>	Read-write	Enable dnssec offload for this zone. Default value: DISABLED Possible values = ENABLED, DISABLED
nsec	<String>	Read-write	Enable nsec generation for dnssec offload. Default value: DISABLED Possible values = ENABLED, DISABLED
keyname	<String[]>	Read-write	Name of the public/private DNS key pair with which to sign the zone. You can sign a zone with up to four keys. Minimum length = 1
type	<String>	Read-write	Type of zone to display. Mutually exclusive with the DNS Zone (zoneName) parameter. Available settings function as follows: * ADNS - Display all the zones for which the NetScaler appliance is authoritative. * PROXY - Display all the zones for which the NetScaler appliance is functioning as a proxy server. * ALL - Display all the zones configured on the appliance. Possible values = ALL, ADNS, PROXY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | SIGN | UNSIGN | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnszone":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "zonename":<String_value>,
    "proxymode":<String_value>,
    "dnssecoffload":<String_value>,
    "nsec":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnszone":{
    "zonename":<String_value>,
    "proxymode":<String_value>,
    "dnssecoffload":<String_value>,
    "nsec":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"dnszone":{
    "zonename":<String_value>,
    "proxymode":true,
    "dnssecoffload":true,
    "nsec":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone/zonename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

sign

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone?**action=sign**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnszone":{
    "zonename":<String_value>,
    "keyname":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unsign

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone?**action=unsign**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dnszone":{
    "zonename":<String_value>,
    "keyname":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/dnszone?**args=zonename:<String_value>,type:<String_value>**

Use this query-parameter to get dnszone resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/dnszone?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dnszone?**filter=property-name1:property-val1,property-name2:property-val2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of dnszone resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dnszone?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnszone?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnszone resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnszone": [ {
zonename:<String_value>,type:<String_value>      "proxymode":<String_value>,
      "flags":<Double_value>,
      "nsecbitarray":<String[]_value>,
      "dnssecoffload":<String_value>,
      "nsec":<String_value>,
      "keyname":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone/zonename_value<String>

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/dnszone/zonename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dnszone/zonename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnszone": [ {
zonename:<String_value>,type:<String_value>      "proxymode":<String_value>,
      "flags":<Double_value>,
      "nsecbitarray":<String[]_value>,
      "dnssecoffload":<String_value>,
      "nsec":<String_value>,
      "keyname":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "dnszone": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnszone_binding

May 20, 2016

Binding object which returns the resources bound to dnszone.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
zonename	<String>	Read-write	Name of the zone. Mutually exclusive with the type parameter. Minimum length = 1
dnszone_domain_binding	<dnszone_domain_binding[]>	Read-only	domain that can be bound to dnszone.
dnszone_dnskey_binding	<dnszone_dnskey_binding[]>	Read-only	dnskey that can be bound to dnszone.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone_binding/zonename_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "dnszone_binding": [ {
    "zonename":<String_value>,
    "dnszone_dnskey_binding":<dnszone_dnskey_binding[]_value>,
    "dnszone_domain_binding":<dnszone_domain_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnszone_dnskey_binding

May 20, 2016

Binding object showing the dnskey that can be bound to dnszone.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
zonename	<String>	Read-write	Name of the zone. Mutually exclusive with the type parameter. Minimum length = 1
keyname	<String[]>	Read-write	Name of the public/private DNS key pair with which to sign the zone. You can sign a zone with up to four keys. Minimum length = 1
siginceptiontime	<Double[]>	Read-only	The time when sign was done with this key. Minimum value = 1
signed	<Double>	Read-only	Integer which denote status of keys.
expires	<Double>	Read-only	Time period for which to consider the key valid, after the key is used to sign a zone.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone_dnskey_binding/zonename_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/dnszone_dnskey_binding/zonename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnszone_dnskey_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnszone_dnskey_binding/zonename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnszone_dnskey_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "dnszone_dnskey_binding": [ {
      "zonename":<String_value>,
      "keyname":<String[]_value>,
      "siginceptiontime":<Double[]_value>,
      "signed":<Double_value>,
      "expires":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone_dnskey_binding/zonename_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"dnszone_dnskey_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnszone_domain_binding

May 20, 2016

Binding object showing the domain that can be bound to dnszone.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
zonename	<String>	Read-write	Name of the zone. Mutually exclusive with the type parameter. Minimum length = 1
domain	<String>	Read-write	Domain name that belongs to the given zone.
nextrecs	<String[]>	Read-only	An array of record types associated with the nsec record. Possible values = A, NS, CNAME, SOA, MX, AAAA, SRV, RRSIG, NSEC, DNSKEY, PTR, TXT, NAPTR
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone_domain_binding/zonename_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/dnszone_domain_binding/zonename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of dnszone_domain_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/dnszone_domain_binding/zonename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dnszone_domain_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnszone_domain_binding": [ {
    "zonename":<String_value>,
    "domain":<String_value>,
    "nextrecs":<String[]_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dnszone_domain_binding/zonename_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"dnszone_domain_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP DoS Protection

May 20, 2016

HTTP DOS protection configuration.

Name	Description
dospolicy	Configuration for DoS policy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dospolicy

May 20, 2016

Configuration for DoS policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the HTTP DoS protection policy. Must begin with a letter, number, or the underscore character (_). Other characters allowed, after the first character, are the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), and colon (:) characters. Minimum length = 1
qdepth	<Double>	Read-write	Queue depth. The queue size (the number of outstanding service requests on the system) before DoS protection is activated on the service to which the DoS protection policy is bound. Minimum value = 21
cltdetectrate	<Double>	Read-write	Client detect rate. Integer representing the percentage of traffic to which the HTTP DoS policy is to be applied after the queue depth condition is satisfied. Minimum value = 0 Maximum value = 100
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/dospolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dospolicy":{
    "name":<String_value>,
    "qdepth":<Double_value>,
    "cltdetectrate":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/dospolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/dospolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dospolicy":{
    "name":<String_value>,
    "qdepth":<Double_value>,
    "cltdetectrate":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/dospolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"dospolicy":{
    "name":<String_value>,
    "cltdetectrate":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/dospolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dospolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/dospolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of dospolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/dospolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/dospolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the dospolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "dospolicy": [ {
    "name":<String_value>,
    "qdepth":<Double_value>,
    "cltdetectrate":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/dospolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/dospolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/dospolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dospolicy": [ {
    "name":<String_value>,
    "qdepth":<Double_value>,
    "cltdetectrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/dospolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dospolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Front-end-optimization

May 20, 2016

Front end optimization configuration. The system's feature to optimize Web content for performance.

Name	Description
feoaction	Configuration for Front end optimization action
feoglobal_binding	Binding object showing the resources that can be bound to feoglobal
feoglobal_feopolicy_binding	Binding object showing the feopolicy that can be bound to feoglobal
feoparameter	Configuration for FEO parameter
feopolicy	Configuration for Front end optimization policy
feopolicy_binding	Binding object showing the resources that can be bound to feopolicy
feopolicy_csvserver_binding	Binding object showing the csvserver that can be bound to feopolicy
feopolicy_feoglobal_binding	Binding object showing the feoglobal that can be bound to feopolicy
feopolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to feopolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

feoaction

May 20, 2016

Configuration for Front end optimization action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the front end optimization action. Minimum length = 1
pageextendcache	<Boolean>	Read-write	Extend the time period during which the browser can use the cached resource.
imgshrinktoattrib	<Boolean>	Read-write	Shrink image dimensions as per the height and width attributes specified in the <img> tag.
imggiftopng	<Boolean>	Read-write	Convert GIF image formats to PNG formats.
imgtowebp	<Boolean>	Read-write	Convert JPEG, GIF, PNG image formats to WEBP format.
imgtojpegxr	<Boolean>	Read-write	Convert JPEG, GIF, PNG image formats to JXR format.
imginline	<Boolean>	Read-write	Inline images whose size is less than 2KB.
cssimginline	<Boolean>	Read-write	Inline small images (less than 2KB) referred within CSS files as background-URLs.
jpgoptimize	<Boolean>	Read-write	Remove non-image data such as comments from JPEG images.
imglazyload	<Boolean>	Read-write	Download images, only when the user scrolls the page to view them.
cssminify	<Boolean>	Read-write	Remove comments and whitespaces from CSSs.
cssinline	<Boolean>	Read-write	Inline CSS files, whose size is less than 2KB, within the main page.
csscombine	<Boolean>	Read-write	Combine one or more CSS files into one file.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
jsminify	<Boolean>	Read-write	Remove comments and whitespaces from JavaScript.
jsinline	<Boolean>	Read-write	Convert linked JavaScript files (less than 2KB) to inline JavaScript files.
htmlminify	<Boolean>	Read-write	Remove comments and whitespaces from an HTML page.
cssmovetohead	<Boolean>	Read-write	Move any CSS file present within the body tag of an HTML page to the head tag.
jsmovetoend	<Boolean>	Read-write	Move any JavaScript present in the body tag to the end of the body tag.
domainsharding	<String>	Read-write	Domain name of the server.
dnsshards	<String[]>	Read-write	Set of domain names that replaces the parent domain.
clientsidemeasurements	<Boolean>	Read-write	Send AppFlow records about the web pages optimized by this action. The records provide FEO statistics, such as the number of HTTP requests that have been reduced for this page. You must enable the Appflow feature before enabling this parameter.
imgadddimensions	<Boolean>	Read-only	Add dimension attributes to images, if not specified within the <img> tag.
imgshrinkformobile	<Boolean>	Read-only	Serve smaller images for mobile users.
imgweaken	<Boolean>	Read-only	Reduce the image quality.
jpgprogressive	<Boolean>	Read-only	Convert JPEG image formats to progressive formats.
cssflattenimports	<Boolean>	Read-only	Replace CSS import statements with the file content.
jscombine	<Boolean>	Read-only	Combine one or more JavaScript files into one file.
htmlrmdefaultattribs	<Boolean>	Read-only	Remove default redundant attributes from an HTML file.
htmlrmattribquotes	<Boolean>	Read-only	Remove unnecessary quotes present within the HTML attributes.
htmltrimurls	<Boolean>	Read-only	Trim URLs.
hits	<Double>	Read-only	The number of times the action has been taken.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
undefhits	<Double>	Read-only	Total number of undefined policy hits.
builtin	<String[]>	Read-only	Flag to determine if front end optimization action is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/feoaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"feoaction":{
    "name":<String_value>,
    "pageextendcache":<Boolean_value>,
    "imgshrinktoattrib":<Boolean_value>,
    "imggiftopng":<Boolean_value>,
    "imgtowebp":<Boolean_value>,
    "imgtojpegxr":<Boolean_value>,
    "imginline":<Boolean_value>,
    "cssimginline":<Boolean_value>,
    "jpgoptimize":<Boolean_value>,
    "imglazyload":<Boolean_value>,
    "cssminify":<Boolean_value>,
    "cssinline":<Boolean_value>,
    "csscombine":<Boolean_value>,
    "convertimporttolink":<Boolean_value>,
    "jsminify":<Boolean_value>,
    "jsinline":<Boolean_value>,
    "htmlminify":<Boolean_value>,
    "cssmovetohead":<Boolean_value>,
    "jsmovetoend":<Boolean_value>,
    "domainsharding":<String_value>,
    "dnsshards":<String[]_value>,
    "clientsidemeasurements":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/feoaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"feoaction":{
    "name":<String_value>,
    "pageextendcache":<Boolean_value>,
    "imgshrinktoattrib":<Boolean_value>,
    "imggiftopng":<Boolean_value>,
    "imgtowebp":<Boolean_value>,
    "imgtojpegxr":<Boolean_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "imginline":<Boolean_value>,
    "cssimginline":<Boolean_value>,
    "jpgoptimize":<Boolean_value>,
    "imglazyload":<Boolean_value>,
    "cssminify":<Boolean_value>,
    "cssinline":<Boolean_value>,
    "csscombine":<Boolean_value>,
    "convertimporttolink":<Boolean_value>,
    "jsminify":<Boolean_value>,
    "jsinline":<Boolean_value>,
    "htmlminify":<Boolean_value>,
    "cssmovetohead":<Boolean_value>,
    "jsmovetoend":<Boolean_value>,
    "domainsharding":<String_value>,
    "dnsshards":<String[]_value>,
    "clientsidemeasurements":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/feoaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"feoaction":{
    "name":<String_value>,
    "pageextendcache":true,
    "imgshrinktoattrib":true,
    "imggiftopng":true,
    "imgtowebp":true,
    "imgtojpegxr":true,
    "imginline":true,
    "cssimginline":true,
    "jpgoptimize":true,
    "imglazyload":true,
    "cssminify":true,
    "cssinline":true,
    "csscombine":true,
    "convertimporttolink":true,
    "jsminify":true,
    "jsinline":true,
    "htmlminify":true,
    "cssmovetohead":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "jsmovetoend":true,
        "clientsidemeasurements":true,
        "domainsharding":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/feoaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/feoaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/feoaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/feoaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of feoaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/feoaction?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/feoaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the feoaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "feoaction": [ {
    "name":<String_value>,
    "pageextendcache":<Boolean_value>,
    "imgshrinktoattrib":<Boolean_value>,
    "imggiftopng":<Boolean_value>,
    "imgtowebp":<Boolean_value>,
    "imgtojpegxr":<Boolean_value>,
    "imgadddimensions":<Boolean_value>,
    "imgshrinkformobile":<Boolean_value>,
    "imgweaken":<Boolean_value>,
    "imginline":<Boolean_value>,
    "cssimginline":<Boolean_value>,
    "jpgoptimize":<Boolean_value>,
    "jpgprogressive":<Boolean_value>,
    "imglazyload":<Boolean_value>,
    "cssminify":<Boolean_value>,
    "cssinline":<Boolean_value>,
    "csscombine":<Boolean_value>,
    "cssflattenimports":<Boolean_value>,
    "convertimporttolink":<Boolean_value>,
    "jsminify":<Boolean_value>,
    "jsinline":<Boolean_value>,
    "jscombine":<Boolean_value>,
    "htmlminify":<Boolean_value>,
    "htmlrmdefaultattribs":<Boolean_value>,
    "htmlrmattribquotes":<Boolean_value>,
    "htmltrimurls":<Boolean_value>,
    "cssmovetohead":<Boolean_value>,
    "jsmovetoend":<Boolean_value>,
    "domainsharding":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "dnsshards":<String[]_value>,
    "clientsidemeasurements":<Boolean_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "builtin":<String[]_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/feoaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/feoaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/feoaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "feoaction": [ {
    "name":<String_value>,
    "pageextendcache":<Boolean_value>,
    "imgshrinktoattrib":<Boolean_value>,
    "imggiftopng":<Boolean_value>,
    "imgtowebp":<Boolean_value>,
    "imgtojpegxr":<Boolean_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "imgadddimensions":<Boolean_value>,
    "imgshrinkformobile":<Boolean_value>,
    "imgweaken":<Boolean_value>,
    "imginline":<Boolean_value>,
    "cssimginline":<Boolean_value>,
    "jpgoptimize":<Boolean_value>,
    "jpgprogressive":<Boolean_value>,
    "imglazyload":<Boolean_value>,
    "cssminify":<Boolean_value>,
    "cssinline":<Boolean_value>,
    "csscombine":<Boolean_value>,
    "cssflattenimports":<Boolean_value>,
    "convertimporttolink":<Boolean_value>,
    "jsminify":<Boolean_value>,
    "jsinline":<Boolean_value>,
    "jscombine":<Boolean_value>,
    "htmlminify":<Boolean_value>,
    "htmlrmdefaultattribs":<Boolean_value>,
    "htmlrmattribquotes":<Boolean_value>,
    "htmltrimurls":<Boolean_value>,
    "cssmovetohead":<Boolean_value>,
    "jsmovetoend":<Boolean_value>,
    "domainsharding":<String_value>,
    "dnsshards":<String[]_value>,
    "clientsidemeasurements":<Boolean_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/feoaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "feoaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

feoglobal_binding

May 20, 2016

Binding object which returns the resources bound to feoglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
feoglobal_feopolicy_binding	<feoglobal_feopolicy_binding[]>	Read-only	feopolicy that can be bound to feoglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/feoglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "feoglobal_binding": [ {
      "feoglobal_feopolicy_binding":<feoglobal_feopolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

feoglobal_feopolicy_binding

Jul 07, 2017

Binding object showing the feopolicy that can be bound to feoglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority assigned to the policy binding.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
gotoprioritiyexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	The name of the globally bound front end optimization policy.
type	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT, NONE
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/feoglobal_feopolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"feoglobal_feopolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "type":<String_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/feoglobal_feopolicy_binding

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/feoglobal_feopolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/feoglobal_feopolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of feoglobal_feopolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/feoglobal_feopolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the feoglobal_feopolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "feoglobal_feopolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "gotopriorityexpression":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "policyname":<String_value>,
    "type":<String_value>,
    "numpol":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/feoglobal_feopolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "feoglobal_feopolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

feoparameter

May 20, 2016

Configuration for FEO parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cachemaxage	<Double>	Read-write	Maximum period (in days), for cache extension. Default value: 30 Minimum value = 0 Maximum value = 360
jpegqualitypercent	<Double>	Read-write	The percentage value of a JPEG image quality to be reduced. Range: 0 - 100. Default value: 75 Minimum value = 0 Maximum value = 100
cssinlinethressize	<Double>	Read-write	Threshold value of the file size (in bytes) for converting external CSS files to inline CSS files. Default value: 1024 Minimum value = 1 Maximum value = 2048
jsinlinethressize	<Double>	Read-write	Threshold value of the file size (in bytes), for converting external JavaScript files to inline JavaScript files. Default value: 1024 Minimum value = 1 Maximum value = 2048
imginlinethressize	<Double>	Read-write	Maximum file size of an image (in bytes), for coverting linked images to inline images. Default value: 1024 Minimum value = 1 Maximum value = 2048
builtin	<String[]>	Read-only	Specify the builtin flags for - set feo param. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/feoparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"feoparameter":{
    "cachemaxage":<Double_value>,
    "jpegqualitypercent":<Double_value>,
    "cssinlinethressize":<Double_value>,
    "jsinlinethressize":<Double_value>,
    "imginlinethressize":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/feoparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"feoparameter":{
    "cachemaxage":true,
    "jpegqualitypercent":true,
    "cssinlinethressize":true,
    "jsinlinethressize":true,
    "imginlinethressize":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/feoparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "feoparameter": [ {
    "cachemaxage":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "jpegqualitypercent":<Double_value>,
    "cssinlinethressize":<Double_value>,
    "jsinlinethressize":<Double_value>,
    "imginlinethressize":<Double_value>,
    "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

feopolicy

May 20, 2016

Configuration for Front end optimization policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the front end optimization policy. Minimum length = 1
rule	<String>	Read-write	The rule associated with the front end optimization policy.
action	<String>	Read-write	The front end optimization action that has to be performed when the rule matches. Minimum length = 1
builtin	<String[]>	Read-only	Flag to determine if the front end optimization policy is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
hits	<Double>	Read-only	Total number of hits.
undefhits	<Double>	Read-only	Total number of undefined policy hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"feopolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"feopolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"feopolicy":{
    "name":<String_value>,
    "rule":true,
    "action":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/feopolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/feopolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of feopolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/feopolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/feopolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the feopolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "feopolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "builtin":<String[]_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/feopolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/feopolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "feopolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "builtin":<String[]_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "feopolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

feopolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to feopolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the front end optimization policy. Minimum length = 1
feopolicy_csvserver_binding	<feopolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to feopolicy.
feopolicy_lbvserver_binding	<feopolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to feopolicy.
feopolicy_feoglobal_binding	<feopolicy_feoglobal_binding[]>	Read-only	feoglobal that can be bound to feopolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "feopolicy_binding": [ {
    "name":<String_value>,
    "feopolicy_csvserver_binding":<feopolicy_csvserver_binding[]_value>,
    "feopolicy_lbvserver_binding":<feopolicy_lbvserver_binding[]_value>,
    "feopolicy_feoglobal_binding":<feopolicy_feoglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

feopolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to feopolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where the policy is bound to.
name	<String>	Read-write	The name of the front end optimization policy. Minimum length = 1
priority	<Double>	Read-only	Priority of the policy.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether a policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/feopolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of feopolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/feopolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the feopolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "feopolicy_csvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"feopolicy_csvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

feopolicy_feoglobal_binding

May 20, 2016

Binding object showing the feoglobal that can be bound to feopolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where the policy is bound to.
name	<String>	Read-write	The name of the front end optimization policy. Minimum length = 1
priority	<Double>	Read-only	Priority of the policy.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether a policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy_feoglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/feopolicy_feoglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of feopolicy_feoglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/feopolicy_feoglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the feopolicy_feoglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "feopolicy_feoglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy_feoglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"feopolicy_feoglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

feopolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to feopolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where the policy is bound to.
name	<String>	Read-write	The name of the front end optimization policy. Minimum length = 1
priority	<Double>	Read-only	Priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether a policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/feopolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of feopolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/feopolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the feopolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "feopolicy_lbvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/feopolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"feopolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Filter

May 20, 2016

Filter configuration. Protection from malicious attacks can be achieved for web sites at the content level with the system's content filtering feature. When this feature is enabled, the system inspects every incoming request with the user-configured rules based on any HTTP URL or headers.

Name	Description
filteraction	Configuration for filter action
filterglobal_binding	Binding object showing the resources that can be bound to filterglobal
filterglobal_filterpolicy_binding	Binding object showing the filterpolicy that can be bound to filterglobal
filterhtmlinjectionparameter	Configuration for HTML injection parameter
filterhtmlinjectionvariable	Configuration for HTML injection variable
filterpolicy	Configuration for filter policy
filterpolicy_binding	Binding object showing the resources that can be bound to filterpolicy
filterpolicy_crvserver_binding	Binding object showing the crvserver that can be bound to filterpolicy
filterpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to filterpolicy
filterpolicy_filterglobal_binding	Binding object showing the filterglobal that can be bound to filterpolicy
filterpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to filterpolicy
filterpostbodyinjection	Configuration for HTML Injection postbody
filterprebodyinjection	Configuration for HTML Injection prebody

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filteraction

May 20, 2016

Configuration for filter action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the filtering action. Must begin with a letter, number, or the underscore character (_). Other characters allowed, after the first character, are the hyphen (-), period (.) hash (#), space ( ), at sign (@), equals (=), and colon (:) characters. Choose a name that helps identify the type of action. The name of a filter action cannot be changed after it is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
qual	<String>	Read-write	Qualifier, which is the action to be performed. The qualifier cannot be changed after it is set. The available options function as follows: ADD - Adds the specified HTTP header. RESET - Terminates the connection, sending the appropriate termination notice to the user's browser. FORWARD - Redirects the request to the designated service. You must specify either a service name or a page, but not both. DROP - Silently deletes the request, without sending a response to the user's browser. CORRUPT - Modifies the designated HTTP header to prevent it from performing the function it was intended to perform, then sends the request/response to the server/browser. ERRORCODE. Returns the designated HTTP error code to the user's browser (for example, 404, the standard HTTP code for a non-existent Web page). Possible values = reset, add, corrupt, forward, errorcode, drop
servicename	<String>	Read-write	Service to which to forward HTTP requests. Required if the qualifier is FORWARD. Minimum length = 1
value	<String>	Read-write	String containing the header_name and header_value. If the qualifier is ADD, specify <header_name>:<header_value>. If the qualifier is CORRUPT, specify only the header_name. Minimum length = 1
respcode	<Double>	Read-write	Response code to be returned for HTTP requests (for use with the ERRORCODE qualifier). Minimum value = 1
page	<String>	Read-write	HTML page to return for HTTP requests (For use with the ERRORCODE qualifier). Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
builtin	String[]>	Read-only	Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/filteraction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"filteraction":{
    "name":<String_value>,
    "qual":<String_value>,
    "servicename":<String_value>,
    "value":<String_value>,
    "respcode":<Double_value>,
    "page":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/filteraction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/filteraction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filteraction":{
    "name":<String_value>,
    "servicename":<String_value>,
    "value":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "respcode":<Double_value>,
    "page":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/filteraction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filteraction":{
    "name":<String_value>,
    "page":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/filteraction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/filteraction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/filteraction?**filter=property-name1:property-val1,property-name2:property-val2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of filteraction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/filteraction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/filteraction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the filteraction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filteraction": [ {
      "name":<String_value>,
      "qual":<String_value>,
      "servicename":<String_value>,
      "value":<String_value>,
      "respcode":<Double_value>,
      "page":<String_value>,
      "isdefault":<Boolean_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/filteraction/name_value<String>

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/filteraction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/filteraction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filteraction": [ {
      "name":<String_value>,
      "qual":<String_value>,
      "servicename":<String_value>,
      "value":<String_value>,
      "respcode":<Double_value>,
      "page":<String_value>,
      "isdefault":<Boolean_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/filteraction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "filteraction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterglobal_binding

Jul 07, 2017

Binding object which returns the resources bound to filterglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
filterglobal_filterpolicy_binding	<filterglobal_filterpolicy_binding[]>	Read-only	filterpolicy that can be bound to filterglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/filterglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "filterglobal_binding": [ {
      "filterglobal_filterpolicy_binding":<filterglobal_filterpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterglobal_filterpolicy_binding

May 20, 2016

Binding object showing the filterpolicy that can be bound to filterglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
state	<String>	Read-write	State of the binding. Possible values = ENABLED, DISABLED
policyname	<String>	Read-write	The name of the filter policy.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/filterglobal_filterpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"filterglobal_filterpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/filterglobal_filterpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/filterglobal_filterpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/filterglobal_filterpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of filterglobal_filterpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/filterglobal_filterpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the filterglobal_filterpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterglobal_filterpolicy_binding": [ {
    "priority":<Double_value>,
    "state":<String_value>,
    "policyname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/filterglobal_filterpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "filterglobal_filterpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterhtmlinjectionparameter

May 20, 2016

Configuration for HTML injection parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
rate	<Double>	Read-write	For a rate of x, HTML injection is done for 1 out of x policy matches. Default value: 1 Minimum value = 1
frequency	<Double>	Read-write	For a frequency of x, HTML injection is done at least once per x milliseconds. Default value: 1 Minimum value = 1
strict	<String>	Read-write	Searching for <html> tag. If this parameter is enabled, HTML injection does not insert the prebody or postbody content unless the <html> tag is found. Default value: ENABLED Possible values = ENABLED, DISABLED
htmlsearchlen	<Double>	Read-write	Number of characters, in the HTTP body, in which to search for the <html> tag if strict mode is set. Default value: 1024 Minimum value = 1

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filterhtmlinjectionparameter":{
    "rate":<Double_value>,
    "frequency":<Double_value>,
    "strict":<String_value>,
    "htmlsearchlen":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"filterhtmlinjectionparameter":{
    "rate":true,
    "frequency":true,
    "strict":true,
    "htmlsearchlen":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterhtmlinjectionparameter": [ {
    "rate":<Double_value>,
    "frequency":<Double_value>,
    "strict":<String_value>,
    "htmlsearchlen":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterhtmlinjectionvariable

May 20, 2016

Configuration for HTML injection variable resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
variable	<String>	Read-write	Name for the HTML injection variable to be added. Minimum length = 1 Maximum length = 31
value	<String>	Read-write	Value to be assigned to the new variable. Minimum length = 1 Maximum length = 31
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
type	<String>	Read-only	Type of the HTML injection variable. Possible values = System internal, User defined
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filterhtmlinjectionvariable":{
    "variable":<String_value>,
    "value":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable/variable_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filterhtmlinjectionvariable":{
    "variable":<String_value>,
    "value":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filterhtmlinjectionvariable":{
    "variable":<String_value>,
    "value":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of filterhtmlinjectionvariable resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable?**pagesize=#no&pageno=#no**

Use this query-parameter to get the filterhtmlinjectionvariable resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterhtmlinjectionvariable": [ {
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "variable":<String_value>,
    "value":<String_value>,
    "builtin":<String[]_value>,
    "type":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable/variable_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable/variable_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable/variable_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterhtmlinjectionvariable": [ {
    "variable":<String_value>,
    "value":<String_value>,
    "builtin":<String[]_value>,
    "type":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/filterhtmlinjectionvariable?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "filterhtmlinjectionvariable": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterpolicy

May 20, 2016

Configuration for filter policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the filtering action. Must begin with a letter, number, or the underscore character (_). Other characters allowed, after the first character, are the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and colon (:) characters. Choose a name that helps identify the type of action. The name cannot be updated after the policy is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	NetScaler classic expression specifying the type of connections that match this policy. Minimum length = 1
reqaction	<String>	Read-write	Name of the action to be performed on requests that match the policy. Cannot be specified if the rule includes condition to be evaluated for responses. Minimum length = 1
resaction	<String>	Read-write	The action to be performed on the response. The string value can be a filter action created filter action or a built-in action. Minimum length = 1
hits	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filterpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>,
    "resaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filterpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>,
    "resaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of filterpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the filterpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "reqaction":<String_value>,
      "resaction":<String_value>,
      "hits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "reqaction":<String_value>,
      "resaction":<String_value>,
      "hits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "filterpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterpolicy_binding

May 20, 2016

Binding object which returns the resources bound to filterpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the filter policy to be displayed. If a name is not provided, information about all the filter policies is shown. Minimum length = 1
filterpolicy_csvserver_binding	<filterpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to filterpolicy.
filterpolicy_lbvserver_binding	<filterpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to filterpolicy.
filterpolicy_crvserver_binding	<filterpolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to filterpolicy.
filterpolicy_filterglobal_binding	<filterpolicy_filterglobal_binding[]>	Read-only	filterglobal that can be bound to filterpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "filterpolicy_binding": [ {
      "name":<String_value>,
      "filterpolicy_crvserver_binding":<filterpolicy_crvserver_binding[]_value>,
      "filterpolicy_lbvserver_binding":<filterpolicy_lbvserver_binding[]_value>,
      "filterpolicy_csvserver_binding":<filterpolicy_csvserver_binding[]_value>,
      "filterpolicy_filterglobal_binding":<filterpolicy_filterglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterpolicy_crvserver_binding

May 20, 2016

Binding object showing the crvserver that can be bound to filterpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the filter policy to be displayed. If a name is not provided, information about all the filter policies is shown. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_crvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_crvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of filterpolicy_crvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_crvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the filterpolicy_crvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterpolicy_crvserver_binding": [ {
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_crvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"filterpolicy_crvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterpolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to filterpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the filter policy to be displayed. If a name is not provided, information about all the filter policies is shown. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of filterpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the filterpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterpolicy_csvserver_binding": [ {
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"filterpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterpolicy_filterglobal_binding

May 20, 2016

Binding object showing the filterglobal that can be bound to filterpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the filter policy to be displayed. If a name is not provided, information about all the filter policies is shown. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_filterglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_filterglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of filterpolicy_filterglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_filterglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the filterpolicy_filterglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterpolicy_filterglobal_binding": [ {
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_filterglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"filterpolicy_filterglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterpolicy_lbvserver_binding

Jul 07, 2017

Binding object showing the lbvserver that can be bound to filterpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the filter policy to be displayed. If a name is not provided, information about all the filter policies is shown. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of filterpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the filterpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterpolicy_lbvserver_binding": [ {
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"filterpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterpostbodyinjection

May 20, 2016

Configuration for HTML Injection postbody resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
postbody	<String>	Read-write	Name of file whose contents are to be inserted after the response body. Minimum length = 1
systemiid	<String>	Read-only	The system IID of the current NetScaler system.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpostbodyinjection

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filterpostbodyinjection":{
    "postbody":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpostbodyinjection?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filterpostbodyinjection":{
    "postbody":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/filterpostbodyinjection

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterpostbodyinjection": [ {
      "postbody":<String_value>,
      "systemiid":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

filterprebodyinjection

May 20, 2016

Configuration for HTML Injection prebody resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
prebody	<String>	Read-write	Name of file whose contents are to be inserted before the response body. Minimum length = 1
systemiid	<String>	Read-only	The system IID of the current NetScaler system.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/filterprebodyinjection

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filterprebodyinjection":{
    "prebody":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/filterprebodyinjection?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"filterprebodyinjection":{
    "prebody":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/filterprebodyinjection

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "filterprebodyinjection": [ {
      "prebody":<String_value>,
      "systemiid":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

High Availability

May 20, 2016

High-availability commands.

Name	Description
hafailover	Configuration for failover
hafiles	Configuration for files
hanode	Configuration for node
hanode_binding	Binding object showing the resources that can be bound to hanode
hanode_ci_binding	Binding object showing the ci that can be bound to hanode
hanode_fis_binding	Binding object showing the fis that can be bound to hanode
hanode_partialfailureinterfaces_binding	Binding object showing the partialfailureinterfaces that can be bound to hanode
hanode_routemonitor6_binding	Binding object showing the routemonitor6 that can be bound to hanode
hanode_routemonitor_binding	Binding object showing the routemonitor that can be bound to hanode
hasync	Configuration for sync

© 1999-2017 Citrix Systems, Inc. All rights reserved.

hafailover

May 20, 2016

Configuration for failover resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
force	<Boolean>	Read-write	Force a failover without prompting for confirmation.

Operations (click to see Properties)

FORCE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Force

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/hafailover?**action=Force**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"hafailover":{
    "force":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

hafiles

May 20, 2016

Configuration for files resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
mode	<String[]>	Read-write	Specify one of the following modes of synchronization. * all - Synchronize files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, and Application Firewall XML objects. * bookmarks - Synchronize all Access Gateway bookmarks. * ssl - Synchronize all certificates, keys, and CRLs for the SSL feature. * htmlinjection. Synchronize all scripts configured for the HTML injection feature. * imports. Synchronize all XML objects (for example, WSDLs, schemas, error pages) configured for the application firewall. * misc - Synchronize all license files and the rc.conf file. * all_plus_misc - Synchronize files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, application firewall XML objects, licenses, and the rc.conf file. Possible values = all, bookmarks, ssl, htmlinjection, imports, misc, dns, krb, all_plus_misc, all_minus_misc

Operations (click to see Properties)

SYNC

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

sync

URL: http://<netscaler-ip-address>/nitro/v1/config/hafiles?**action=sync**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"hafiles":{
    "mode":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

hanode

May 20, 2016

Configuration for node resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Number that uniquely identifies the node. For self node, it will always be 0. Peer node values can range from 1-64. Minimum value = 1 Maximum value = 64
ipaddress	<String>	Read-write	The NSIP or NSIP6 address of the node to be added for an HA configuration. This setting is neither propagated nor synchronized. Minimum length = 1
inc	<String>	Read-write	This option is required if the HA nodes reside on different networks. When this mode is enabled, the following independent network entities and configurations are neither propagated nor synced to the other node: MIPs, SNIPs, VLANs, routes (except LLB routes), route monitors, RNAT rules (except any RNAT rule with a VIP as the NAT IP), and dynamic routing configurations. They are maintained independently on each node. Default value: DISABLED Possible values = ENABLED, DISABLED
hastatus	<String>	Read-write	The HA status of the node. The HA status STAYSECONDARY is used to force the secondary device stay as secondary independent of the state of the Primary device. For example, in an existing HA setup, the Primary node has to be upgraded and this process would take few seconds. During the upgradation, it is possible that the Primary node may suffer from a downtime for a few seconds. However, the Secondary should not take over as the Primary node. Thus, the Secondary node should remain as Secondary even if there is a failure in the Primary node. STAYPRIMARY configuration keeps the node in primary state in case if it is healthy, even if the peer node was the primary node initially. If the node with STAYPRIMARY setting (and no peer node) is added to a primary node (which has this node as the peer) then this node takes over as the new primary and the older node becomes secondary. ENABLED state means normal HA operation without any constraints/preferences. DISABLED state disables the normal HA operation of the node. Possible values = ENABLED, STAYSECONDARY, DISABLED, STAYPRIMARY
hasync	<String>	Read-write	Automatically maintain synchronization by duplicating the configuration of the primary node on the secondary node. This setting is not propagated. Automatic synchronization requires that this setting be enabled (the default) on the current secondary node. Synchronization uses TCP port 3010. Default value: ENABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
haprop	<String>	Read-write	Automatically propagate all commands from the primary to the secondary node, except the following: * All HA configuration related commands. For example, add ha node, set ha node, and bind ha node. * All Interface related commands. For example, set interface and unset interface. * All channels related commands. For example, add channel, set channel, and bind channel. The propagated command is executed on the secondary node before it is executed on the primary. If command propagation fails, or if command execution fails on the secondary, the primary node executes the command and logs an error. Command propagation uses port 3010. Note: After enabling propagation, run force synchronization on either node. Default value: ENABLED Possible values = ENABLED, DISABLED
hellointerval	<Double>	Read-write	Interval, in milliseconds, between heartbeat messages sent to the peer node. The heartbeat messages are UDP packets sent to port 3003 of the peer node. Default value: 200 Minimum value = 200 Maximum value = 1000
deadinterval	<Double>	Read-write	Number of seconds after which a peer node is marked DOWN if heartbeat messages are not received from the peer node. Default value: 3 Minimum value = 3 Maximum value = 60
failsafe	<String>	Read-write	Keep one node primary if both nodes fail the health check, so that a partially available node can back up data and handle traffic. This mode is set independently on each node. Default value: OFF Possible values = ON, OFF
maxflips	<Double>	Read-write	Max number of flips allowed before becoming sticky primary. Default value: 0
maxfliptime	<Double>	Read-write	Interval after which flipping of node states can again start. Default value: 0
syncvlan	<Double>	Read-write	Vlan on which HA related communication is sent. This include sync, propagation , connection mirroring , LB persistency config sync, persistent session sync and session state sync. However HA heartbeats can go all interfaces. Minimum value = 1 Maximum value = 4094
name	<String>	Read-only	Node Name.
flags	<Double>	Read-only	The flags for this entry.
state	<String>	Read-only	HA Master State. Possible values = Primary, Secondary, UNKNOWN

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
enaifaces	<String>	Read-only	Enabled interfaces.
disifaces	<String>	Read-only	Disabled interfaces.
hamonifaces	<String>	Read-only	HAMON ON interfaces.
pfifaces	<String>	Read-only	Interfaces causing Partial Failure.
ifaces	<String>	Read-only	Interfaces on which non-multicast is not seen.
network	<String>	Read-only	The network.
netmask	<String>	Read-only	The netmask.
ssl2	<String>	Read-only	SSL card status. Possible values = DOWN, UP, NOT PRESENT, UNKNOWN
masterstatetime	<Double>	Read-only	Time elapsed in current master state.
routemonitor	<String>	Read-only	The IP address (IPv4 or IPv6).
curflips	<Double>	Read-only	Keeps track of number of flips that have happened till now in current interval.
completedfliptime	<Double>	Read-only	To inform user whether flip time is elapsed or not.
routemonitorstate	<String>	Read-only	State for route monitor. Possible values = UP, DOWN
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"hanode":{
    "id":<Double_value>,
    "ipaddress":<String_value>,
    "inc":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode/id_value<Double>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"hanode":{
    "id":<Double_value>,
    "hastatus":<String_value>,
    "hasync":<String_value>,
    "haprop":<String_value>,
    "hellointerval":<Double_value>,
    "deadinterval":<Double_value>,
    "failsafe":<String_value>,
    "maxflips":<Double_value>,
    "maxfliptime":<Double_value>,
    "syncvlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode?**action=unset**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"hanode":{
    "id":<Double_value>,
    "hastatus":true,
    "hasync":true,
    "haprop":true,
    "hellointerval":true,
    "deadinterval":true,
    "failsafe":true,
    "maxflips":true,
    "maxfliptime":true,
    "syncvlan":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/hanode?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/hanode?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of hanode resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/hanode?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/hanode?**pagesize=#no&pageno=#no**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the hanode resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "hanode": [ {
    "id":<Double_value>,
    "name":<String_value>,
    "ipaddress":<String_value>,
    "flags":<Double_value>,
    "hastatus":<String_value>,
    "state":<String_value>,
    "hasync":<String_value>,
    "haprop":<String_value>,
    "enaifaces":<String_value>,
    "disifaces":<String_value>,
    "hamonifaces":<String_value>,
    "pfifaces":<String_value>,
    "ifaces":<String_value>,
    "network":<String_value>,
    "netmask":<String_value>,
    "inc":<String_value>,
    "ssl2":<String_value>,
    "hellointerval":<Double_value>,
    "deadinterval":<Double_value>,
    "masterstatetime":<Double_value>,
    "failsafe":<String_value>,
    "routemonitor":<String_value>,
    "maxflips":<Double_value>,
    "maxfliptime":<Double_value>,
    "curflips":<Double_value>,
    "completedfliptime":<Double_value>,
    "syncvlan":<Double_value>,
    "routemonitorstate":<String_value>
}]}
```

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode/id_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/hanode/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/hanode/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "hanode": [ {
    "id":<Double_value>,
    "name":<String_value>,
    "ipaddress":<String_value>,
    "flags":<Double_value>,
    "hastatus":<String_value>,
    "state":<String_value>,
    "hasync":<String_value>,
    "haprop":<String_value>,
    "enaifaces":<String_value>,
    "disifaces":<String_value>,
    "hamonifaces":<String_value>,
    "pfifaces":<String_value>,
    "ifaces":<String_value>,
    "network":<String_value>,
    "netmask":<String_value>,
    "inc":<String_value>,
    "ssl2":<String_value>,
    "hellointerval":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "deadinterval":<Double_value>,
    "masterstatetime":<Double_value>,
    "failsafe":<String_value>,
    "routemonitor":<String_value>,
    "maxflips":<Double_value>,
    "maxfliptime":<Double_value>,
    "curflips":<Double_value>,
    "completedfliptime":<Double_value>,
    "syncvlan":<Double_value>,
    "routemonitorstate":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "hanode": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

hanode_binding

May 20, 2016

Binding object which returns the resources bound to hanode.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	ID of the node whose HA settings you want to display. (The ID of the local node is always 0.). Minimum value = 0 Maximum value = 64
hanode_partialfailureinterfaces_binding	<hanode_partialfailureinterfaces_binding[]>	Read-only	partialfailureinterfaces that can be bound to hanode.
hanode_routemonitor6_binding	<hanode_routemonitor6_binding[]>	Read-only	routemonitor6 that can be bound to hanode.
hanode_ci_binding	<hanode_ci_binding[]>	Read-only	ci that can be bound to hanode.
hanode_routemonitor_binding	<hanode_routemonitor_binding[]>	Read-only	routemonitor that can be bound to hanode.
hanode_fis_binding	<hanode_fis_binding[]>	Read-only	fis that can be bound to hanode.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_binding/id_value<Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "hanode_binding": [ {
      "id":<Double_value>,
      "hanode_routemonitor_binding":<hanode_routemonitor_binding[]_value>,
      "hanode_partialfailureinterfaces_binding":<hanode_partialfailureinterfaces_binding[]_value>,
      "hanode_fis_binding":<hanode_fis_binding[]_value>,
      "hanode_routemonitor6_binding":<hanode_routemonitor6_binding[]_value>,
      "hanode_ci_binding":<hanode_ci_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

hanode_ci_binding

May 20, 2016

Binding object showing the ci that can be bound to hanode.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
enaifaces	<String>	Read-write	Enabled interfaces.
routemonitor	<String>	Read-write	A route that you want the NetScaler appliance to monitor in its internal routing table. You can specify an IPv4 address or network, or an IPv6 address or network prefix. If you specify an IPv4 network address or IPv6 network prefix, the appliance monitors any route that matches the network or prefix. Minimum length = 1
id	<Double>	Read-write	Number that uniquely identifies the local node. The ID of the local node is always 0. Minimum value = 0 Maximum value = 64
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_ci_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/hanode_ci_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of hanode_ci_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/hanode_ci_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the hanode_ci_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "hanode_ci_binding": [ {
    "enaifaces":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "routemonitor":<String_value>,
    "id":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_ci_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"hanode_ci_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

hanode_fis_binding

May 20, 2016

Binding object showing the fis that can be bound to hanode.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Node Name.
routemonitor	<String>	Read-write	A route that you want the NetScaler appliance to monitor in its internal routing table. You can specify an IPv4 address or network, or an IPv6 address or network prefix. If you specify an IPv4 network address or IPv6 network prefix, the appliance monitors any route that matches the network or prefix. Minimum length = 1
id	<Double>	Read-write	Number that uniquely identifies the local node. The ID of the local node is always 0. Minimum value = 0 Maximum value = 64
enaifaces	<String>	Read-only	Enabled interfaces.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_fis_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/hanode_fis_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of hanode_fis_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/hanode_fis_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the hanode_fis_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "hanode_fis_binding": [ {
      "name":<String_value>,
      "routemonitor":<String_value>,
      "id":<Double_value>,
      "enaifaces":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_fis_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"hanode_fis_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

hanode_partialfailureinterfaces_binding

May 20, 2016

Binding object showing the partialfailureinterfaces that can be bound to hanode.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
pfifaces	<String>	Read-write	Interfaces causing Partial Failure.
routemonitor	<String>	Read-write	A route that you want the NetScaler appliance to monitor in its internal routing table. You can specify an IPv4 address or network, or an IPv6 address or network prefix. If you specify an IPv4 network address or IPv6 network prefix, the appliance monitors any route that matches the network or prefix. Minimum length = 1
id	<Double>	Read-write	Number that uniquely identifies the local node. The ID of the local node is always 0. Minimum value = 0 Maximum value = 64
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_partialfailureinterfaces_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/hanode_partialfailureinterfaces_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of hanode_partialfailureinterfaces_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/hanode_partialfailureinterfaces_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the hanode_partialfailureinterfaces_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "hanode_partialfailureinterfaces_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "pfifaces":<String_value>,
    "routemonitor":<String_value>,
    "id":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_partialfailureinterfaces_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"hanode_partialfailureinterfaces_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

hanode_routemonitor6_binding

May 20, 2016

Binding object showing the routemonitor6 that can be bound to hanode.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
routemonitor	<String>	Read-write	The IP address (IPv4 or IPv6).
id	<Double>	Read-write	Number that uniquely identifies the local node. The ID of the local node is always 0. Minimum value = 0 Maximum value = 64
netmask	<String>	Read-write	The netmask.
routemonitorstate	<String>	Read-only	State for route monitor. Possible values = UP, DOWN
flags	<Double>	Read-only	The flags for this entry.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/hanode_routemonitor6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"hanode_routemonitor6_binding":{
    "id":<Double_value>,
    "routemonitor":<String_value>,
    "netmask":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of hanode_routemonitor6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the hanode_routemonitor6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "hanode_routemonitor6_binding": [ {
      "routemonitor":<String_value>,
      "id":<Double_value>,
      "netmask":<String_value>,
      "routemonitorstate":<String_value>,
      "flags":<Double_value>
}]}
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_routemonitor6_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"hanode_routemonitor6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

hanode_routemonitor_binding

May 20, 2016

Binding object showing the routemonitor that can be bound to hanode.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
routemonitor	<String>	Read-write	The IP address (IPv4 or IPv6).
id	<Double>	Read-write	Number that uniquely identifies the local node. The ID of the local node is always 0. Minimum value = 0 Maximum value = 64
netmask	<String>	Read-write	The netmask.
routemonitorstate	<String>	Read-only	State for route monitor. Possible values = UP, DOWN
flags	<Double>	Read-only	The flags for this entry.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/hanode_routemonitor_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"hanode_routemonitor_binding":{
    "id":<Double_value>,
    "routemonitor":<String_value>,
    "netmask":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of hanode_routemonitor_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the hanode_routemonitor_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "hanode_routemonitor_binding": [ {
      "routemonitor":<String_value>,
      "id":<Double_value>,
      "netmask":<String_value>,
      "routemonitorstate":<String_value>,
      "flags":<Double_value>
}]}
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/hanode_routemonitor_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"hanode_routemonitor_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

hasync

May 20, 2016

Configuration for sync resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
force	<Boolean>	Read-write	Force synchronization regardless of the state of HA propagation and HA synchronization on either node.
save	<String>	Read-write	After synchronization, automatically save the configuration in the secondary node configuration file (ns.conf) without prompting for confirmation. Possible values = YES, NO

Operations (click to see Properties)

FORCE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Force

URL: http://<netscaler-ip-address>/nitro/v1/config/hasync?**action=Force**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"hasync":{
    "force":<Boolean_value>,
    "save":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Global Server Load Balancing

May 20, 2016

Global Server Load Balancing (GSLB) configuration. GSLB feature ensures that client requests are directed to a best performing site available in a global enterprise and distributed Internet environment.

Name	Description
gslbconfig	Configuration for gslb config
gslbdomain	Configuration for GSLB domain
gslbdomain_binding	Binding object showing the resources that can be bound to gslbdomain
gslbdomain_gslbservice_binding	Binding object showing the gslbservice that can be bound to gslbdomain
gslbdomain_gslbvserver_binding	Binding object showing the gslbvserver that can be bound to gslbdomain
gslbdomain_lbmonitor_binding	Binding object showing the lbmonitor that can be bound to gslbdomain
gslbldnsentries	Configuration for LDNS entry
gslbldnsentry	Configuration for LDNS entry
gslbparameter	Configuration for GSLB parameter
gslbrunningconfig	Configuration for running GSLB configuration
gslbservice	Configuration for GSLB service
gslbservice_binding	Binding object showing the resources that can be bound to gslbservice
gslbservice_dnsview_binding	Binding object showing the dnsview that can be bound to gslbservice
gslbservice_lbmonitor_binding	Binding object showing the lbmonitor that can be bound to gslbservice
gslbsite	Configuration for GSLB site

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
gslbsite_binding	Binding object showing the resources that can be bound to gslbsite
gslbsite_gslbservice_binding	Binding object showing the gslbservice that can be bound to gslbsite
gslbsyncstatus	Configuration for sync status
gslbvserver	Configuration for Global Server Load Balancing Virtual Server
gslbvserver_binding	Binding object showing the resources that can be bound to gslbvserver
gslbvserver_domain_binding	Binding object showing the domain that can be bound to gslbvserver
gslbvserver_gslbservice_binding	Binding object showing the gslbservice that can be bound to gslbvserver
gslbvserver_spilloverpolicy_binding	Binding object showing the spilloverpolicy that can be bound to gslbvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbconfig

Jul 07, 2017

Configuration for gslb config resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
preview	<Boolean>	Read-write	Do not synchronize the GSLB sites, but display the commands that would be applied on the slave node upon synchronization. Mutually exclusive with the Save Configuration option.
debug	<Boolean>	Read-write	Generate verbose output when synchronizing the GSLB sites. The Debug option generates more verbose output than the sync gslb config command in which the option is not used, and is useful for analyzing synchronization issues.
forcesync	<String>	Read-write	Force synchronization of the specified site even if a dependent configuration on the remote site is preventing synchronization or if one or more GSLB entities on the remote site have the same name but are of a different type. You can specify either the name of the remote site that you want to synchronize with the local site, or you can specify All Sites in the configuration utility (the string all-sites in the CLI). If you specify All Sites, all the sites in the GSLB setup are synchronized with the site on the master node. Note: If you select the Force Sync option, the synchronization starts without displaying the commands that are going to be executed.
nowarn	<Boolean>	Read-write	Suppress the warning and the confirmation prompt that are displayed before site synchronization begins. This option can be used in automation scripts that must not be interrupted by a prompt.
saveconfig	<Boolean>	Read-write	Save the configuration on all the nodes participating in the synchronization process, automatically. The master saves its configuration immediately before synchronization begins. Slave nodes save their configurations after the process of synchronization is complete. A slave node saves its configuration only if the configuration difference was successfully applied to it. Mutually exclusive with the Preview option.
command	<String>	Read-write	Run the specified command on the master node and then on all the slave nodes. You cannot use this option with the force sync and preview options.

Operations (click to see Properties)

SYNC

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

sync

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbconfig?**action=sync**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbconfig":{
    "preview":<Boolean_value>,
    "debug":<Boolean_value>,
    "forcesync":<String_value>,
    "nowarn":<Boolean_value>,
    "saveconfig":<Boolean_value>,
    "command":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbdomain

May 20, 2016

Configuration for GSLB domain resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the Domain. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbdomain

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of gslbdomain resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbdomain resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbdomain": [ {
    "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbdomain/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbdomain": [ {
    "name":<String_value>,

}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbdomain?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "gslbdomain": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbdomain_binding

May 20, 2016

Binding object which returns the resources bound to gslbdomain.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the Domain. Minimum length = 1
gslbdomain_gslbvserver_binding	<gslbdomain_gslbvserver_binding[]>	Read-only	gslbvserver that can be bound to gslbdomain.
gslbdomain_gslbservice_binding	<gslbdomain_gslbservice_binding[]>	Read-only	gslbservice that can be bound to gslbdomain.
gslbdomain_lbmonitor_binding	<gslbdomain_lbmonitor_binding[]>	Read-only	lbmonitor that can be bound to gslbdomain.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "gslbdomain_binding": [ {
    "name":<String_value>,
    "gslbdomain_lbmonitor_binding":<gslbdomain_lbmonitor_binding[]_value>,
    "gslbdomain_gslbvserver_binding":<gslbdomain_gslbvserver_binding[]_value>,
    "gslbdomain_gslbservice_binding":<gslbdomain_gslbservice_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbdomain_gslbservice_binding

Jul 07, 2017

Binding object showing the gslbservice that can be bound to gslbdomain.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the Domain. Minimum length = 1
servicename	<String>	Read-write	The service name.
cnameentry	<String>	Read-only	The cname of the gslb service.
state	<String>	Read-only	The state of the vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
dynamicconfwt	<Double>	Read-only	dynamic weight.
weight	<Double>	Read-only	weight assigned. Minimum value = 1 Maximum value = 100
servicetype	<String>	Read-only	The type GSLB service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
cumulativeweight	<Double>	Read-only	cumlative weight.
gslbthreshold	<Integer>	Read-only	The threshold value of the service.
port	<Integer>	Read-only	Port Number. Minimum value = 1 Range 1 - 65535
vservername	<String>	Read-only	.
			GSLB server state.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, unknown, DISABLED
ipaddress	<String>	Read-only	The Ip address of the service.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_gslbservice_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_gslbservice_binding/name_value<String>?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of gslbdomain_gslbservice_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_gslbservice_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbdomain_gslbservice_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbdomain_gslbservice_binding": [ {
    "name":<String_value>,
    "servicename":<String_value>,
    "cnameentry":<String_value>,
    "state":<String_value>,
    "dynamicconfwt":<Double_value>,
    "weight":<Double_value>,
    "servicetype":<String_value>,
    "cumulativeweight":<Double_value>,
    "gslbthreshold":<Integer_value>,
    "port":<Integer_value>,
    "vservername":<String_value>,
    "svreffgslbstate":<String_value>,
    "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_gslbservice_binding/name_value<String>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"gslbdomain_gslbservice_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbdomain_gslbvserver_binding

Jul 07, 2017

Binding object showing the gslbvserver that can be bound to gslbdomain.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the Domain. Minimum length = 1
vservername	<String>	Read-write	.
sitename	<String>	Read-only	Name of the site to which the service belongs.
backuplbmethod	<String>	Read-only	Indicates the backup method in case the primary fails. Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, SOURCEIPHASH, LEASTBANDWIDTH, LEASTPACKETS, STATICPROXIMITY, RTT, CUSTOMLOAD
customheaders	<String>	Read-only	The string that is sent to the service. Applicable to HTTP ,HTTP-ECV and RTSP monitor types.
persistenceid	<Double>	Read-only	Persistence id of the gslb vserver.
state	<String>	Read-only	The state of the vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
lbmethod	<String>	Read-only	The load balancing method set for the virtual server. Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, SOURCEIPHASH, LEASTBANDWIDTH, LEASTPACKETS, STATICPROXIMITY, RTT, CUSTOMLOAD
edr	<String>	Read-only	Send clients an empty DNS response when the GSLB virtual server is DOWN. Default value: DISABLED Possible values = ENABLED, DISABLED
cip	<String>	Read-only	Indicates if Client IP option is enabled. Possible values = ENABLED, DISABLED
statechangetimesec	<String>	Read-only	Time since last state change.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
netmask	<String>	Read-only	Minimum length = 1
dnsrecordtype	<String>	Read-only	The IP type for this GSLB vserver. Possible values = A, AAAA, CNAME, NAPTR
persistmask	<String>	Read-only	The optional IPv4 network mask applied to IPv4 addresses to establish source IP address based persistence. Minimum length = 1
sitepersistence	<String>	Read-only	Indicates the type of cookie persistence set. Possible values = ConnectionProxy, HTTPRedirect, NONE
servicetype	<String>	Read-only	The type GSLB service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
v6persistmasklen	<Double>	Read-only	Number of bits to consider in an IPv6 source IP address when creating source IP address based persistence sessions. Default value: 128 Minimum value = 1 Maximum value = 128
v6netmasklen	<Double>	Read-only	Number of bits to consider, in an IPv6 source IP address, for creating the hash that is required by the SOURCEIPHASH load balancing method. Default value: 128 Minimum value = 1 Maximum value = 128
persistencetype	<String>	Read-only	Indicates if persistence is set on the gslb vserver. Possible values = SOURCEIP, NONE
siteprefix	<String>	Read-only	The site prefix string.
dynamicweight	<String>	Read-only	Dynamic weight method of the vserver. Default value: DISABLED Possible values = SERVICECOUNT, SERVICEWEIGHT, DISABLED
mir	<String>	Read-only	. Default value: DISABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_gslbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_gslbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbdomain_gslbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_gslbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbdomain_gslbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbdomain_gslbvserver_binding": [ {
    "name":<String_value>,
    "vservername":<String_value>,
    "sitename":<String_value>,
    "backuplbmethod":<String_value>,
    "customheaders":<String_value>,
    "persistenceid":<Double_value>,
    "state":<String_value>,
    "lbmethod":<String_value>,
    "edr":<String_value>,
    "cip":<String_value>,
    "statechangetimesec":<String_value>,
    "netmask":<String_value>,
    "dnsrecordtype":<String_value>,
    "persistmask":<String_value>,
    "sitepersistence":<String_value>,
    "servicetype":<String_value>,
    "v6persistmasklen":<Double_value>,
    "v6netmasklen":<Double_value>,
    "persistencetype":<String_value>,
    "siteprefix":<String_value>,
    "dynamicweight":<String_value>,
    "mir":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_gslbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"gslbdomain_gslbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbdomain_lbmonitor_binding

Jul 07, 2017

Binding object showing the lbmonitor that can be bound to gslbdomain.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the Domain. Minimum length = 1
monitorname	<String>	Read-write	Monitor name.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
customheaders	<String>	Read-only	The string that is sent to the service. Applicable to HTTP ,HTTP-ECV and RTSP monitor types.
iptunnel	<String>	Read-only	The state of the monitor for tunneled devices. Possible values = YES, NO
responsetime	<Double>	Read-only	Response time of this monitor.
monstate	<String>	Read-only	Monitor state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
lastresponse	<String>	Read-only	The string form of monstatcode.
servicename	<String>	Read-only	The service name.
monitortotalprobes	<Double>	Read-only	Total monitor probes.
respcode	<String>	Read-only	The response codes.
vservername	<String>	Read-only	.
httprequest	<String>	Read-only	HTTP request to the backend server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
monitortotalfailedprobes	<Double>	Read-only	Total probes failed.
monitorcurrentfailedprobes	<Double>	Read-only	Total number of current failed probes.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_lbmonitor_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_lbmonitor_binding/name_value<String>?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of gslbdomain_lbmonitor_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_lbmonitor_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbdomain_lbmonitor_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbdomain_lbmonitor_binding": [ {
    "name":<String_value>,
    "monitorname":<String_value>,
    "monstatcode":<Integer_value>,
    "customheaders":<String_value>,
    "iptunnel":<String_value>,
    "responsetime":<Double_value>,
    "monstate":<String_value>,
    "lastresponse":<String_value>,
    "servicename":<String_value>,
    "monitortotalprobes":<Double_value>,
    "respcode":<String_value>,
    "vservername":<String_value>,
    "httprequest":<String_value>,
    "monitortotalfailedprobes":<Double_value>,
    "monitorcurrentfailedprobes":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbdomain_lbmonitor_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"gslbdomain_lbmonitor_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbldnsentries

May 20, 2016

Configuration for LDNS entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sitename	<String>	Read-only	The GSLB site name.
numsites	<Double>	Read-only	Specifies the number of gslb sites.
ipaddress	<String>	Read-only	IP address of the LDNS server.
ttl	<Double>	Read-only	TTL value of the LDNS entry.
name	<String>	Read-only	Monitor that is currently being used to monitor the LDNS ip..
rtt	<Double[]>	Read-only	RTT value of the LDNS entry for all gslb sites.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

CLEAR | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbldnsentries?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbldnsentries":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbldnsentries

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/gslbldnsentries?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbldnsentries?**filter=property-name1:property-val1,property-name2:property-val2**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of gslbldnsentries resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/gslbldnsentries?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbldnsentries?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbldnsentries resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbldnsentries": [ {
      "sitename":<String_value>,
      "numsites":<Double_value>,
      "ipaddress":<String_value>,
      "ttl":<Double_value>,
      "name":<String_value>,
      "rtt":<Double[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbldnsentries?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "gslbldnsentries": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbldnsentry

May 20, 2016

Configuration for LDNS entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipaddress	<String>	Read-write	IP address of the LDNS server. Minimum length = 1

Operations (click to see Properties)

DELETE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

delete

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbldnsentry

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbparameter

May 20, 2016

Configuration for GSLB parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ldnsentrytimeout	<Double>	Read-write	Time, in seconds, after which an inactive LDNS entry is removed. Default value: 180 Minimum value = 30 Maximum value = 65534
rtttolerance	<Double>	Read-write	Tolerance, in milliseconds, for newly learned round-trip time (RTT) values. If the difference between the old RTT value and the newly computed RTT value is less than or equal to the specified tolerance value, the LDNS entry in the network metric table is not updated with the new RTT value. Prevents the exchange of metrics when variations in RTT values are negligible. Default value: 5 Minimum value = 1 Maximum value = 100
ldnsmask	<String>	Read-write	The IPv4 network mask with which to create LDNS entries. Minimum length = 1
v6ldnsmasklen	<Double>	Read-write	Mask for creating LDNS entries for IPv6 source addresses. The mask is defined as the number of leading bits to consider, in the source IP address, when creating an LDNS entry. Default value: 128 Minimum value = 1 Maximum value = 128
ldnsprobeorder	<String[]>	Read-write	Order in which monitors should be initiated to calculate RTT. Possible values = PING, DNS, TCP
dropldnsreq	<String>	Read-write	Drop LDNS requests if round-trip time (RTT) information is not available. Default value: DISABLED Possible values = ENABLED, DISABLED
flags	<Double>	Read-only	State of the GSLB parameter.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbparameter":{
    "ldnsentrytimeout":<Double_value>,
    "rtttolerance":<Double_value>,
    "ldnsmask":<String_value>,
    "v6ldnsmasklen":<Double_value>,
    "ldnsprobeorder":<String[]_value>,
    "dropldnsreq":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbparameter":{
    "ldnsentrytimeout":true,
    "rtttolerance":true,
    "ldnsmask":true,
    "v6ldnsmasklen":true,
    "ldnsprobeorder":true,
    "dropldnsreq":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "gslbparameter": [ {
      "flags":<Double_value>,
      "ldnsentrytimeout":<Double_value>,
      "rtttolerance":<Double_value>,
      "ldnsmask":<String_value>,
      "v6ldnsmasklen":<Double_value>,
      "ldnsprobeorder":<String[]_value>,
      "dropldnsreq":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbrunningconfig

May 20, 2016

Configuration for running GSLB configuration resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
response	<String>	Read-only	gslb sync status as text blob.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbrunningconfig

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbrunningconfig": [ {
      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbservice

Jul 07, 2017

Configuration for GSLB service resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	Name for the GSLB service. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the GSLB service is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my gslbsvc" or 'my gslbsvc'). Minimum length = 1
cnameentry	<String>	Read-write	Canonical name of the GSLB service. Used in CNAME-based GSLB. Minimum length = 1
ip	<String>	Read-write	IP address for the GSLB service. Should represent a load balancing, content switching, or VPN virtual server on the NetScaler appliance, or the IP address of another load balancing device. Minimum length = 1
servername	<String>	Read-write	Name of the server hosting the GSLB service. Minimum length = 1
servicetype	<String>	Read-write	Type of service to create. Default value: NSSVC_SERVICE_UNKNOWN Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
port	<Integer>	Read-write	Port on which the load balancing entity represented by this GSLB service listens. Minimum value = 1 Range 1 - 65535
publicip	<String>	Read-write	The public IP address that a NAT device translates to the GSLB service's private IP address. Optional.
publicport	<Integer>	Read-write	The public port associated with the GSLB service's public IP address. The port is mapped to the service's private port number. Applicable to the local GSLB service. Optional.
			The maximum number of open connections that the service can support at any given time. A GSLB service whose connection count reaches the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			connection count drops below the maximum. Minimum value = 0 Maximum value = 4294967294
healthmonitor	<String>	Read-write	Monitor the health of the GSLB service. Default value: YES Possible values = YES, NO
sitename	<String>	Read-write	Name of the GSLB site to which the service belongs. Minimum length = 1
state	<String>	Read-write	Enable or disable the service. Default value: ENABLED Possible values = ENABLED, DISABLED
cip	<String>	Read-write	In the request that is forwarded to the GSLB service, insert a header that stores the client's IP address. Client IP header insertion is used in connection-proxy based site persistence. Default value: DISABLED Possible values = ENABLED, DISABLED
cipheader	<String>	Read-write	Name for the HTTP header that stores the client's IP address. Used with the Client IP option. If client IP header insertion is enabled on the service and a name is not specified for the header, the NetScaler appliance uses the name specified by the cipHeader parameter in the set ns param command or, in the GUI, the Client IP Header parameter in the Configure HTTP Parameters dialog box. Minimum length = 1
sitepersistence	<String>	Read-write	Use cookie-based site persistence. Applicable only to HTTP and SSL GSLB services. Possible values = ConnectionProxy, HTTPRedirect, NONE
cookietimeout	<Double>	Read-write	Timeout value, in minutes, for the cookie, when cookie based site persistence is enabled. Minimum value = 0 Maximum value = 1440
siteprefix	<String>	Read-write	The site's prefix string. When the service is bound to a GSLB virtual server, a GSLB site domain is generated internally for each bound service-domain pair by concatenating the site prefix of the service and the name of the domain. If the special string NONE is specified, the site-prefix string is unset. When implementing HTTP redirect site persistence, the NetScaler appliance redirects GSLB requests to GSLB services by using their site domains.
clttimeout	<Double>	Read-write	Idle time, in seconds, after which a client connection is terminated. Applicable if connection proxy based site persistence is used. Minimum value = 0 Maximum value = 31536000
svrtimeout	<Double>	Read-write	Idle time, in seconds, after which a server connection is terminated. Applicable if connection proxy based site persistence is used. Minimum value = 0 Maximum value = 31536000

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
bandwidth	<Double>	Read-write	Integer specifying the maximum bandwidth allowed for the service. A GSLB service whose bandwidth reaches the maximum is not considered when a GSLB decision is made, until its bandwidth consumption drops below the maximum.
downstateflush	<String>	Read-write	Flush all active transactions associated with the GSLB service when its state transitions from UP to DOWN. Do not enable this option for services that must complete their transactions. Applicable if connection proxy based site persistence is used. Possible values = ENABLED, DISABLED
maxaaausers	<Double>	Read-write	Maximum number of SSL VPN users that can be logged on concurrently to the VPN virtual server that is represented by this GSLB service. A GSLB service whose user count reaches the maximum is not considered when a GSLB decision is made, until the count drops below the maximum. Minimum value = 0 Maximum value = 65535
monthreshold	<Double>	Read-write	Monitoring threshold value for the GSLB service. If the sum of the weights of the monitors that are bound to this GSLB service and are in the UP state is not equal to or greater than this threshold value, the service is marked as DOWN. Minimum value = 0 Maximum value = 65535
hashid	<Double>	Read-write	Unique hash identifier for the GSLB service, used by hash based load balancing methods. Minimum value = 1
comment	<String>	Read-write	Any comments that you might want to associate with the GSLB service.
appflowlog	<String>	Read-write	Enable logging appflow flow information. Default value: ENABLED Possible values = ENABLED, DISABLED
naptrreplacement	<String>	Read-write	The replacement domain name for this NAPTR. Maximum length = 255
naptrorder	<Double>	Read-write	An integer specifying the order in which the NAPTR records MUST be processed in order to accurately represent the ordered list of Rules. The ordering is from lowest to highest. Default value: 1 Minimum value = 1 Maximum value = 65535
naptrservices	<String>	Read-write	Service Parameters applicable to this delegation path. Maximum length = 255
naptrdomainttl	<Double>	Read-write	Modify the TTL of the internally created naptr domain. Default value: 3600 Minimum value = 1
			An integer specifying the preference of this NAPTR among NAPTR records having same order. lower the number, higher the preference.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 1 Maximum value = 65535
ipaddress	<String>	Read-write	The new IP address of the service.
viewname	<String>	Read-write	Name of the DNS view of the service. A DNS view is used in global server load balancing (GSLB) to return a predetermined IP address to a specific group of clients, which are identified by using a DNS policy. Minimum length = 1
viewip	<String>	Read-write	IP address to be used for the given view.
weight	<Double>	Read-write	Weight to assign to the monitor-service binding. A larger number specifies a greater weight. Contributes to the monitoring threshold, which determines the state of the service. Minimum value = 1 Maximum value = 100
monitor_name_svc	<String>	Read-write	Name of the monitor to bind to the service. Minimum length = 1
newname	<String>	Read-write	New name for the GSLB service. Minimum length = 1
gslb	<String>	Read-only	. Default value: GSLB Possible values = REMOTE, LOCAL
svrstate	<String>	Read-only	Server state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
svreffgslbstate	<String>	Read-only	Effective state of the gslb svc. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
gslbthreshold	<Integer>	Read-only	Indicates if gslb svc has reached threshold.
gslbsvcstats	<Integer>	Read-only	Indicates if gslb svc has stats of the primary or the whole chain.
monstate	<String>	Read-only	State of the monitor bound to gslb service. Possible values = ENABLED, DISABLED
preferredlocation	<String>	Read-only	Prefered location.
			The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
ticksincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
threshold	<String>	Read-only	. Possible values = ABOVE, BELOW
clmonowner	<Double>	Read-only	Tells the mon owner of the gslb service. Minimum value = 0 Maximum value = 32
clmonview	<Double>	Read-only	Tells the view id of the monitoring owner.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbservice":{
    "servicename":<String_value>,
    "cnameentry":<String_value>,
    "ip":<String_value>,
    "servername":<String_value>,
    "servicetype":<String_value>,
    "port":<Integer_value>,
    "publicip":<String_value>,
    "publicport":<Integer_value>,
    "maxclient":<Double_value>,
    "healthmonitor":<String_value>,
    "sitename":<String_value>,
    "state":<String_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "sitepersistence":<String_value>,
    "cookietimeout":<Double_value>,
    "siteprefix":<String_value>,
    "clttimeout":<Double_value>,
    "svrtimeout":<Double_value>,
    "maxbandwidth":<Double_value>,
    "downstateflush":<String_value>,
    "maxaaausers":<Double_value>,
    "monthreshold":<Double_value>,
    "hashid":<Double_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "naptrreplacement":<String_value>,
    "naptrorder":<Double_value>,
    "naptrservices":<String_value>,
    "naptrdomainttl":<Double_value>,
    "naptrpreference":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice/servicename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbservice":{
    "servicename":<String_value>,
    "ipaddress":<String_value>,
    "publicip":<String_value>,
    "publicport":<Integer_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "sitepersistence":<String_value>,
    "siteprefix":<String_value>,
    "maxclient":<Double_value>,
    "healthmonitor":<String_value>,
    "maxbandwidth":<Double_value>,
    "downstateflush":<String_value>,
    "maxaaausers":<Double_value>,
    "viewname":<String_value>,
    "viewip":<String_value>,
    "monthreshold":<Double_value>,
    "weight":<Double_value>,
    "monitor_name_svc":<String_value>,
    "hashid":<Double_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "naptrorder":<Double_value>,
    "naptrpreference":<Double_value>,
    "naptrservices":<String_value>,
    "naptrreplacement":<String_value>,
    "naptrdomainttl":<Double_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbservice":{
    "servicename":<String_value>,
    "publicip":true,
    "publicport":true,
    "cip":true,
    "cipheader":true,
    "sitepersistence":true,
    "siteprefix":true,
    "maxclient":true,
    "healthmonitor":true,
    "maxbandwidth":true,
    "downstateflush":true,
    "maxaaausers":true,
    "monthreshold":true,
    "hashid":true,
    "comment":true,
    "appflowlog":true,
    "naptrorder":true,
    "naptrpreference":true,
    "naptrservices":true,
    "naptrreplacement":true,
    "naptrdomainttl":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbservice":{
    "servicename":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/gslbservice?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbservice?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of gslbservice resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/gslbservice?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbservice?**pagesize=#no&pageno=#no**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the gslbservice resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbservice": [ {
    "servicename":<String_value>,
    "gslb":<String_value>,
    "ipaddress":<String_value>,
    "ip":<String_value>,
    "servername":<String_value>,
    "servicetype":<String_value>,
    "port":<Integer_value>,
    "publicip":<String_value>,
    "publicport":<Integer_value>,
    "maxclient":<Double_value>,
    "maxaaausers":<Double_value>,
    "sitename":<String_value>,
    "svrstate":<String_value>,
    "svreffgslbstate":<String_value>,
    "gslbthreshold":<Integer_value>,
    "gslbsvcstats":<Integer_value>,
    "state":<String_value>,
    "monstate":<String_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "sitepersistence":<String_value>,
    "siteprefix":<String_value>,
    "clttimeout":<Double_value>,
    "svrtimeout":<Double_value>,
    "preferredlocation":<String_value>,
    "maxbandwidth":<Double_value>,
    "downstateflush":<String_value>,
    "cnameentry":<String_value>,
    "viewname":<String_value>,
    "viewip":<String_value>,
    "weight":<Double_value>,
    "monthreshold":<Double_value>,
    "monitor_state":<String_value>,
    "hashid":<Double_value>,
    "comment":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "healthmonitor":<String_value>,
    "appflowlog":<String_value>,
    "statechangetimesec":<String_value>,
    "tickssincelaststatechange":<Double_value>,
    "threshold":<String_value>,
    "clmonowner":<Double_value>,
    "clmonview":<Double_value>,
    "naptrorder":<Double_value>,
    "naptrpreference":<Double_value>,
    "naptrservices":<String_value>,
    "naptrreplacement":<String_value>,
    "naptrdomainttl":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice/servicename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/gslbservice/servicename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/gslbservice/servicename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "gslbservice": [ {
    "servicename":<String_value>,
    "gslb":<String_value>,
    "ipaddress":<String_value>,
    "ip":<String_value>,
    "servername":<String_value>,
    "servicetype":<String_value>,
    "port":<Integer_value>,
    "publicip":<String_value>,
    "publicport":<Integer_value>,
    "maxclient":<Double_value>,
    "maxaaausers":<Double_value>,
    "sitename":<String_value>,
    "svrstate":<String_value>,
    "svreffgslbstate":<String_value>,
    "gslbthreshold":<Integer_value>,
    "gslbsvcstats":<Integer_value>,
    "state":<String_value>,
    "monstate":<String_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "sitepersistence":<String_value>,
    "siteprefix":<String_value>,
    "clttimeout":<Double_value>,
    "svrtimeout":<Double_value>,
    "preferredlocation":<String_value>,
    "maxbandwidth":<Double_value>,
    "downstateflush":<String_value>,
    "cnameentry":<String_value>,
    "viewname":<String_value>,
    "viewip":<String_value>,
    "weight":<Double_value>,
    "monthreshold":<Double_value>,
    "monitor_state":<String_value>,
    "hashid":<Double_value>,
    "comment":<String_value>,
    "healthmonitor":<String_value>,
    "appflowlog":<String_value>,
    "statechangetimesec":<String_value>,
    "tickssincelaststatechange":<Double_value>,
    "threshold":<String_value>,
    "clmonowner":<Double_value>,
    "clmonview":<Double_value>,
    "naptrorder":<Double_value>,
    "naptrpreference":<Double_value>,
    "naptrservices":<String_value>,
    "naptrreplacement":<String_value>,
    "naptrdomainttl":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbservice": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbservice_binding

May 20, 2016

Binding object which returns the resources bound to gslbservice.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	Name of the GSLB service. Minimum length = 1
gslbservice_lbmonitor_binding	<gslbservice_lbmonitor_binding[]>	Read-only	lbmonitor that can be bound to gslbservice.
gslbservice_dnsview_binding	<gslbservice_dnsview_binding[]>	Read-only	dnsview that can be bound to gslbservice.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice_binding/servicename_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "gslbservice_binding": [ {
      "servicename":<String_value>,
      "gslbservice_lbmonitor_binding":<gslbservice_lbmonitor_binding[]_value>,
      "gslbservice_dnsview_binding":<gslbservice_dnsview_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbservice_dnsview_binding

May 20, 2016

Binding object showing the dnsview that can be bound to gslbservice.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
viewname	<String>	Read-write	Name of the DNS view of the service. A DNS view is used in global server load balancing (GSLB) to return a predetermined IP address to a specific group of clients, which are identified by using a DNS policy. Minimum length = 1
servicename	<String>	Read-write	Name of the GSLB service. Minimum length = 1
viewip	<String>	Read-write	IP address to be used for the given view.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/gslbservice_dnsview_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"gslbservice_dnsview_binding":{
    "servicename":<String_value>,
    "viewname":<String_value>,
    "viewip":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbservice_dnsview_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbservice_dnsview_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbservice_dnsview_binding": [ {
    "viewname":<String_value>,
    "servicename":<String_value>,
    "viewip":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice_dnsview_binding/servicename_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"gslbservice_dnsview_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbsite

May 20, 2016

Configuration for GSLB site resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sitename	<String>	Read-write	Name for the GSLB site. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my gslbsite" or 'my gslbsite'). Minimum length = 1
sitetype	<String>	Read-write	Type of site to create. If the type is not specified, the appliance automatically detects and sets the type on the basis of the IP address being assigned to the site. If the specified site IP address is owned by the appliance (for example, a MIP address or SNIP address), the site is a local site. Otherwise, it is a remote site. Default value: NONE Possible values = REMOTE, LOCAL
siteipaddress	<String>	Read-write	IP address for the GSLB site. The GSLB site uses this IP address to communicate with other GSLB sites. For a local site, use any IP address that is owned by the appliance (for example, a SNIP or MIP address, or the IP address of the ADNS service). Minimum length = 1
publicip	<String>	Read-write	Public IP address for the local site. Required only if the appliance is deployed in a private address space and the site has a public IP address hosted on an external firewall or a NAT device. Minimum length = 1
metricexchange	<String>	Read-write	Exchange metrics with other sites. Metrics are exchanged by using Metric Exchange Protocol (MEP). The appliances in the GSLB setup exchange health information once every second. If you disable metrics exchange, you can use only static load balancing methods (such as round robin, static proximity, or the hash-based methods), and if you disable metrics exchange when a dynamic load balancing method (such as least connection) is in operation, the appliance falls back to round robin. Also, if you disable metrics exchange, you must use a monitor to determine the state of GSLB services. Otherwise, the service is marked as DOWN.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = ENABLED, DISABLED
nwmetricexchange	<String>	Read-write	Exchange, with other GSLB sites, network metrics such as round-trip time (RTT), learned from communications with various local DNS (LDNS) servers used by clients. RTT information is used in the dynamic RTT load balancing method, and is exchanged every 5 seconds. Default value: ENABLED Possible values = ENABLED, DISABLED
sessionexchange	<String>	Read-write	Exchange persistent session entries with other GSLB sites every five seconds. Default value: ENABLED Possible values = ENABLED, DISABLED
triggermonitor	<String>	Read-write	Specify the conditions under which the GSLB service must be monitored by a monitor, if one is bound. Available settings function as follows: * ALWAYS - Monitor the GSLB service at all times. * MEPDOWN - Monitor the GSLB service only when the exchange of metrics through the Metrics Exchange Protocol (MEP) is disabled. MEPDOWN_SVCDOWN - Monitor the service in either of the following situations: * The exchange of metrics through MEP is disabled. * The exchange of metrics through MEP is enabled but the status of the service, learned through metrics exchange, is DOWN. Default value: ALWAYS Possible values = ALWAYS, MEPDOWN, MEPDOWN_SVCDOWN
parentsite	<String>	Read-write	Parent site of the GSLB site, in a parent-child topology.
clip	<String>	Read-write	Cluster IP address. Specify this parameter to connect to the remote cluster site for GSLB auto-sync. Note: The cluster IP address is defined when creating the cluster.
publicclip	<String>	Read-write	IP address to be used to globally access the remote cluster when it is deployed behind a NAT. It can be same as the normal cluster IP address.
naptrreplacementsuffix	<String>	Read-write	The naptr replacement suffix configured here will be used to construct the naptr replacement field in NAPTR record. Minimum length = 1
status	<String>	Read-only	Current metric exchange status. Possible values = ACTIVE, INACTIVE, DOWN
persistencemepstatus	<String>	Read-only	Network metric and persistence exchange MEP connection status. Possible values = ACTIVE, INACTIVE, DOWN
version	<Double>	Read-only	will be true if the remote site's version is ncore compatible with the local site.(>= 9.2).
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsite

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbsite":{
    "sitename":<String_value>,
    "sitetype":<String_value>,
    "siteipaddress":<String_value>,
    "publicip":<String_value>,
    "metricexchange":<String_value>,
    "nwmetricexchange":<String_value>,
    "sessionexchange":<String_value>,
    "triggermonitor":<String_value>,
    "parentsite":<String_value>,
    "clip":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "publicclip":<String_value>,
    "naptrreplacementsuffix":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsite/sitename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsite

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbsite":{
    "sitename":<String_value>,
    "metricexchange":<String_value>,
    "nwmetricexchange":<String_value>,
    "sessionexchange":<String_value>,
    "triggermonitor":<String_value>,
    "naptrreplacementsuffix":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsite?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbsite":{
    "sitename":<String_value>,
    "metricexchange":true,
    "nwmetricexchange":true,
    "sessionexchange":true,
    "triggermonitor":true,
    "naptrreplacementsuffix":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsite

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/gslbsite?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbsite?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of gslbsite resources configured on NetScaler.Filtering can be

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/gslbsite?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbsite?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbsite resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbsite": [ {
    "sitename":<String_value>,
    "sitetype":<String_value>,
    "siteipaddress":<String_value>,
    "publicip":<String_value>,
    "metricexchange":<String_value>,
    "status":<String_value>,
    "persistencemepstatus":<String_value>,
    "nwmetricexchange":<String_value>,
    "sessionexchange":<String_value>,
    "triggermonitor":<String_value>,
    "parentsite":<String_value>,
    "version":<Double_value>,
    "clip":<String_value>,
    "publicclip":<String_value>,
    "naptrreplacementsuffix":<String_value>
}]}
```

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsite/sitename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/gslbsite/sitename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/gslbsite/sitename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbsite": [ {
    "sitename":<String_value>,
    "sitetype":<String_value>,
    "siteipaddress":<String_value>,
    "publicip":<String_value>,
    "metricexchange":<String_value>,
    "status":<String_value>,
    "persistencemepstatus":<String_value>,
    "nwmetricexchange":<String_value>,
    "sessionexchange":<String_value>,
    "triggermonitor":<String_value>,
    "parentsite":<String_value>,
    "version":<Double_value>,
    "clip":<String_value>,
    "publicclip":<String_value>,
    "naptrreplacementsuffix":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsite?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "gslbsite": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbservice_lbmonitor_binding

Jul 07, 2017

Binding object showing the lbmonitor that can be bound to gslbservice.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
weight	<Double>	Read-write	Weight to assign to the monitor-service binding. A larger number specifies a greater weight. Contributes to the monitoring threshold, which determines the state of the service. Minimum value = 1 Maximum value = 100
servicename	<String>	Read-write	Name of the GSLB service. Minimum length = 1
monitor_name	<String>	Read-write	Monitor name.
monstate	<String>	Read-write	State of the monitor bound to gslb service. Possible values = ENABLED, DISABLED
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
responsetime	<Double>	Read-only	Response time of this monitor.
totalfailedprobes	<Double>	Read-only	The total number of failed probs.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
failedprobes	<Double>	Read-only	Number of the current failed monitoring probes.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
monitor_state	<String>	Read-only	The running state of the monitor on this service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monitortotalprobes	<Double>	Read-only	Total number of probes sent to monitor this service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monitortotalfailedprobes	<Double>	Read-only	Total number of failed probes.
monitorcurrentfailedprobes	<Double>	Read-only	Total number of currently failed probes.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/gslbservice_lbmonitor_binding

HTTP Method: PUT

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"gslbservice_lbmonitor_binding":{
    "servicename":<String_value>,
    "monitor_name":<String_value>,
    "monstate":<String_value>,
    "weight":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbservice_lbmonitor_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbservice_lbmonitor_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbservice_lbmonitor_binding": [ {
    "weight":<Double_value>,
    "servicename":<String_value>,
    "monitor_name":<String_value>,
    "monstate":<String_value>,
    "monstatcode":<Integer_value>,
    "responsetime":<Double_value>,
    "totalfailedprobes":<Double_value>,
    "monstatparam2":<Integer_value>,
    "failedprobes":<Double_value>,
    "monstatparam3":<Integer_value>,
    "monitor_state":<String_value>,
    "monitortotalprobes":<Double_value>,
    "monstatparam1":<Integer_value>,
    "monitortotalfailedprobes":<Double_value>,
    "monitorcurrentfailedprobes":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbservice_lbmonitor_binding/servicename_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"gslbservice_lbmonitor_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbsite_binding

May 20, 2016

Binding object which returns the resources bound to gslbsite.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sitename	<String>	Read-write	Name of the GSLB site. If you specify a site name, details of all the site's constituent services are also displayed. Minimum length = 1
gslbsite_gslbservice_binding	<gslbsite_gslbservice_binding[]>	Read-only	gslbservice that can be bound to gslbsite.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsite_binding/sitename_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "gslbsite_binding": [ {
    "sitename":<String_value>,
    "gslbsite_gslbservice_binding":<gslbsite_gslbservice_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbsite_gslbservice_binding

May 20, 2016

Binding object showing the gslbservice that can be bound to gslbsite.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	Service name.
sitename	<String>	Read-write	Name of the GSLB site. If you specify a site name, details of all the site's constituent services are also displayed. Minimum length = 1
cnameentry	<String>	Read-only	The cname of the gslb service. Minimum length = 1
state	<String>	Read-only	State of the gslb service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
servicetype	<String>	Read-only	Service type. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP, SYSLOGUDP, FIX
port	<Integer>	Read-only	Port number of the gslb service. Range 1 - 65535
ipaddress	<String>	Read-only	IP Address of the gslb service.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsite_gslbservice_binding/sitename_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbsite_gslbservice_binding/sitename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbsite_gslbservice_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbsite_gslbservice_binding/sitename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbsite_gslbservice_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbsite_gslbservice_binding": [ {
    "servicename":<String_value>,
    "sitename":<String_value>,
    "cnameentry":<String_value>,
    "state":<String_value>,
    "servicetype":<String_value>,
    "port":<Integer_value>,
    "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsite_gslbservice_binding/sitename_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"gslbsite_gslbservice_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbsyncstatus

Jul 07, 2017

Configuration for sync status resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
response	<String>	Read-only	gslb sync status as text blob.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbsyncstatus

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbsyncstatus": [ {
      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbvserver

May 20, 2016

Configuration for Global Server Load Balancing Virtual Server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the GSLB virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). Minimum length = 1
servicetype	<String>	Read-write	Protocol used by services bound to the virtual server. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
iptype	<String>	Read-write	The IP type for this GSLB vserver. Default value: IPV4 Possible values = IPV4, IPV6
dnsrecordtype	<String>	Read-write	DNS record type to associate with the GSLB virtual server's domain name. Default value: A Possible values = A, AAAA, CNAME, NAPTR
lbmethod	<String>	Read-write	Load balancing method for the GSLB virtual server. Default value: LEASTCONNECTION Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, SOURCEIPHASH, LEASTBANDWIDTH, LEASTPACKETS, STATICPROXIMITY, RTT, CUSTOMLOAD
backupsessiontimeout	<Double>	Read-write	A non zero value enables the feature whose minimum value is 2 minutes. The feature can be disabled by setting the value to zero. The created session is in effect for a specific client per domain. Minimum value = 0 Maximum value = 1440
backuplbmethod	<String>	Read-write	Backup load balancing method. Becomes operational if the primary load balancing method fails or cannot be used. Valid only if the primary method is based on either round-trip time (RTT) or static proximity. Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, SOURCEIPHASH, LEASTBANDWIDTH, LEASTPACKETS, STATICPROXIMITY,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
netmask	<String>	Read-write	IPv4 network mask for use in the SOURCEIPHASH load balancing method. Minimum length = 1
v6netmasklen	<Double>	Read-write	Number of bits to consider, in an IPv6 source IP address, for creating the hash that is required by the SOURCEIPHASH load balancing method. Default value: 128 Minimum value = 1 Maximum value = 128
tolerance	<Double>	Read-write	Site selection tolerance, in milliseconds, for implementing the RTT load balancing method. If a site's RTT deviates from the lowest RTT by more than the specified tolerance, the site is not considered when the NetScaler appliance makes a GSLB decision. The appliance implements the round robin method of global server load balancing between sites whose RTT values are within the specified tolerance. If the tolerance is 0 (zero), the appliance always sends clients the IP address of the site with the lowest RTT. Minimum value = 0 Maximum value = 100
persistencetype	<String>	Read-write	Use source IP address based persistence for the virtual server. After the load balancing method selects a service for the first packet, the IP address received in response to the DNS query is used for subsequent requests from the same client. Possible values = SOURCEIP, NONE
persistenceid	<Double>	Read-write	The persistence ID for the GSLB virtual server. The ID is a positive integer that enables GSLB sites to identify the GSLB virtual server, and is required if source IP address based or spill over based persistence is enabled on the virtual server. Minimum value = 0 Maximum value = 65535
persistmask	<String>	Read-write	The optional IPv4 network mask applied to IPv4 addresses to establish source IP address based persistence. Minimum length = 1
v6persistmasklen	<Double>	Read-write	Number of bits to consider in an IPv6 source IP address when creating source IP address based persistence sessions. Default value: 128 Minimum value = 1 Maximum value = 128
timeout	<Double>	Read-write	Idle time, in minutes, after which a persistence entry is cleared. Default value: 2 Minimum value = 2 Maximum value = 1440
edr	<String>	Read-write	Send clients an empty DNS response when the GSLB virtual server is DOWN. Default value: DISABLED Possible values = ENABLED, DISABLED
			Include multiple IP addresses in the DNS responses sent to clients. Default value: DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
disableprimaryondown	<String>	Read-write	Configure to direct traffic to the backup chain even after the primary GSLB virtual server returns to the UP state. Used when spillover is configured for the virtual server. Default value: DISABLED Possible values = ENABLED, DISABLED
dynamicweight	<String>	Read-write	Specify if the appliance should consider the service count, service weights, or ignore both when using weight-based load balancing methods. The state of the number of services bound to the virtual server help the appliance to select the service. Default value: DISABLED Possible values = SERVICECOUNT, SERVICEWEIGHT, DISABLED
state	<String>	Read-write	State of the GSLB virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
considereffectivestate	<String>	Read-write	If the primary state of all bound GSLB services is DOWN, consider the effective states of all the GSLB services, obtained through the Metrics Exchange Protocol (MEP), when determining the state of the GSLB virtual server. To consider the effective state, set the parameter to STATE_ONLY. To disregard the effective state, set the parameter to NONE. The effective state of a GSLB service is the ability of the corresponding virtual server to serve traffic. The effective state of the load balancing virtual server, which is transferred to the GSLB service, is UP even if only one virtual server in the backup chain of virtual servers is in the UP state. Default value: NONE Possible values = NONE, STATE_ONLY
comment	<String>	Read-write	Any comments that you might want to associate with the GSLB virtual server.
somethod	<String>	Read-write	Type of threshold that, when exceeded, triggers spillover. Available settings function as follows: * CONNECTION - Spillover occurs when the number of client connections exceeds the threshold. * DYNAMICCONNECTION - Spillover occurs when the number of client connections at the GSLB virtual server exceeds the sum of the maximum client (Max Clients) settings for bound GSLB services. Do not specify a spillover threshold for this setting, because the threshold is implied by the Max Clients settings of the bound GSLB services. * BANDWIDTH - Spillover occurs when the bandwidth consumed by the GSLB virtual server's incoming and outgoing traffic exceeds the threshold. * HEALTH - Spillover occurs when the percentage of weights of the GSLB services that are UP drops below the threshold. For example, if services gslbSvc1, gslbSvc2, and gslbSvc3 are bound to a virtual server, with weights 1, 2, and 3, and the spillover threshold is 50%, spillover occurs if gslbSvc1 and gslbSvc3 or gslbSvc2 and gslbSvc3 transition to DOWN. * NONE - Spillover does not occur. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sopersistence	<String>	Read-write	primary and backup GSLB virtual servers. Default value: DISABLED Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-write	Timeout for spillover persistence, in minutes. Default value: 2 Minimum value = 2 Maximum value = 1440
sothreshold	<Double>	Read-write	Threshold at which spillover occurs. Specify an integer for the CONNECTION spillover method, a bandwidth value in kilobits per second for the BANDWIDTH method (do not enter the units), or a percentage for the HEALTH method (do not enter the percentage symbol). Minimum value = 1 Maximum value = 4294967287
sobackupaction	<String>	Read-write	Action to be performed if spillover is to take effect, but no backup chain to spillover is usable or exists. Possible values = DROP, ACCEPT, REDIRECT
appflowlog	<String>	Read-write	Enable logging appflow flow information. Default value: ENABLED Possible values = ENABLED, DISABLED
backupvserver	<String>	Read-write	Name of the backup GSLB virtual server to which the appliance should to forward requests if the status of the primary GSLB virtual server is down or exceeds its spillover threshold. Minimum length = 1
servicename	<String>	Read-write	Name of the GSLB service for which to change the weight. Minimum length = 1
weight	<Double>	Read-write	Weight to assign to the GSLB service. Minimum value = 1 Maximum value = 100
domainname	<String>	Read-write	Domain name for which to change the time to live (TTL) and/or backup service IP address. Minimum length = 1
ttl	<Double>	Read-write	Time to live (TTL) for the domain. Minimum value = 1
backupip	<String>	Read-write	The IP address of the backup service for the specified domain name. Used when all the services bound to the domain are down, or when the backup chain of virtual servers is down. Minimum length = 1
cookie_domain	<String>	Read-write	The cookie domain for the GSLB site. Used when inserting the GSLB site cookie in the HTTP response. Minimum length = 1
			Timeout, in minutes, for the GSLB site cookie.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum value = 1440
sitedomainttl	<Double>	Read-write	TTL, in seconds, for all internally created site domains (created when a site prefix is configured on a GSLB service) that are associated with this virtual server. Minimum value = 1
newname	<String>	Read-write	New name for the GSLB virtual server. Minimum length = 1
curstate	<String>	Read-only	State of the gslb vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
status	<Integer>	Read-only	Current status of the gslb vserver. During the initial phase if the configured lb method is not round robin , the vserver will adopt round robin to distribute traffic for a predefined number of requests.
lbrrreason	<Integer>	Read-only	Reason why a vserver is in RR. The following are the reasons: 1 - MEP is DOWN (GSLB) 2 - LB method has changed 3 - Bound service's state changed to UP 4 - A new service is bound 5 - Startup RR factor has changed 6 - LB feature is enabled 7 - Load monitor is not active on a service 8 - Vserver is Enabled 9 - SSL feature is Enabled 10 - All bound services have reached threshold. Using effective state to load balance (GSLB) 11 - Primary state of bound services are not UP. Using effective state to load balance (GSLB) 12 - No LB decision can be made as all bound services have either reached threshold or are not UP (GSLB) 13 - All load monitors are active .
iscname	<String>	Read-only	is cname feature set on vserver. Possible values = ENABLED, DISABLED
sitepersistence	<String>	Read-only	Type of Site Persistence set. Possible values = ConnectionProxy, HTTPRedirect, NONE
totalservices	<Double>	Read-only	Total number of services bound to the vserver.
activeservices	<Double>	Read-only	Total number of active services bound to the vserver.
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
statechangetimemsec	<Double>	Read-only	Time at which last state change happened. Milliseconds part.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ticksincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
health	<Double>	Read-only	Health of vserver based on percentage of weights of active svcs/all svcs. This does not consider administratively disabled svcs.
policyname	<String>	Read-only	Name of the policy bound to the GSLB vserver.
priority	<Double>	Read-only	Priority. Minimum value = 1 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE. o If gotoPriorityExpression is not present or if it is equal to END then the policy bank evaluation ends here o Else if the gotoPriorityExpression is equal to NEXT then the next policy in the priority order is evaluated. o Else gotoPriorityExpression is evaluated. The result of gotoPriorityExpression (which has to be a number) is processed as follows: - An UNDEF event is triggered if . gotoPriorityExpression cannot be evaluated . gotoPriorityExpression evaluates to number which is smaller than the maximum priority in the policy bank but is not same as any policy's priority . gotoPriorityExpression evaluates to a priority that is smaller than the current policy's priority - If the gotoPriorityExpression evaluates to the priority of the current policy then the next policy in the priority order is evaluated. - If the gotoPriorityExpression evaluates to the priority of a policy further ahead in the list then that policy will be evaluated next. This field is applicable only to rewrite and responder policies.
type	<String>	Read-only	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
vsvrbindsvcip	<String>	Read-only	used for showing the ip of bound entities.
vsvrbindsvcport	<Integer>	Read-only	used for showing ports of bound entities. Range 1 - 65535
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbvserver":{
    "name":<String_value>,
    "servicetype":<String_value>,
    "iptype":<String_value>,
    "dnsrecordtype":<String_value>,
    "lbmethod":<String_value>,
    "backupsessiontimeout":<Double_value>,
    "backuplbmethod":<String_value>,
    "netmask":<String_value>,
    "v6netmasklen":<Double_value>,
    "tolerance":<Double_value>,
    "persistencetype":<String_value>,
    "persistenceid":<Double_value>,
    "persistmask":<String_value>,
    "v6persistmasklen":<Double_value>,
    "timeout":<Double_value>,
    "edr":<String_value>,
    "mir":<String_value>,
    "disableprimaryondown":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "dynamicweight":<String_value>,
    "state":<String_value>,
    "considereffectivestate":<String_value>,
    "comment":<String_value>,
    "somethod":<String_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "sothreshold":<Double_value>,
    "sobackupaction":<String_value>,
    "appflowlog":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbvserver":{
    "name":<String_value>,
    "iptype":<String_value>,
    "dnsrecordtype":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "backupvserver":<String_value>,
    "backupsessiontimeout":<Double_value>,
    "lbmethod":<String_value>,
    "backuplbmethod":<String_value>,
    "netmask":<String_value>,
    "v6netmasklen":<Double_value>,
    "tolerance":<Double_value>,
    "persistencetype":<String_value>,
    "persistenceid":<Double_value>,
    "persistmask":<String_value>,
    "v6persistmasklen":<Double_value>,
    "timeout":<Double_value>,
    "edr":<String_value>,
    "mir":<String_value>,
    "disableprimaryondown":<String_value>,
    "dynamicweight":<String_value>,
    "considereffectivestate":<String_value>,
    "somethod":<String_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "sothreshold":<Double_value>,
    "sobackupaction":<String_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "domainname":<String_value>,
    "ttl":<Double_value>,
    "backupip":<String_value>,
    "cookie_domain":<String_value>,
    "cookietimeout":<Double_value>,
    "sitedomainttl":<Double_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbvserver":{
    "name":<String_value>,
    "backupvserver":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "sothreshold":true,
    "iptype":true,
    "dnsrecordtype":true,
    "backupsessiontimeout":true,
    "lbmethod":true,
    "backuplbmethod":true,
    "netmask":true,
    "v6netmasklen":true,
    "tolerance":true,
    "persistencetype":true,
    "persistenceid":true,
    "persistmask":true,
    "v6persistmasklen":true,
    "timeout":true,
    "edr":true,
    "mir":true,
    "disableprimaryondown":true,
    "dynamicweight":true,
    "considereffectivestate":true,
    "somethod":true,
    "sopersistence":true,
    "sopersistencetimeout":true,
    "sobackupaction":true,
    "servicename":true,
    "weight":true,
    "comment":true,
    "appflowlog":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"gslbvserver":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of gslbvserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbvserver resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbvserver": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "servicetype":<String_value>,
    "state":<String_value>,
    "iptype":<String_value>,
    "dnsrecordtype":<String_value>,
    "persistencetype":<String_value>,
    "persistenceid":<Double_value>,
    "lbmethod":<String_value>,
    "backuplbmethod":<String_value>,
    "tolerance":<Double_value>,
    "timeout":<Double_value>,
    "curstate":<String_value>,
    "netmask":<String_value>,
    "v6netmasklen":<Double_value>,
    "persistmask":<String_value>,
    "v6persistmasklen":<Double_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "domainname":<String_value>,
    "ttl":<Double_value>,
    "backupip":<String_value>,
    "cookie_domain":<String_value>,
    "cookietimeout":<Double_value>,
    "sitedomainttl":<Double_value>,
    "status":<Integer_value>,
    "lbrrreason":<Integer_value>,
    "backupvserver":<String_value>,
    "backupsessiontimeout":<Double_value>,
    "edr":<String_value>,
    "mir":<String_value>,
    "disableprimaryondown":<String_value>,
    "dynamicweight":<String_value>,
    "iscname":<String_value>,
    "sitepersistence":<String_value>,
    "considereffectivestate":<String_value>,
    "totalservices":<Double_value>,
    "activeservices":<Double_value>,
    "statechangetimesec":<String_value>,
    "statechangetimemsec":<Double_value>,
    "tickssincelaststatechange":<Double_value>,
    "comment":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "somethod":<String_value>,
    "sobackupaction":<String_value>,
    "sopersistence":<String_value>,
    "sothreshold":<Double_value>,
    "health":<Double_value>,
    "appflowlog":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "vsvrbindsvcip":<String_value>,
    "vsvrbindsvcport":<Integer_value>
}]}
```


get

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbvserver": [ {
    "name":<String_value>,
    "servicetype":<String_value>,
    "state":<String_value>,
    "iptype":<String_value>,
    "dnsrecordtype":<String_value>,
    "persistencetype":<String_value>,
    "persistenceid":<Double_value>,
    "lbmethod":<String_value>,
    "backuplbmethod":<String_value>,
    "tolerance":<Double_value>,
    "timeout":<Double_value>,
    "curstate":<String_value>,
    "netmask":<String_value>,
    "v6netmasklen":<Double_value>,
    "persistmask":<String_value>,
    "v6persistmasklen":<Double_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "domainname":<String_value>,
    "ttl":<Double_value>,
    "backupip":<String_value>,
    "cookie_domain":<String_value>,
    "cookietimeout":<Double_value>,
    "sitedomainttl":<Double_value>,
    "status":<Integer_value>,
    "lbrrreason":<Integer_value>,
    "backupvserver":<String_value>,
    "backupsessiontimeout":<Double_value>,
    "edr":<String_value>,
    "mir":<String_value>,
    "disableprimaryondown":<String_value>,
    "dynamicweight":<String_value>,
    "iscname":<String_value>,
    "sitepersistence":<String_value>,
    "considereffectivestate":<String_value>,
    "totalservices":<Double_value>,
    "activeservices":<Double_value>,
    "statechangetimesec":<String_value>,
    "statechangetimemsec":<Double_value>,
    "tickssincelaststatechange":<Double_value>,
    "comment":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "somethod":<String_value>,
    "sobackupaction":<String_value>,
    "sopersistence":<String_value>,
    "sothreshold":<Double_value>,
    "health":<Double_value>,
    "appflowlog":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "vsvrbindsvcip":<String_value>,
    "vsvrbindsvcport":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbvserver": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbvserver_binding

May 20, 2016

Binding object which returns the resources bound to gslbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the GSLB virtual server. Minimum length = 1
gslbvserver_spilloverpolicy_binding	<gslbvserver_spilloverpolicy_binding[]>	Read-only	spilloverpolicy that can be bound to gslbvserver.
gslbvserver_gslbservice_binding	<gslbvserver_gslbservice_binding[]>	Read-only	gslbservice that can be bound to gslbvserver.
gslbvserver_domain_binding	<gslbvserver_domain_binding[]>	Read-only	domain that can be bound to gslbvserver.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "gslbvserver_binding": [ {
    "name":<String_value>,
    "gslbvserver_gslbservice_binding":<gslbvserver_gslbservice_binding[]_value>,
    "gslbvserver_spilloverpolicy_binding":<gslbvserver_spilloverpolicy_binding[]_value>,
    "gslbvserver_domain_binding":<gslbvserver_domain_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbvserver_domain_binding

May 20, 2016

Binding object showing the domain that can be bound to gslbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
backupipflag	<Boolean>	Read-write	The IP address of the backup service for the specified domain name. Used when all the services bound to the domain are down, or when the backup chain of virtual servers is down.
cookietimeout	<Double>	Read-write	Timeout, in minutes, for the GSLB site cookie. Minimum value = 0 Maximum value = 1440
backupip	<String>	Read-write	The IP address of the backup service for the specified domain name. Used when all the services bound to the domain are down, or when the backup chain of virtual servers is down. Minimum length = 1
name	<String>	Read-write	Name of the virtual server on which to perform the binding operation. Minimum length = 1
ttl	<Double>	Read-write	Time to live (TTL) for the domain. Minimum value = 1
domainname	<String>	Read-write	Domain name for which to change the time to live (TTL) and/or backup service IP address. Minimum length = 1
sitedomainttl	<Double>	Read-write	TTL, in seconds, for all internally created site domains (created when a site prefix is configured on a GSLB service) that are associated with this virtual server. Minimum value = 1
cookie_domainflag	<Boolean>	Read-write	The cookie domain for the GSLB site. Used when inserting the GSLB site cookie in the HTTP response.
cookie_domain	<String>	Read-write	The cookie domain for the GSLB site. Used when inserting the GSLB site cookie in the HTTP response. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/gslbvserver_domain_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"gslbvserver_domain_binding":{
    "name":<String_value>,
    "domainname":<String_value>,
    "ttl":<Double_value>,
    "backupip":<String_value>,
    "cookie_domain":<String_value>,
    "cookietimeout":<Double_value>,
    "sitedomainttl":<Double_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbvserver_domain_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbvserver_domain_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbvserver_domain_binding": [ {
    "backupipflag":<Boolean_value>,
    "cookietimeout":<Double_value>,
    "backupip":<String_value>,
    "name":<String_value>,
    "ttl":<Double_value>,
    "domainname":<String_value>,
    "sitedomainttl":<Double_value>,
    "cookie_domainflag":<Boolean_value>,
    "cookie_domain":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_domain_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"gslbvserver_domain_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbvserver_gslbservice_binding

May 20, 2016

Binding object showing the gslbservice that can be bound to gslbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
weight	<Double>	Read-write	Weight to assign to the GSLB service. Minimum value = 1 Maximum value = 100
name	<String>	Read-write	Name of the virtual server on which to perform the binding operation. Minimum length = 1
servicename	<String>	Read-write	Name of the GSLB service for which to change the weight. Minimum length = 1
domainname	<String>	Read-write	Domain name for which to change the time to live (TTL) and/or backup service IP address. Minimum length = 1
cnameentry	<String>	Read-only	The cname of the gslb service.
svcsitepersistence	<String>	Read-only	Type of Site Persistence set on the bound service. Possible values = ConnectionProxy, HTTPRedirect, NONE
gslbboundsvctype	<String>	Read-only	Protocol used by services bound to the GSLBvirtual server. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, ANY, SIP_UDP, SIP_TCP, SIP_SSL, RADIUS, RDP, RTSP, MYSQL, MSSQL, ORACLE
preferredlocation	<String>	Read-only	The target site to be returned in the DNS response when a policy is successfully evaluated against the incoming DNS request. Target site is specified in dotted notation with up to 6 qualifiers. Wildcard `*' is accepted as a valid qualifier token.
dynamicconfwt	<Double>	Read-only	Weight obtained by the virtue of bound service count or weight.
cumulativeweight	<Double>	Read-only	Cumulative weight is the weight of GSLB service considering both its configured weight and dynamic weight. It is equal to product of dynamic weight and configured weight of the gslb service .
gslbthreshold	<Integer>	Read-only	Indicates if gslb svc has reached threshold.
			This field is introduced for displaying the cookie in cluster setup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
port	<Integer>	Read-only	Port number. Range 1 - 65535
iscname	<String>	Read-only	is cname feature set on vserver. Possible values = ENABLED, DISABLED
curstate	<String>	Read-only	State of the gslb vserver. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
svreffgslbstate	<String>	Read-only	Effective state of the gslb svc. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
thresholdvalue	<Integer>	Read-only	Tells whether threshold exceeded for this service participating in CUSTOMLB.
ipaddress	<String>	Read-only	IP address.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/gslbvserver_gslbservice_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"gslbvserver_gslbservice_binding":{
    "name":<String_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "domainname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbvserver_gslbservice_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbvserver_gslbservice_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbvserver_gslbservice_binding": [ {
    "weight":<Double_value>,
    "name":<String_value>,
    "servicename":<String_value>,
    "domainname":<String_value>,
    "cnameentry":<String_value>,
    "svcsitepersistence":<String_value>,
    "gslbboundsvctype":<String_value>,
    "preferredlocation":<String_value>,
    "dynamicconfwt":<Double_value>,
    "cumulativeweight":<Double_value>,
    "gslbthreshold":<Integer_value>,
    "sitepersistcookie":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "port":<Integer_value>,
    "iscname":<String_value>,
    "curstate":<String_value>,
    "svreffgslbstate":<String_value>,
    "thresholdvalue":<Integer_value>,
    "ipaddress":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_gslbservice_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"gslbvserver_gslbservice_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbvserver_spilloverpolicy_binding

May 20, 2016

Binding object showing the spilloverpolicy that can be bound to gslbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority. Minimum value = 1 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE. o If gotoPriorityExpression is not present or if it is equal to END then the policy bank evaluation ends here o Else if the gotoPriorityExpression is equal to NEXT then the next policy in the priority order is evaluated. o Else gotoPriorityExpression is evaluated. The result of gotoPriorityExpression (which has to be a number) is processed as follows: - An UNDEF event is triggered if . gotoPriorityExpression cannot be evaluated . gotoPriorityExpression evaluates to number which is smaller than the maximum priority in the policy bank but is not same as any policy's priority . gotoPriorityExpression evaluates to a priority that is smaller than the current policy's priority - If the gotoPriorityExpression evaluates to the priority of the current policy then the next policy in the priority order is evaluated. - If the gotoPriorityExpression evaluates to the priority of a policy further ahead in the list then that policy will be evaluated next. This field is applicable only to rewrite and responder policies.
policyname	<String>	Read-write	Name of the policy bound to the GSLB vserver.
name	<String>	Read-write	Name of the virtual server on which to perform the binding operation. Minimum length = 1
type	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/gslbvserver_spilloverpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"gslbvserver_spilloverpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of gslbvserver_spilloverpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the gslbvserver_spilloverpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbvserver_spilloverpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "type":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/gslbvserver_spilloverpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"gslbvserver_spilloverpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Ica

May 20, 2016

ICA configuration.

Name	Description
icaaccessprofile	Configuration for ica accessprofile
icaaction	Configuration for ica action
icaglobal_binding	Binding object showing the resources that can be bound to icaglobal
icaglobal_icapolicy_binding	Binding object showing the icapolicy that can be bound to icaglobal
icapolicy	Configuration for ICA policy
icapolicy_binding	Binding object showing the resources that can be bound to icapolicy
icapolicy_crvserver_binding	Binding object showing the crvserver that can be bound to icapolicy
icapolicy_icaglobal_binding	Binding object showing the icaglobal that can be bound to icapolicy
icapolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to icapolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

icaaccessprofile

May 20, 2016

Configuration for ica accessprofile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the ICA accessprofile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the ICA accessprofile is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my ica accessprofile" or 'my ica accessprofile'). Each of the features can be configured as DEFAULT/DISABLED. Here, DISABLED means that the policy settings on the backend XenApp/XenDesktop server are overridden and the NetScaler makes the decision to deny access. Whereas DEFAULT means that the NetScaler allows the request to reach the XenApp/XenDesktop that takes the decision to allow/deny access based on the policy configured on it. For example, if ClientAudioRedirection is enabled on the backend XenApp/XenDesktop server, and the configured profile has ClientAudioRedirection as DISABLED, the NetScaler makes the decision to deny the request irrespective of the configuration on the backend. If the configured profile has ClientAudioRedirection as DEFAULT, then the NetScaler forwards the requests to the backend XenApp/XenDesktop server.It then makes the decision to allow/deny access based on the policy configured on it. Minimum length = 1
connectclientlptports	<String>	Read-write	Allow Default access/Disable automatic connection of LPT ports from the client when the user logs on. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientaudioredirection	<String>	Read-write	Allow Default access/Disable applications hosted on the server to play sounds through a sound device installed on the client computer, also allows or prevents users to record audio input. Default value: DISABLED Possible values = DEFAULT, DISABLED
localremotedatasharing	<String>	Read-write	Allow Default access/Disable file/data sharing via the Reciever for HTML5. Default value: DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
clientclipboardredirection	<String>	Read-write	Allow Default access/Disable the clipboard on the client device to be mapped to the clipboard on the server. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientcomportredirection	<String>	Read-write	Allow Default access/Disable COM port redirection to and from the client. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientdriveredirection	<String>	Read-write	Allow Default access/Disables drive redirection to and from the client. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientprinterredirection	<String>	Read-write	Allow Default access/Disable client printers to be mapped to a server when a user logs on to a session. Default value: DISABLED Possible values = DEFAULT, DISABLED
multistream	<String>	Read-write	Allow Default access/Disable the multistream feature for the specified users. Default value: DISABLED Possible values = DEFAULT, DISABLED
clientusbdriveredirection	<String>	Read-write	Allow Default access/Disable the redirection of USB devices to and from the client. Default value: DISABLED Possible values = DEFAULT, DISABLED
refcnt	<Double>	Read-only	Number of entities using this accessprofile.
builtin	<String[]>	Read-only	Indicates that the ICA accessprofile is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default accessprofile.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"icaaccessprofile":{
    "name":<String_value>,
    "connectclientlptports":<String_value>,
    "clientaudioredirection":<String_value>,
    "localremotedatasharing":<String_value>,
    "clientclipboardredirection":<String_value>,
    "clientcomportredirection":<String_value>,
    "clientdriveredirection":<String_value>,
    "clientprinterredirection":<String_value>,
    "multistream":<String_value>,
    "clientusbdriveredirection":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"icaaccessprofile":{
    "name":<String_value>,
    "connectclientlptports":<String_value>,
    "clientaudioredirection":<String_value>,
    "localremotedatasharing":<String_value>,
    "clientclipboardredirection":<String_value>,
    "clientcomportredirection":<String_value>,
    "clientdriveredirection":<String_value>,
    "clientprinterredirection":<String_value>,
    "multistream":<String_value>,
    "clientusbdriveredirection":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile?**action=unset**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"icaaccessprofile":{
    "name":<String_value>,
    "connectclientlptports":true,
    "clientaudioredirection":true,
    "localremotedatasharing":true,
    "clientclipboardredirection":true,
    "clientcomportredirection":true,
    "clientdriveredirection":true,
    "clientprinterredirection":true,
    "multistream":true,
    "clientusbdriveredirection":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of icaaccessprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the icaaccessprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icaaccessprofile": [ {
      "name":<String_value>,
      "connectclientlptports":<String_value>,
      "clientaudioredirection":<String_value>,
      "localremotedatasharing":<String_value>,
      "clientclipboardredirection":<String_value>,
      "clientcomportredirection":<String_value>,
      "clientdriveredirection":<String_value>,
      "clientprinterredirection":<String_value>,
      "multistream":<String_value>,
      "clientusbdriveredirection":<String_value>,
      "refcnt":<Double_value>,
      "builtin":<String[]_value>,
      "isdefault":<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile/name_value<String>?**attrs=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1,property-name2

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icaaccessprofile": [ {
    "name":<String_value>,
    "connectclientlptports":<String_value>,
    "clientaudioredirection":<String_value>,
    "localremotedatasharing":<String_value>,
    "clientclipboardredirection":<String_value>,
    "clientcomportredirection":<String_value>,
    "clientdriveredirection":<String_value>,
    "clientprinterredirection":<String_value>,
    "multistream":<String_value>,
    "clientusbdriveredirection":<String_value>,
    "refcnt":<Double_value>,
    "builtin":<String[]_value>,
    "isdefault":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaccessprofile?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "icaaccessprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

icaaction

May 20, 2016

Configuration for ica action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the ICA action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the ICA action is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my ica action" or 'my ica action'). Minimum length = 1
accessprofilename	<String>	Read-write	Name of the ica accessprofile to be associated with this action.
newname	<String>	Read-write	New name for the ICA action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#),period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my ica action" or 'my ica action'). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<Double>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
builtin	<String[]>	Read-only	Indicates that the ICA action is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default action.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD | DELETE | UPDATE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"icaaction":{
    "name":<String_value>,
    "accessprofilename":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"icaaction":{
    "name":<String_value>,
    "accessprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaction?**action=rename**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"icaaction":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/icaaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/icaaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of icaaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/icaaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/icaaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the icaaction resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icaaction": [ {
      "name":<String_value>,
      "accessprofilename":<String_value>,
      "hits":<Double_value>,
      "referencecount":<Double_value>,
      "undefhits":<Double_value>,
      "builtin":<String[]_value>,
      "isdefault":<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/icaaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/icaaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icaaction": [ {
      "name":<String_value>,
      "accessprofilename":<String_value>,
      "hits":<Double_value>,
      "referencecount":<Double_value>,
      "undefhits":<Double_value>,
      "builtin":<String[]_value>,
      "isdefault":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/icaaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icaaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

icaglobal_binding

May 20, 2016

Binding object which returns the resources bound to icaglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
icaglobal_icapolicy_binding	<icaglobal_icapolicy_binding[]>	Read-only	icapolicy that can be bound to icaglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/icaglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "icaglobal_binding": [ {
      "icaglobal_icapolicy_binding":<icaglobal_icapolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

icaglobal_icapolicy_binding

May 20, 2016

Binding object showing the icapolicy that can be bound to icaglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the ICA policy.
type	<String>	Read-write	Global bind point for which to show detailed information about the policies bound to the bind point. Possible values = ICA_REQ_OVERRIDE, ICA_REQ_DEFAULT
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
flowtype	<Double>	Read-only	Flow type of the bound ICA policy.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/icaglobal_icapolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"icaglobal_icapolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/icaglobal_icapolicy_binding

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/icaglobal_icapolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/icaglobal_icapolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of icaglobal_icapolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/icaglobal_icapolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the icaglobal_icapolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icaglobal_icapolicy_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "type":<String_value>,
    "numpol":<Double_value>,
    "flowtype":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/icaglobal_icapolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icaglobal_icapolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

icapolicy

Jul 07, 2017

Configuration for ICA policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my ica policy" or 'my ica policy').
rule	<String>	Read-write	Expression or other value against which the traffic is evaluated. Must be a Boolean, default syntax expression. Note: Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the ica action to be associated with this policy.
comment	<String>	Read-write	Any type of information about this ICA policy.
logaction	<String>	Read-write	Name of the messagelog action to use for requests that match this policy.
newname	<String>	Read-write	New name for the policy. Must begin with an ASCII alphabetic or underscore (_)character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), s pace, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my ica policy" or 'my ica policy'). Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
builtin	<String[]>	Read-only	Indicates that the ICA policy is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"icapolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"icapolicy":{
    "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"icapolicy":{
    "name":<String_value>,
    "comment":true,
    "logaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"icapolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/icapolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/icapolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of icapolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/icapolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/icapolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the icapolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icapolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "comment":<String_value>,
      "logaction":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/icapolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/icapolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icapolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icapolicy": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

icapolicy_binding

May 20, 2016

Binding object which returns the resources bound to icapolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy about which to display detailed information.
icapolicy_vpnvserver_binding	<icapolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to icapolicy.
icapolicy_icaglobal_binding	<icapolicy_icaglobal_binding[]>	Read-only	icaglobal that can be bound to icapolicy.
icapolicy_crvserver_binding	<icapolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to icapolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "icapolicy_binding": [ {
      "name":<String_value>,
      "icapolicy_crvserver_binding":<icapolicy_crvserver_binding[]_value>,
      "icapolicy_icaglobal_binding":<icapolicy_icaglobal_binding[]_value>,
      "icapolicy_vpnvserver_binding":<icapolicy_vpnvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

icapolicy_crvserver_binding

May 20, 2016

Binding object showing the crvserver that can be bound to icapolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy_crvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/icapolicy_crvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of icapolicy_crvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/icapolicy_crvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the icapolicy_crvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icapolicy_crvserver_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy_crvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"icapolicy_crvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

icapolicy_icaglobal_binding

Jul 07, 2017

Binding object showing the icaglobal that can be bound to icapolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy_icaglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/icapolicy_icaglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of icapolicy_icaglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/icapolicy_icaglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the icapolicy_icaglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "icapolicy_icaglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy_icaglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"icapolicy_icaglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

icapolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to icapolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the policy about which to display detailed information.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/icapolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of icapolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/icapolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the icapolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "icapolicy_vpnvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/icapolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"icapolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Ipsec

May 20, 2016

IPSEC configuration.

Name	Description
ipsecparameter	Configuration for IPSEC paramter
ipsecprofile	Configuration for IPSEC profile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ipsecparameter

Jul 07, 2017

Configuration for IPSEC paramter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ikeversion	<String>	Read-write	IKE Protocol Version. Default value: V2 Possible values = V1, V2
encalgo	<String[]>	Read-write	Type of encryption algorithm. Default value: AES Possible values = AES, 3DES
hashalgo	<String[]>	Read-write	Type of hashing algorithm. Default value: HMAC_SHA256 Possible values = HMAC_SHA1, HMAC_SHA256, HMAC_SHA384, HMAC_SHA512, HMAC_MD5
lifetime	<Double>	Read-write	Lifetime of IKE SA in seconds. Lifetime of IPSec SA will be (lifetime of IKE SA/8). Minimum value = 480 Maximum value = 31536000
livenesscheckinterval	<Double>	Read-write	Number of seconds after which a notify payload is sent to check the liveliness of the peer. Additional retries are done as per retransmit interval setting. Zero value disables liveliness checks. Minimum value = 0 Maximum value = 64999
replaywindowsize	<Double>	Read-write	IPSec Replay window size for the data traffic. Minimum value = 0 Maximum value = 16384
ikeretryinterval	<Double>	Read-write	IKE retry interval for bringing up the connection. Minimum value = 60 Maximum value = 3600
perfectforwardsecrecy	<String>	Read-write	Enable/Disable PFS. Default value: DISABLE Possible values = ENABLE, DISABLE
retransmissiontime	<Double>	Read-write	The interval in seconds to retry sending the IKE messages to peer, three consecutive attempts are done with doubled interval after every failure, increases for every retransmit till 6 retransmits. Minimum value = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (Click to show Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/ipsecparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ipsecparameter":{
    "ikeversion":<String_value>,
    "encalgo":<String[]_value>,
    "hashalgo":<String[]_value>,
    "lifetime":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "livenesscheckinterval":<Double_value>,
    "replaywindowsize":<Double_value>,
    "ikeretryinterval":<Double_value>,
    "perfectforwardsecrecy":<String_value>,
    "retransmissiontime":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/ipsecparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ipsecparameter":{
    "ikeversion":true,
    "encalgo":true,
    "hashalgo":true,
    "lifetime":true,
    "livenesscheckinterval":true,
    "replaywindowsize":true,
    "ikeretryinterval":true,
    "perfectforwardsecrecy":true,
    "retransmissiontime":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ipsecparameter

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ipsecparameter": [ {
      "ikeversion":<String_value>,
      "encalgo":<String[]_value>,
      "hashalgo":<String[]_value>,
      "lifetime":<Double_value>,
      "livenesscheckinterval":<Double_value>,
      "replaywindowsize":<Double_value>,
      "ikeretryinterval":<Double_value>,
      "perfectforwardsecrecy":<String_value>,
      "retransmissiontime":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ipsecprofile

May 20, 2016

Configuration for IPSEC profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the ipsec profile. Minimum length = 1 Maximum length = 32
ikeversion	<String>	Read-write	IKE Protocol Version. Possible values = V1, V2
encalgo	<String[]>	Read-write	Type of encryption algorithm. Possible values = AES, 3DES
hashalgo	<String[]>	Read-write	Type of hashing algorithm. Possible values = HMAC_SHA1, HMAC_SHA256, HMAC_SHA384, HMAC_SHA512, HMAC_MD5
lifetime	<Double>	Read-write	Lifetime of IKE SA in seconds. Lifetime of IPSec SA will be (lifetime of IKE SA/8). Minimum value = 480 Maximum value = 31536000
psk	<String>	Read-write	Pre shared key value.
publickey	<String>	Read-write	Public key file path.
privatekey	<String>	Read-write	Private key file path.
peerpublickey	<String>	Read-write	Peer public key file path.
livenesscheckinterval	<Double>	Read-write	Number of seconds after which a notify payload is sent to check the liveliness of the peer. Additional retries are done as per retransmit interval setting. Zero value disables liveliness checks. Minimum value = 0 Maximum value = 64999
replaywindowsize	<Double>	Read-write	IPSec Replay window size for the data traffic. Minimum value = 0 Maximum value = 16384

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ikeretryinterval	<Double>	Read-write	Minimum value = 60 Maximum value = 3600
retransmissiontime	<Double>	Read-write	The interval in seconds to retry sending the IKE messages to peer, three consecutive attempts are done with doubled interval after every failure. Minimum value = 1 Maximum value = 99
perfectforwardsecrecy	<String>	Read-write	Enable/Disable PFS. Possible values = ENABLE, DISABLE
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ipsecprofile":{
    "name":<String_value>,
    "ikeversion":<String_value>,
    "encalgo":<String[]_value>,
    "hashalgo":<String[]_value>,
    "lifetime":<Double_value>,
    "psk":<String_value>,
    "publickey":<String_value>,
    "privatekey":<String_value>,
    "peerpublickey":<String_value>,
    "livenesscheckinterval":<Double_value>,
    "replaywindowsize":<Double_value>,
    "ikeretryinterval":<Double_value>,
    "retransmissiontime":<Double_value>,
    "perfectforwardsecrecy":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ipsecprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the ipsecprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ipsecprofile": [ {
      "name":<String_value>,
      "ikeversion":<String_value>,
      "encalgo":<String[]_value>,
      "hashalgo":<String[]_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "lifetime":<Double_value>,
    "livenesscheckinterval":<Double_value>,
    "replaywindowsize":<Double_value>,
    "retransmissiontime":<Double_value>,
    "psk":<String_value>,
    "publickey":<String_value>,
    "privatekey":<String_value>,
    "peerpublickey":<String_value>,
    "ikeretryinterval":<Double_value>,
    "perfectforwardsecrecy":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ipsecprofile": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "ikeversion":<String_value>,
    "encalgo":<String[]_value>,
    "hashalgo":<String[]_value>,
    "lifetime":<Double_value>,
    "livenesscheckinterval":<Double_value>,
    "replaywindowsize":<Double_value>,
    "retransmissiontime":<Double_value>,
    "psk":<String_value>,
    "publickey":<String_value>,
    "privatekey":<String_value>,
    "peerpublickey":<String_value>,
    "ikeretryinterval":<Double_value>,
    "perfectforwardsecrecy":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/ipsecprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ipsecprofile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing

May 20, 2016

Load Balancing configuration. The load balancing methods manage the selection of an appropriate physical server in a server farm.

Name	Description
lbgroup	Configuration for LB group
lbgroup_binding	Binding object showing the resources that can be bound to lbgroup
lbgroup_lbvserver_binding	Binding object showing the lbvserver that can be bound to lbgroup
lbmetrictable	Configuration for metric table
lbmetrictable_binding	Binding object showing the resources that can be bound to lbmetrictable
lbmetrictable_metric_binding	Binding object showing the metric that can be bound to lbmetrictable
lbmonbindings	Configuration for monitro bindings
lbmonbindings_binding	Binding object showing the resources that can be bound to lbmonbindings
lbmonbindings_service_binding	Binding object showing the service that can be bound to lbmonbindings
lbmonbindings_servicegroup_binding	Binding object showing the servicegroup that can be bound to lbmonbindings
lbmonitor	Configuration for monitor
lbmonitor_binding	Binding object showing the resources that can be bound to lbmonitor
lbmonitor_metric_binding	Binding object showing the metric that can be bound to lbmonitor
lbmonitor_service_binding	Binding object showing the service that can be bound to lbmonitor
lbmonitor_servicegroup_binding	Binding object showing the servicegroup that can be bound to lbmonitor

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
lbparameter	Configuration for LB parameter
lbpersistentsessions	Configuration for persistence session
lbroute	Configuration for LB route
lbroute6	Configuration for LB route6
lbsipparameters	Configuration for SIP parameters
lbvserver	Configuration for Load Balancing Virtual Server
lbvserver_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to lbvserver
lbvserver_appfwpolicy_binding	Binding object showing the appfwpolicy that can be bound to lbvserver
lbvserver_appqoepolicy_binding	Binding object showing the appqoepolicy that can be bound to lbvserver
lbvserver_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to lbvserver
lbvserver_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to lbvserver
lbvserver_authorizationpolicy_binding	Binding object showing the authorizationpolicy that can be bound to lbvserver
lbvserver_binding	Binding object showing the resources that can be bound to lbvserver
lbvserver_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to lbvserver
lbvserver_capolicy_binding	Binding object showing the capolicy that can be bound to lbvserver
lbvserver_cmppolicy_binding	Binding object showing the cmppolicy that can be bound to lbvserver
lbvserver_csvserver_binding	Binding object showing the csvserver that can be bound to lbvserver
lbvserver_dnspolicy64_binding	Binding object showing the dnspolicy64 that can be bound to lbvserver
lbvserver_dospolicy_binding	Binding object showing the dospolicy that can be bound to lbvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
lbvserver_feopolicy_binding	Binding object showing the feopolicy that can be bound to lbvserver
lbvserver_filterpolicy_binding	Binding object showing the filterpolicy that can be bound to lbvserver
lbvserver_pqpolicy_binding	Binding object showing the pqpolicy that can be bound to lbvserver
lbvserver_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to lbvserver
lbvserver_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to lbvserver
lbvserver_scpolicy_binding	Binding object showing the scpolicy that can be bound to lbvserver
lbvserver_service_binding	Binding object showing the service that can be bound to lbvserver
lbvserver_servicegroup_binding	Binding object showing the servicegroup that can be bound to lbvserver
lbvserver_servicegroupmember_binding	Binding object showing the servicegroupmember that can be bound to lbvserver
lbvserver_spilloverpolicy_binding	Binding object showing the spilloverpolicy that can be bound to lbvserver
lbvserver_tmtrafficpolicy_binding	Binding object showing the tmtrafficpolicy that can be bound to lbvserver
lbvserver_transformpolicy_binding	Binding object showing the transformpolicy that can be bound to lbvserver
lbwlm	Configuration for web log manager
lbwlm_binding	Binding object showing the resources that can be bound to lbwlm
lbwlm_lbvserver_binding	Binding object showing the lbvserver that can be bound to lbwlm

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbgroup

Jul 07, 2017

Configuration for LB group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the load balancing virtual server group. Minimum length = 1
persistencetype	<String>	Read-write	Type of persistence for the group. Available settings function as follows: * SOURCEIP - Create persistence sessions based on the client IP. * COOKIEINSERT - Create persistence sessions based on a cookie in client requests. The cookie is inserted by a Set-Cookie directive from the server, in its first response to a client. * RULE - Create persistence sessions based on a user defined rule. * NONE - Disable persistence for the group. Possible values = SOURCEIP, COOKIEINSERT, RULE, NONE
persistencebackup	<String>	Read-write	Type of backup persistence for the group. Possible values = SOURCEIP, NONE
backuppersistencetimeout	<Double>	Read-write	Time period, in minutes, for which backup persistence is in effect. Default value: 2 Minimum value = 2 Maximum value = 1440
persistmask	<String>	Read-write	Persistence mask to apply to source IPv4 addresses when creating source IP based persistence sessions. Minimum length = 1
cookiename	<String>	Read-write	Use this parameter to specify the cookie name for COOKIE peristence type. It specifies the name of cookie with a maximum of 32 characters. If not specified, cookie name is internally generated.
v6persistmasklen	<Double>	Read-write	Persistence mask to apply to source IPv6 addresses when creating source IP based persistence sessions. Default value: 128 Minimum value = 1 Maximum value = 128
cookiedomain	<String>	Read-write	Domain attribute for the HTTP cookie. Minimum length = 1
			Time period for which a persistence session is in effect. Default value: 2

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum value = 1440
rule	<String>	Read-write	Expression, or name of a named expression, against which traffic is evaluated. Written in the classic or default syntax. Note: Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + " <string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks. Default value: "None"
newname	<String>	Read-write	New name for the load balancing virtual server group. Minimum length = 1
td	<Double>	Read-only	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lbgroup

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbgroup":{
    "name":<String_value>,
    "persistencetype":<String_value>,
    "persistencebackup":<String_value>,
    "backuppersistencetimeout":<Double_value>,
    "persistmask":<String_value>,
    "cookiename":<String_value>,
    "v6persistmasklen":<Double_value>,
    "cookiedomain":<String_value>,
    "timeout":<Double_value>,
    "rule":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lbgroup?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"lbgroup":{
    "name":<String_value>,
    "persistencetype":true,
    "persistencebackup":true,
    "backuppersistencetimeout":true,
    "persistmask":true,
    "cookiename":true,
    "v6persistmasklen":true,
    "cookiedomain":true,
    "timeout":true,
    "rule":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/lbgroup?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbgroup":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbgroup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbgroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbgroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbgroup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lbgroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbgroup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbgroup resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbgroup": [ {
      "name":<String_value>,
      "persistencetype":<String_value>,
      "persistencebackup":<String_value>,
      "backuppersistencetimeout":<Double_value>,
      "persistmask":<String_value>,
      "v6persistmasklen":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cookiename":<String_value>,
    "cookiedomain":<String_value>,
    "timeout":<Double_value>,
    "rule":<String_value>,
    "td":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbgroup/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbgroup/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lbgroup/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbgroup": [ {
    "name":<String_value>,
    "persistencetype":<String_value>,
    "persistencebackup":<String_value>,
    "backuppersistencetimeout":<Double_value>,
    "persistmask":<String_value>,
    "v6persistmasklen":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cookiename":<String_value>,
    "cookiedomain":<String_value>,
    "timeout":<Double_value>,
    "rule":<String_value>,
    "td":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbgroup?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbgroup": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbgroup_binding

May 20, 2016

Binding object which returns the resources bound to lbgroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the load balancing virtual server group. Minimum length = 1
lbgroup_lbvserver_binding	<lbgroup_lbvserver_binding[]>	Read-only	lbvserver that can be bound to lbgroup.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbgroup_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lbgroup_binding": [ {
      "name":<String_value>,
      "lbgroup_lbvserver_binding":<lbgroup_lbvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbgroup_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to lbgroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the load balancing virtual server group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my lbgroup" or 'my lbgroup'). Minimum length = 1
vservername	<String>	Read-write	Virtual server name.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbgroup_lbvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbgroup_lbvserver_binding":{
    "name":<String_value>,
    "vservername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbgroup_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbgroup_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbgroup_lbvserver_binding": [ {
    "name":<String_value>,
    "vservername":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbgroup_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbgroup_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmetrictable

May 20, 2016

Configuration for metric table resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
metrictable	<String>	Read-write	Name for the metric table. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my metrictable" or 'my metrictable'). Minimum length = 1 Maximum length = 31
metric	<String>	Read-write	Name of the metric for which to change the SNMP OID. Minimum length = 1
Snmpoid	<String>	Read-write	New SNMP OID of the metric. Minimum length = 1
metrictype	<String>	Read-only	Indication if it is a configured or internal. Possible values = INTERNAL, CONFIGURED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbmetrictable":{
    "metrictable":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable/metrictable_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbmetrictable":{
    "metrictable":<String_value>,
    "metric":<String_value>,
    "Snmpoid":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbmetrictable resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbmetrictable resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbmetrictable": [ {
      "metrictable":<String_value>,
      "metric":<String_value>,
      "Snmpoid":<String_value>,
      "flags":<Boolean_value>,
      "metrictype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable/metrictable_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable/metrictable_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable/metrictable_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbmetrictable": [ {
      "metrictable":<String_value>,
      "metric":<String_value>,
      "Snmpoid":<String_value>,
      "flags":<Boolean_value>,
      "metrictype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{ "lbmetrictable": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmetrictable_binding

Jul 07, 2017

Binding object which returns the resources bound to lbmetrictable.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
metrictable	<String>	Read-write	Name of the metric table. Minimum length = 1 Maximum length = 31
lbmetrictable_metric_binding	<lbmetrictable_metric_binding[]>	Read-only	metric that can be bound to lbmetrictable.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable_binding/metrictable_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lbmetrictable_binding": [ {
      "metrictable":<String_value>,
      "lbmetrictable_metric_binding":<lbmetrictable_metric_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmetrictable_metric_binding

May 20, 2016

Binding object showing the metric that can be bound to lbmetrictable.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
metric	<String>	Read-write	Name of the metric for which to change the SNMP OID. Minimum length = 1
metrictable	<String>	Read-write	Name of the metric table. Minimum length = 1 Maximum length = 31
Snmpoid	<String>	Read-write	New SNMP OID of the metric. Minimum length = 1
metrictype	<String>	Read-only	Indication if it is a configured or internal. Possible values = INTERNAL, CONFIGURED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbmetrictable_metric_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbmetrictable_metric_binding":{
    "metrictable":<String_value>,
    "metric":<String_value>,
    "Snmpoid":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbmetrictable_metric_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbmetrictable_metric_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbmetrictable_metric_binding": [ {
    "metric":<String_value>,
    "metrictable":<String_value>,
    "Snmpoid":<String_value>,
    "metrictype":<String_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmetrictable_metric_binding/metrictable_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbmetrictable_metric_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmonbindings

May 20, 2016

Configuration for monitro bindings resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
monitorname	<String>	Read-write	The name of the monitor. Minimum length = 1
type	<String>	Read-only	The type of monitor. Possible values = PING, TCP, HTTP, TCP-ECV, HTTP-ECV, UDP-ECV, DNS, FTP, LDNS-PING, LDNS-TCP, LDNS-DNS, RADIUS, USER, HTTP-INLINE, SIP-UDP, SIP-TCP, LOAD, FTP-EXTENDED, SMTP, SNMP, NNTP, MYSQL, MYSQL-ECV, MSSQL-ECV, ORACLE-ECV, LDAP, POP3, CITRIX-XML-SERVICE, CITRIX-WEB-INTERFACE, DNS-TCP, RTSP, ARP, CITRIX-AG, CITRIX-AAC-LOGINPAGE, CITRIX-AAC-LAS, CITRIX-XD-DDC, ND6, CITRIX-WI-EXTENDED, DIAMETER, RADIUS_ACCOUNTING, STOREFRONT, APPC, SMPP, CITRIX-XNC-ECV, CITRIX-XDM
state	<String>	Read-only	The state of the monitor. Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings?**args=monitorname:<String_value>,**

Use this query-parameter to get lbmonbindings resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbmonbindings resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbmonbindings resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbmonbindings": [ {
monitorname:<String_value>,      "type":<String_value>,
      "state":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings/monitorname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings/monitorname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings/monitorname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbmonbindings": [ {
monitorname:<String_value>,      "type":<String_value>,
      "state":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings?**args=monitorname:<String_value>,&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbmonbindings": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmonbindings_binding

May 20, 2016

Binding object which returns the resources bound to lbmonbindings.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
monitorname	<String>	Read-write	The name of the monitor. Minimum length = 1
lbmonbindings_servicegroup_binding	<lbmonbindings_servicegroup_binding[]>	Read-only	servicegroup that can be bound to lbmonbindings.
lbmonbindings_service_binding	<lbmonbindings_service_binding[]>	Read-only	service that can be bound to lbmonbindings.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings_binding/monitorname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lbmonbindings_binding": [ {
      "monitorname":<String_value>,
      "lbmonbindings_service_binding":<lbmonbindings_service_binding[]_value>,
      "lbmonbindings_servicegroup_binding":<lbmonbindings_servicegroup_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmonbindings_servicegroup_binding

Jul 07, 2017

Binding object showing the servicegroup that can be bound to lbmonbindings.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
monitorname	<String>	Read-write	The name of the monitor. Minimum length = 1
servicegroupname	<String>	Read-write	The name of the service group.
boundservicegroupsvrstate	<String>	Read-only	The state of the servicegroup. Possible values = ENABLED, DISABLED
monstate	<String>	Read-only	The configured state (enable/disable) of Monitor on this service. Possible values = ENABLED, DISABLED
servicetype	<String>	Read-only	The type of service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings_servicegroup_binding/monitorname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings_servicegroup_binding/monitorname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbmonbindings_servicegroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings_servicegroup_binding/monitorname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbmonbindings_servicegroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "lbmonbindings_servicegroup_binding": [ {
    "monitorname":<String_value>,
    "servicegroupname":<String_value>,
    "boundservicegroupsvrstate":<String_value>,
    "monstate":<String_value>,
    "servicetype":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings_servicegroup_binding/monitorname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbmonbindings_servicegroup_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmonbindings_service_binding

May 20, 2016

Binding object showing the service that can be bound to lbmonbindings.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	The name of the service.
monitorname	<String>	Read-write	The name of the monitor. Minimum length = 1
servicetype	<String>	Read-only	The type of service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX
svrstate	<String>	Read-only	The state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
monsvcstate	<String>	Read-only	The configured state (enable/disable) of Monitor on this service. Possible values = ENABLED, DISABLED
port	<Integer>	Read-only	The port of the service. Range 1 - 65535
ipaddress	<String>	Read-only	The IPAddress of the service.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings_service_binding/monitorname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings_service_binding/monitorname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbmonbindings_service_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings_service_binding/monitorname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbmonbindings_service_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbmonbindings_service_binding": [ {
      "servicename":<String_value>,
      "monitorname":<String_value>,
      "servicetype":<String_value>,
      "svrstate":<String_value>,
      "monsvcstate":<String_value>,
      "port":<Integer_value>,
      "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonbindings_service_binding/monitorname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbmonbindings_service_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmonitor

May 20, 2016

Configuration for monitor resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
monitorname	<String>	Read-write	Name for the monitor. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my monitor" or 'my monitor'). Minimum length = 1
type	<String>	Read-write	Type of monitor that you want to create. Possible values = PING, TCP, HTTP, TCP-ECV, HTTP-ECV, UDP-ECV, DNS, FTP, LDNS-PING, LDNS-TCP, LDNS-DNS, RADIUS, USER, HTTP-INLINE, SIP-UDP, SIP-TCP, LOAD, FTP-EXTENDED, SMTP, SNMP, NNTP, MYSQL, MYSQL-ECV, MSSQL-ECV, ORACLE-ECV, LDAP, POP3, CITRIX-XML-SERVICE, CITRIX-WEB-INTERFACE, DNS-TCP, RTSP, ARP, CITRIX-AG, CITRIX-AAC-LOGINPAGE, CITRIX-AAC-LAS, CITRIX-XD-DDC, ND6, CITRIX-WI-EXTENDED, DIAMETER, RADIUS_ACCOUNTING, STOREFRONT, APPC, SMPP, CITRIX-XNC-ECV, CITRIX-XDM
action	<String>	Read-write	Action to perform when the response to an inline monitor (a monitor of type HTTP-INLINE) indicates that the service is down. A service monitored by an inline monitor is considered DOWN if the response code is not one of the codes that have been specified for the Response Code parameter. Available settings function as follows: * NONE - Do not take any action. However, the show service command and the show lb monitor command indicate the total number of responses that were checked and the number of consecutive error responses received after the last successful probe. * LOG - Log the event in NSLOG or SYSLOG. * DOWN - Mark the service as being down, and then do not direct any traffic to the service until the configured down time has expired. Persistent connections to the service are terminated as soon as the service is marked as DOWN. Also, log the event in NSLOG or SYSLOG. Default value: DOWN Possible values = NONE, LOG, DOWN

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
respcode	<String[]>	Read-write	response code, the action performed depends on the monitor type. HTTP monitors and RADIUS monitors mark the service as DOWN, while HTTP-INLINE monitors perform the action indicated by the Action parameter.
httprequest	<String>	Read-write	HTTP request to send to the server (for example, "HEAD /file.html").
rtsprequest	<String>	Read-write	RTSP request to send to the server (for example, "OPTIONS *").
customheaders	<String>	Read-write	Custom header string to include in the monitoring probes.
maxforwards	<Double>	Read-write	Maximum number of hops that the SIP request used for monitoring can traverse to reach the server. Applicable only to monitors of type SIP-UDP. Default value: 1 Minimum value = 0 Maximum value = 255
sipmethod	<String>	Read-write	SIP method to use for the query. Applicable only to monitors of type SIP-UDP. Possible values = OPTIONS, INVITE, REGISTER
sipuri	<String>	Read-write	SIP URI string to send to the service (for example, sip:sip.test). Applicable only to monitors of type SIP-UDP. Minimum length = 1
sipreguri	<String>	Read-write	SIP user to be registered. Applicable only if the monitor is of type SIP-UDP and the SIP Method parameter is set to REGISTER. Minimum length = 1
send	<String>	Read-write	String to send to the service. Applicable to TCP-ECV, HTTP-ECV, and UDP-ECV monitors.
recv	<String>	Read-write	String expected from the server for the service to be marked as UP. Applicable to TCP-ECV, HTTP-ECV, and UDP-ECV monitors.
query	<String>	Read-write	Domain name to resolve as part of monitoring the DNS service (for example, example.com).
querytype	<String>	Read-write	Type of DNS record for which to send monitoring queries. Set to Address for querying A records, AAAA for querying AAAA records, and Zone for querying the SOA record. Possible values = Address, Zone, AAAA
scriptname	<String>	Read-write	Path and name of the script to execute. The script must be available on the NetScaler appliance, in the /nsconfig/monitors/ directory. Minimum length = 1
scriptargs	<String>	Read-write	String of arguments for the script. The string is copied verbatim into the request.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
dispatcherport	<Integer>	Read-write	Port number on which the dispatcher listens for the monitoring probe.
username	<String>	Read-write	User name with which to probe the RADIUS, NNTP, FTP, FTP-EXTENDED, MYSQL, MSSQL, POP3, CITRIX-AG, CITRIX-XD-DDC, CITRIX-WI-EXTENDED, CITRIX-XNC or CITRIX-XDM server. Minimum length = 1
password	<String>	Read-write	Password that is required for logging on to the RADIUS, NNTP, FTP, FTP-EXTENDED, MYSQL, MSSQL, POP3, CITRIX-AG, CITRIX-XD-DDC, CITRIX-WI-EXTENDED, CITRIX-XNC-ECV or CITRIX-XDM server. Used in conjunction with the user name specified for the User Name parameter. Minimum length = 1
secondarypassword	<String>	Read-write	Secondary password that users might have to provide to log on to the Access Gateway server. Applicable to CITRIX-AG monitors.
logonpointname	<String>	Read-write	Name of the logon point that is configured for the Citrix Access Gateway Advanced Access Control software. Required if you want to monitor the associated login page or Logon Agent. Applicable to CITRIX-AAC-LAS and CITRIX-AAC-LOGINPAGE monitors.
lasversion	<String>	Read-write	Version number of the Citrix Advanced Access Control Logon Agent. Required by the CITRIX-AAC-LAS monitor.
radkey	<String>	Read-write	Authentication key (shared secret text string) for RADIUS clients and servers to exchange. Applicable to monitors of type RADIUS and RADIUS_ACCOUNTING. Minimum length = 1
radnasid	<String>	Read-write	NAS-Identifier to send in the Access-Request packet. Applicable to monitors of type RADIUS. Minimum length = 1
radnasip	<String>	Read-write	Network Access Server (NAS) IP address to use as the source IP address when monitoring a RADIUS server. Applicable to monitors of type RADIUS and RADIUS_ACCOUNTING.
radaccounttype	<Double>	Read-write	Account Type to be used in Account Request Packet. Applicable to monitors of type RADIUS_ACCOUNTING. Default value: 1 Minimum value = 0 Maximum value = 15
radframedip	<String>	Read-write	Source ip with which the packet will go out. Applicable to monitors of type RADIUS_ACCOUNTING.
radapn	<String>	Read-write	Called Station Id to be used in Account Request Packet. Applicable to monitors of type RADIUS_ACCOUNTING. Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
radmsisdn	<String>	Read-write	to monitors of type RADIUS_ACCOUNTING. Minimum length = 1
radaccountsession	<String>	Read-write	Account Session ID to be used in Account Request Packet. Applicable to monitors of type RADIUS_ACCOUNTING. Minimum length = 1
lrtm	<String>	Read-write	Calculate the least response times for bound services. If this parameter is not enabled, the appliance does not learn the response times of the bound services. Also used for LRTM load balancing. Possible values = ENABLED, DISABLED
deviation	<Double>	Read-write	Time value added to the learned average response time in dynamic response time monitoring (DRTM). When a deviation is specified, the appliance learns the average response time of bound services and adds the deviation to the average. The final value is then continually adjusted to accommodate response time variations over time. Specified in milliseconds, seconds, or minutes. Minimum value = 0 Maximum value = 20939000
units1	<String>	Read-write	Unit of measurement for the Deviation parameter. Cannot be changed after the monitor is created. Default value: SEC Possible values = SEC, MSEC, MIN
interval	<Integer>	Read-write	Time interval between two successive probes. Must be greater than the value of Response Time-out. Default value: 5 Minimum value = 1 Maximum value = 20940000
units3	<String>	Read-write	monitor interval units. Default value: SEC Possible values = SEC, MSEC, MIN
resptimeout	<Integer>	Read-write	Amount of time for which the appliance must wait before it marks a probe as FAILED. Must be less than the value specified for the Interval parameter. Note: For UDP-ECV monitors for which a receive string is not configured, response timeout does not apply. For UDP-ECV monitors with no receive string, probe failure is indicated by an ICMP port unreachable error received from the service. Default value: 2 Minimum value = 1 Maximum value = 20939000
units4	<String>	Read-write	monitor response timeout units. Default value: SEC Possible values = SEC, MSEC, MIN
			Response time threshold, specified as a percentage of the Response Time-out parameter. If the response to a monitor probe

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
resptimeoutthresh	<Double>	Read-write	generates an SNMP trap called monRespTimeoutAboveThresh. After the response time returns to a value below the threshold, the appliance generates a monRespTimeoutBelowThresh SNMP trap. For the traps to be generated, the "MONITOR-RTO-THRESHOLD" alarm must also be enabled. Minimum value = 0 Maximum value = 100
retries	<Integer>	Read-write	Maximum number of probes to send to establish the state of a service for which a monitoring probe failed. Default value: 3 Minimum value = 1 Maximum value = 127
failureretries	<Integer>	Read-write	Number of retries that must fail, out of the number specified for the Retries parameter, for a service to be marked as DOWN. For example, if the Retries parameter is set to 10 and the Failure Retries parameter is set to 6, out of the ten probes sent, at least six probes must fail if the service is to be marked as DOWN. The default value of 0 means that all the retries must fail if the service is to be marked as DOWN. Minimum value = 0 Maximum value = 32
alertretries	<Integer>	Read-write	Number of consecutive probe failures after which the appliance generates an SNMP trap called monProbeFailed. Minimum value = 0 Maximum value = 32
successretries	<Integer>	Read-write	Number of consecutive successful probes required to transition a service's state from DOWN to UP. Default value: 1 Minimum value = 1 Maximum value = 32
downtime	<Integer>	Read-write	Time duration for which to wait before probing a service that has been marked as DOWN. Expressed in milliseconds, seconds, or minutes. Default value: 30 Minimum value = 1 Maximum value = 20939000
units2	<String>	Read-write	Unit of measurement for the Down Time parameter. Cannot be changed after the monitor is created. Default value: SEC Possible values = SEC, MSEC, MIN
destip	<String>	Read-write	IP address of the service to which to send probes. If the parameter is set to 0, the IP address of the server to which the monitor is bound is considered the destination IP address.
destport	<Integer>	Read-write	TCP or UDP port to which to send the probe. If the parameter is set to 0, the port number of the service to which the monitor is bound is considered the destination port. For a monitor of type USER,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			the HTTP request sent to the dispatcher. Does not apply to monitors of type PING.
state	<String>	Read-write	State of the monitor. The DISABLED setting disables not only the monitor being configured, but all monitors of the same type, until the parameter is set to ENABLED. If the monitor is bound to a service, the state of the monitor is not taken into account when the state of the service is determined. Default value: ENABLED Possible values = ENABLED, DISABLED
reverse	<String>	Read-write	Mark a service as DOWN, instead of UP, when probe criteria are satisfied, and as UP instead of DOWN when probe criteria are not satisfied. Default value: NO Possible values = YES, NO
transparent	<String>	Read-write	The monitor is bound to a transparent device such as a firewall or router. The state of a transparent device depends on the responsiveness of the services behind it. If a transparent device is being monitored, a destination IP address must be specified. The probe is sent to the specified IP address by using the MAC address of the transparent device. Default value: NO Possible values = YES, NO
iptunnel	<String>	Read-write	Send the monitoring probe to the service through an IP tunnel. A destination IP address must be specified. Default value: NO Possible values = YES, NO
tos	<String>	Read-write	Probe the service by encoding the destination IP address in the IP TOS (6) bits. Possible values = YES, NO
tosid	<Double>	Read-write	The TOS ID of the specified destination IP. Applicable only when the TOS parameter is set. Minimum value = 1 Maximum value = 63
secure	<String>	Read-write	Use a secure SSL connection when monitoring a service. Applicable only to TCP based monitors. The secure option cannot be used with a CITRIX-AG monitor, because a CITRIX-AG monitor uses a secure connection by default. Default value: NO Possible values = YES, NO
validatecred	<String>	Read-write	Validate the credentials of the Xen Desktop DDC server user. Applicable to monitors of type CITRIX-XD-DDC. Default value: NO Possible values = YES, NO
domain	<String>	Read-write	Domain in which the XenDesktop Desktop Delivery Controller (DDC) servers or Web Interface servers are present. Required by CITRIX-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			DDC servers and Web Interface servers, respectively.
ipaddress	<String[]>	Read-write	Set of IP addresses expected in the monitoring response from the DNS server, if the record type is A or AAAA. Applicable to DNS monitors. Minimum length = 1
group	<String>	Read-write	Name of a newsgroup available on the NNTP service that is to be monitored. The appliance periodically generates an NNTP query for the name of the newsgroup and evaluates the response. If the newsgroup is found on the server, the service is marked as UP. If the newsgroup does not exist or if the search fails, the service is marked as DOWN. Applicable to NNTP monitors. Minimum length = 1
filename	<String>	Read-write	Name of a file on the FTP server. The appliance monitors the FTP service by periodically checking the existence of the file on the server. Applicable to FTP-EXTENDED monitors. Minimum length = 1
basedn	<String>	Read-write	The base distinguished name of the LDAP service, from where the LDAP server can begin the search for the attributes in the monitoring query. Required for LDAP service monitoring. Minimum length = 1
binddn	<String>	Read-write	The distinguished name with which an LDAP monitor can perform the Bind operation on the LDAP server. Optional. Applicable to LDAP monitors. Minimum length = 1
filter	<String>	Read-write	Filter criteria for the LDAP query. Optional. Minimum length = 1
attribute	<String>	Read-write	Attribute to evaluate when the LDAP server responds to the query. Success or failure of the monitoring probe depends on whether the attribute exists in the response. Optional. Minimum length = 1
database	<String>	Read-write	Name of the database to connect to during authentication. Minimum length = 1
oraclesid	<String>	Read-write	Name of the service identifier that is used to connect to the Oracle database during authentication. Minimum length = 1
sqlquery	<String>	Read-write	SQL query for a MYSQL-ECV or MSSQL-ECV monitor. Sent to the database server after the server authenticates the connection. Minimum length = 1
evalrule	<String>	Read-write	Default syntax expression that evaluates the database server's response to a MYSQL-ECV or MSSQL-ECV monitoring query. Must produce a Boolean result. The result determines the state of the server. If the expression returns TRUE, the probe succeeds.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			message to determine the state of the server, use the rule MYSQL.RES.ROW(10) .TEXT_ELEM(2).EQ("MySQL").
mssqlprotocolversion	<String>	Read-write	Version of MSSQL server that is to be monitored. Default value: 70 Possible values = 70, 2000, 2000SP1, 2005, 2008, 2008R2, 2012, 2014
Snmpoid	<String>	Read-write	SNMP OID for SNMP monitors. Minimum length = 1
snmpcommunity	<String>	Read-write	Community name for SNMP monitors. Minimum length = 1
snmpthreshold	<String>	Read-write	Threshold for SNMP monitors. Minimum length = 1
snmpversion	<String>	Read-write	SNMP version to be used for SNMP monitors. Possible values = V1, V2
metrictable	<String>	Read-write	Metric table to which to bind metrics. Minimum length = 1 Maximum length = 99
application	<String>	Read-write	Name of the application used to determine the state of the service. Applicable to monitors of type CITRIX-XML-SERVICE. Minimum length = 1
sitepath	<String>	Read-write	URL of the logon page. For monitors of type CITRIX-WEB-INTERFACE, to monitor a dynamic page under the site path, terminate the site path with a slash (/). Applicable to CITRIX-WEB-INTERFACE, CITRIX-WI-EXTENDED and CITRIX-XDM monitors. Minimum length = 1
storename	<String>	Read-write	Store Name. For monitors of type STOREFRONT, STORENAME is an optional argument defining storefront service store name. Applicable to STOREFRONT monitors. Minimum length = 1
storefrontacctservice	<String>	Read-write	Enable/Disable probing for Account Service. Applicable only to Store Front monitors. For multi-tenancy configuration users my skip account service. Default value: YES Possible values = YES, NO
hostname	<String>	Read-write	Hostname in the FQDN format (Example: porche.cars.org). Applicable to STOREFRONT monitors. Minimum length = 1
netprofile	<String>	Read-write	Name of the network profile. Minimum length = 1 Maximum length = 127
			Origin-Host value for the Capabilities-Exchange-Request (CER)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
originrealm	<String>	Read-write	Origin-Realm value for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. Minimum length = 1
hostipaddress	<String>	Read-write	Host-IP-Address value for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. If Host-IP-Address is not specified, the appliance inserts the mapped IP (MIP) address or subnet IP (SNIP) address from which the CER request (the monitoring probe) is sent. Minimum length = 1
vendorid	<Double>	Read-write	Vendor-Id value for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers.
productname	<String>	Read-write	Product-Name value for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. Minimum length = 1
firmwarerevision	<Double>	Read-write	Firmware-Revision value for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers.
authapplicationid	<Double[]>	Read-write	List of Auth-Application-Id attribute value pairs (AVPs) for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. A maximum of eight of these AVPs are supported in a monitoring CER message. Minimum value = 0 Maximum value = 4294967295
acctapplicationid	<Double[]>	Read-write	List of Acct-Application-Id attribute value pairs (AVPs) for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. A maximum of eight of these AVPs are supported in a monitoring message. Minimum value = 0 Maximum value = 4294967295
inbandsecurityid	<String>	Read-write	Inband-Security-Id for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. Possible values = NO_INBAND_SECURITY, TLS
supportedvendorids	<Double[]>	Read-write	List of Supported-Vendor-Id attribute value pairs (AVPs) for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. A maximum eight of these AVPs are supported in a monitoring message. Minimum value = 1 Maximum value = 4294967295
vendorspecificvendorid	<Double>	Read-write	Vendor-Id to use in the Vendor-Specific-Application-Id grouped attribute-value pair (AVP) in the monitoring CER message. To specify Auth-Application-Id or Acct-Application-Id in Vendor-Specific-Application-Id, use vendorSpecificAuthApplicationIds or vendorSpecificAcctApplicationIds, respectively. Only one Vendor-Id is supported for all the Vendor-Specific-Application-Id AVPs in a CER

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 1
vendorspecificauthapplicationids	<Double[]>	Read-write	List of Vendor-Specific-Auth-Application-Id attribute value pairs (AVPs) for the Capabilities-Exchange-Request (CER) message to use for monitoring Diameter servers. A maximum of eight of these AVPs are supported in a monitoring message. The specified value is combined with the value of vendorSpecificVendorId to obtain the Vendor-Specific-Application-Id AVP in the CER monitoring message. Minimum value = 0 Maximum value = 4294967295
vendorspecificacctapplicationids	<Double[]>	Read-write	List of Vendor-Specific-Acct-Application-Id attribute value pairs (AVPs) to use for monitoring Diameter servers. A maximum of eight of these AVPs are supported in a monitoring message. The specified value is combined with the value of vendorSpecificVendorId to obtain the Vendor-Specific-Application-Id AVP in the CER monitoring message. Minimum value = 0 Maximum value = 4294967295
kcdaccount	<String>	Read-write	KCD Account used by MSSQL monitor. Minimum length = 1 Maximum length = 32
storedb	<String>	Read-write	Store the database list populated with the responses to monitor probes. Used in database specific load balancing if MSSQL-ECV/MYSQL-ECV monitor is configured. Default value: DISABLED Possible values = ENABLED, DISABLED
storefrontcheckbackendservices	<String>	Read-write	This option will enable monitoring of services running on storefront server. Storefront services are monitored by probing to a Windows service that runs on the Storefront server and exposes details of which storefront services are running. Default value: NO Possible values = YES, NO
trofscode	<Double>	Read-write	Code expected when the server is under maintenance.
trofsstring	<String>	Read-write	String expected from the server for the service to be marked as trofs. Applicable to HTTP-ECV/TCP-ECV monitors.
metric	<String>	Read-write	Metric name in the metric table, whose setting is changed. A value zero disables the metric and it will not be used for load calculation. Minimum length = 1 Maximum length = 37
metricthreshold	<Double>	Read-write	Threshold to be used for that metric.
metricweight	<Double>	Read-write	The weight for the specified service metric with respect to others. Minimum value = 1 Maximum value = 100

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	The name of the service to which the monitor is bound. Minimum length = 1
servicegroupname	<String>	Read-write	The name of the service group to which the monitor is to be bound. Minimum length = 1
lrtmconf	<Integer>	Read-only	State of LRTM configuration on the monitor.
lrtmconfstr	<String>	Read-only	State of LRTM configuration on the monitor as STRING. Possible values = ENABLED, DISABLED
dynamicresponsetimeout	<Integer>	Read-only	Response timeout of the DRTM enabled monitor , calculated dynamically based on the history and current response time.
dynamicinterval	<Integer>	Read-only	Interval between monitoring probes for DRTM enabled monitor , calculated dynamically based monitor response time.
multimetrictable	<String[]>	Read-only	Metric table to which to bind metrics, to be used only for output purposes.
dup_state	<String>	Read-only	. Default value: ENABLED Possible values = ENABLED, DISABLED
dup_weight	<Double>	Read-only	. Default value: 1 Minimum value = 1 Maximum value = 100
weight	<Double>	Read-only	. Minimum value = 1 Maximum value = 100
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbmonitor":{
    "monitorname":<String_value>,
    "type":<String_value>,
    "action":<String_value>,
    "respcode":<String[]_value>,
    "httprequest":<String_value>,
    "rtsprequest":<String_value>,
    "customheaders":<String_value>,
    "maxforwards":<Double_value>,
    "sipmethod":<String_value>,
    "sipuri":<String_value>,
    "sipreguri":<String_value>,
    "send":<String_value>,
    "recv":<String_value>,
    "query":<String_value>,
    "querytype":<String_value>,
    "scriptname":<String_value>,
    "scriptargs":<String_value>,
    "dispatcherip":<String_value>,
    "dispatcherport":<Integer_value>,
    "username":<String_value>,
    "password":<String_value>,
    "secondarypassword":<String_value>,
    "logonpointname":<String_value>,
    "lasversion":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

"radkey":<String_value>,
"radnasid":<String_value>,
"radnasip":<String_value>,
"radaccounttype":<Double_value>,
"radframedip":<String_value>,
"radapn":<String_value>,
"radmsisdn":<String_value>,
"radaccountsession":<String_value>,
"lrtm":<String_value>,
"deviation":<Double_value>,
"units1":<String_value>,
"interval":<Integer_value>,
"units3":<String_value>,
"resptimeout":<Integer_value>,
"units4":<String_value>,
"resptimeoutthresh":<Double_value>,
"retries":<Integer_value>,
"failureretries":<Integer_value>,
"alertretries":<Integer_value>,
"successretries":<Integer_value>,
"downtime":<Integer_value>,
"units2":<String_value>,
"destip":<String_value>,
"destport":<Integer_value>,
"state":<String_value>,
"reverse":<String_value>,
"transparent":<String_value>,
"iptunnel":<String_value>,
"tos":<String_value>,
"tosid":<Double_value>,
"secure":<String_value>,
"validatecred":<String_value>,
"domain":<String_value>,
"ipaddress":<String[]_value>,
"group":<String_value>,
"filename":<String_value>,
"basedn":<String_value>,
"binddn":<String_value>,
"filter":<String_value>,
"attribute":<String_value>,
"database":<String_value>,
"oraclesid":<String_value>,
"sqlquery":<String_value>,
"evalrule":<String_value>,
"mssqlprotocolversion":<String_value>,
"Snmpoid":<String_value>,
"snmpcommunity":<String_value>,
"snmpthreshold":<String_value>,
"snmpversion":<String_value>,
"metrictable":<String_value>,
"application":<String_value>,
"sitepath":<String_value>,
"storename":<String_value>,
"storefrontacctservice":<String_value>,
"hostname":<String_value>,
"netprofile":<String_value>,
"originhost":<String_value>,
"originrealm":<String_value>,
"hostipaddress":<String_value>,
"vendorid":<Double_value>,
"productname":<String_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "firmwarerevision":<Double_value>,
    "authapplicationid":<Double[]_value>,
    "acctapplicationid":<Double[]_value>,
    "inbandsecurityid":<String_value>,
    "supportedvendorids":<Double[]_value>,
    "vendorspecificvendorid":<Double_value>,
    "vendorspecificauthapplicationids":<Double[]_value>,
    "vendorspecificacctapplicationids":<Double[]_value>,
    "kcdaccount":<String_value>,
    "storedb":<String_value>,
    "storefrontcheckbackendservices":<String_value>,
    "trofscode":<Double_value>,
    "trofsstring":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor/monitorname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbmonitor":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
"monitorname":<String_value>,
"type":<String_value>,
"action":<String_value>,
"respcode":<String[]_value>,
"httprequest":<String_value>,
"rtsprequest":<String_value>,
"customheaders":<String_value>,
"maxforwards":<Double_value>,
"sipmethod":<String_value>,
"sipreguri":<String_value>,
"sipuri":<String_value>,
"send":<String_value>,
"recv":<String_value>,
"query":<String_value>,
"querytype":<String_value>,
"username":<String_value>,
"password":<String_value>,
"secondarypassword":<String_value>,
"logonpointname":<String_value>,
"lasversion":<String_value>,
"radkey":<String_value>,
"radnasid":<String_value>,
"radnasip":<String_value>,
"radaccounttype":<Double_value>,
"radframedip":<String_value>,
"radapn":<String_value>,
"radmsisdn":<String_value>,
"radaccountsession":<String_value>,
"lrtm":<String_value>,
"deviation":<Double_value>,
"units1":<String_value>,
"scriptname":<String_value>,
"scriptargs":<String_value>,
"validatecred":<String_value>,
"domain":<String_value>,
"dispatcherip":<String_value>,
"dispatcherport":<Integer_value>,
"interval":<Integer_value>,
"units3":<String_value>,
"resptimeout":<Integer_value>,
"units4":<String_value>,
"resptimeoutthresh":<Double_value>,
"retries":<Integer_value>,
"failureretries":<Integer_value>,
"alertretries":<Integer_value>,
"successretries":<Integer_value>,
"downtime":<Integer_value>,
"units2":<String_value>,
"destip":<String_value>,
"destport":<Integer_value>,
"state":<String_value>,
"reverse":<String_value>,
"transparent":<String_value>,
"iptunnel":<String_value>,
"tos":<String_value>,
"tosid":<Double_value>,
"secure":<String_value>,
"ipaddress":<String[]_value>,
"group":<String_value>,
"filename":<String_value>,
"basedn":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "binddn":<String_value>,
    "filter":<String_value>,
    "attribute":<String_value>,
    "database":<String_value>,
    "oraclesid":<String_value>,
    "sqlquery":<String_value>,
    "evalrule":<String_value>,
    "Snmpoid":<String_value>,
    "snmpcommunity":<String_value>,
    "snmpthreshold":<String_value>,
    "snmpversion":<String_value>,
    "metrictable":<String_value>,
    "metric":<String_value>,
    "metricthreshold":<Double_value>,
    "metricweight":<Double_value>,
    "application":<String_value>,
    "sitepath":<String_value>,
    "storename":<String_value>,
    "storefrontacctservice":<String_value>,
    "storefrontcheckbackendservices":<String_value>,
    "hostname":<String_value>,
    "netprofile":<String_value>,
    "mssqlprotocolversion":<String_value>,
    "originhost":<String_value>,
    "originrealm":<String_value>,
    "hostipaddress":<String_value>,
    "vendorid":<Double_value>,
    "productname":<String_value>,
    "firmwarerevision":<Double_value>,
    "authapplicationid":<Double[]_value>,
    "acctapplicationid":<Double[]_value>,
    "inbandsecurityid":<String_value>,
    "supportedvendorids":<Double[]_value>,
    "vendorspecificvendorid":<Double_value>,
    "vendorspecificauthapplicationids":<Double[]_value>,
    "vendorspecificacctapplicationids":<Double[]_value>,
    "kcdaccount":<String_value>,
    "storedb":<String_value>,
    "trofscode":<Double_value>,
    "trofsstring":<String_value>
}}
```


Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor?**action=unset**

HTTP Method: POST

Request Headers:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbmonitor":{
    "monitorname":<String_value>,
    "type":<String_value>,
    "ipaddress":true,
    "scriptname":true,
    "destport":true,
    "netprofile":true,
    "action":true,
    "respcode":true,
    "httprequest":true,
    "rtsprequest":true,
    "customheaders":true,
    "maxforwards":true,
    "sipmethod":true,
    "sipreguri":true,
    "send":true,
    "recv":true,
    "query":true,
    "querytype":true,
    "username":true,
    "password":true,
    "secondarypassword":true,
    "logonpointname":true,
    "lasversion":true,
    "radkey":true,
    "radnasid":true,
    "radnasip":true,
    "radaccounttype":true,
    "radframedip":true,
    "radapn":true,
    "radmsisdn":true,
    "radaccountsession":true,
    "lrtm":true,
    "deviation":true,
    "scriptargs":true,
    "validatecred":true,
    "domain":true,
    "dispatcherip":true,
    "dispatcherport":true,
    "interval":true,
    "resptimeout":true,
    "resptimeoutthresh":true,
    "retries":true,
    "failureretries":true,
    "alertretries":true,
    "successretries":true,
    "downtime":true,
    "destip":true,
    "state":true,
    "reverse":true,
    "transparent":true,
    "iptunnel":true,
    "tos":true,
    "tosid":true,
    "secure":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "group":true,
    "filename":true,
    "basedn":true,
    "binddn":true,
    "filter":true,
    "attribute":true,
    "database":true,
    "oraclesid":true,
    "sqlquery":true,
    "Snmpoid":true,
    "snmpcommunity":true,
    "snmpthreshold":true,
    "snmpversion":true,
    "metrictable":true,
    "mssqlprotocolversion":true,
    "originhost":true,
    "originrealm":true,
    "hostipaddress":true,
    "vendorid":true,
    "productname":true,
    "firmwarerevision":true,
    "authapplicationid":true,
    "acctapplicationid":true,
    "inbandsecurityid":true,
    "supportedvendorids":true,
    "vendorspecificvendorid":true,
    "vendorspecificauthapplicationids":true,
    "vendorspecificacctapplicationids":true,
    "kcdaccount":true,
    "storedb":true,
    "trofscode":true,
    "trofsstring":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbmonitor":{
    "servicename":<String_value>,
    "servicegroupname":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "monitorname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbmonitor":{
    "servicename":<String_value>,
    "servicegroupname":<String_value>,
    "monitorname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/lbmonitor?**args=monitorname:<String_value>,type:<String_value>**

Use this query-parameter to get lbmonitor resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbmonitor?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbmonitor?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbmonitor resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lbmonitor?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbmonitor?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbmonitor resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbmonitor": [ {
monitorname:<String_value>,type:<String_value>      "interval":<Integer_value>,
      "units3":<String_value>,
      "resptimeout":<Integer_value>,
      "resptimeoutthresh":<Double_value>,
      "units4":<String_value>,
      "retries":<Integer_value>,
      "failureretries":<Integer_value>,
      "alertretries":<Integer_value>,
      "successretries":<Integer_value>,
      "downtime":<Integer_value>,
      "units2":<String_value>,
      "destip":<String_value>,
      "destport":<Integer_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
"state":<String_value>,
"reverse":<String_value>,
"transparent":<String_value>,
"iptunnel":<String_value>,
"tos":<String_value>,
"tosid":<Double_value>,
"secure":<String_value>,
"action":<String_value>,
"respcode":<String[]_value>,
"httprequest":<String_value>,
"rtsprequest":<String_value>,
"send":<String_value>,
"recv":<String_value>,
"query":<String_value>,
"querytype":<String_value>,
"username":<String_value>,
"password":<String_value>,
"secondarypassword":<String_value>,
"logonpointname":<String_value>,
"lasversion":<String_value>,
"validatecred":<String_value>,
"domain":<String_value>,
"radkey":<String_value>,
"radnasid":<String_value>,
"radnasip":<String_value>,
"radaccounttype":<Double_value>,
"radframedip":<String_value>,
"radapn":<String_value>,
"radmsisdn":<String_value>,
"radaccountsession":<String_value>,
"lrtm":<String_value>,
"lrtmconf":<Integer_value>,
"lrtmconfstr":<String_value>,
"deviation":<Double_value>,
"units1":<String_value>,
"dynamicresponsetimeout":<Integer_value>,
"dynamicinterval":<Integer_value>,
"scriptname":<String_value>,
"scriptargs":<String_value>,
"dispatcherip":<String_value>,
"dispatcherport":<Integer_value>,
"sipuri":<String_value>,
"sipmethod":<String_value>,
"maxforwards":<Double_value>,
"sipreguri":<String_value>,
"customheaders":<String_value>,
"ipaddress":<String[]_value>,
"group":<String_value>,
"filename":<String_value>,
"basedn":<String_value>,
"binddn":<String_value>,
"filter":<String_value>,
"attribute":<String_value>,
"database":<String_value>,
"oraclesid":<String_value>,
"sqlquery":<String_value>,
"evalrule":<String_value>,
"Snmpoid":<String_value>,
"snmpcommunity":<String_value>,
"snmpthreshold":<String_value>,
"snmpversion":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "metric":<String_value>,
    "metrictable":<String_value>,
    "multimetrictable":<String[]_value>,
    "metricthreshold":<Double_value>,
    "metricweight":<Double_value>,
    "application":<String_value>,
    "sitepath":<String_value>,
    "storename":<String_value>,
    "storefrontacctservice":<String_value>,
    "storefrontcheckbackendservices":<String_value>,
    "hostname":<String_value>,
    "netprofile":<String_value>,
    "mssqlprotocolversion":<String_value>,
    "originhost":<String_value>,
    "originrealm":<String_value>,
    "hostipaddress":<String_value>,
    "vendorid":<Double_value>,
    "productname":<String_value>,
    "firmwarerevision":<Double_value>,
    "authapplicationid":<Double[]_value>,
    "acctapplicationid":<Double[]_value>,
    "inbandsecurityid":<String_value>,
    "supportedvendorids":<Double[]_value>,
    "vendorspecificvendorid":<Double_value>,
    "vendorspecificauthapplicationids":<Double[]_value>,
    "vendorspecificacctapplicationids":<Double[]_value>,
    "servicename":<String_value>,
    "dup_state":<String_value>,
    "dup_weight":<Double_value>,
    "servicegroupname":<String_value>,
    "weight":<Double_value>,
    "kcdaccount":<String_value>,
    "storedb":<String_value>,
    "trofscode":<Double_value>,
    "trofsstring":<String_value>
}]}
```


---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor/monitorname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbmonitor/monitorname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lbmonitor/monitorname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbmonitor": [ {
monitorname:<String_value>,type:<String_value>      "interval":<Integer_value>,
      "units3":<String_value>,
      "resptimeout":<Integer_value>,
      "resptimeoutthresh":<Double_value>,
      "units4":<String_value>,
      "retries":<Integer_value>,
      "failureretries":<Integer_value>,
      "alertretries":<Integer_value>,
      "successretries":<Integer_value>,
      "downtime":<Integer_value>,
      "units2":<String_value>,
      "destip":<String_value>,
      "destport":<Integer_value>,
      "state":<String_value>,
      "reverse":<String_value>,
      "transparent":<String_value>,
      "iptunnel":<String_value>,
      "tos":<String_value>,
      "tosid":<Double_value>,
      "secure":<String_value>,
      "action":<String_value>,
      "respcode":<String[]_value>,
      "httprequest":<String_value>,
      "rtsprequest":<String_value>,
      "send":<String_value>,
      "recv":<String_value>,
      "query":<String_value>,
      "querytype":<String_value>,
      "username":<String_value>,
      "password":<String_value>,
      "secondarypassword":<String_value>,
      "logonpointname":<String_value>,
      "lasversion":<String_value>,
      "validatecred":<String_value>,
      "domain":<String_value>,
      "radkey":<String_value>,
      "radnasid":<String_value>,
      "radnasip":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
"radaccounttype":<Double_value>,
"radframedip":<String_value>,
"radapn":<String_value>,
"radmsisdn":<String_value>,
"radaccountsession":<String_value>,
"lrtm":<String_value>,
"lrtmconf":<Integer_value>,
"lrtmconfstr":<String_value>,
"deviation":<Double_value>,
"units1":<String_value>,
"dynamicresponsetimeout":<Integer_value>,
"dynamicinterval":<Integer_value>,
"scriptname":<String_value>,
"scriptargs":<String_value>,
"dispatcherip":<String_value>,
"dispatcherport":<Integer_value>,
"sipuri":<String_value>,
"sipmethod":<String_value>,
"maxforwards":<Double_value>,
"sipreguri":<String_value>,
"customheaders":<String_value>,
"ipaddress":<String[]_value>,
"group":<String_value>,
"filename":<String_value>,
"basedn":<String_value>,
"binddn":<String_value>,
"filter":<String_value>,
"attribute":<String_value>,
"database":<String_value>,
"oraclesid":<String_value>,
"sqlquery":<String_value>,
"evalrule":<String_value>,
"Snmpoid":<String_value>,
"snmpcommunity":<String_value>,
"snmpthreshold":<String_value>,
"snmpversion":<String_value>,
"metric":<String_value>,
"metrictable":<String_value>,
"multimetrictable":<String[]_value>,
"metricthreshold":<Double_value>,
"metricweight":<Double_value>,
"application":<String_value>,
"sitepath":<String_value>,
"storename":<String_value>,
"storefrontacctservice":<String_value>,
"storefrontcheckbackendservices":<String_value>,
"hostname":<String_value>,
"netprofile":<String_value>,
"mssqlprotocolversion":<String_value>,
"originhost":<String_value>,
"originrealm":<String_value>,
"hostipaddress":<String_value>,
"vendorid":<Double_value>,
"productname":<String_value>,
"firmwarerevision":<Double_value>,
"authapplicationid":<Double[]_value>,
"acctapplicationid":<Double[]_value>,
"inbandsecurityid":<String_value>,
"supportedvendorids":<Double[]_value>,
"vendorspecificvendorid":<Double_value>,
"vendorspecificauthapplicationids":<Double[]_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "vendorspecificacctapplicationids":<Double[]_value>,
    "servicename":<String_value>,
    "dup_state":<String_value>,
    "dup_weight":<Double_value>,
    "servicegroupname":<String_value>,
    "weight":<Double_value>,
    "kcdaccount":<String_value>,
    "storedb":<String_value>,
    "trofscode":<Double_value>,
    "trofsstring":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbmonitor": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmonitor_binding

May 20, 2016

Binding object which returns the resources bound to lbmonitor.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
monitorname	<String>	Read-write	Name of the monitor. Minimum length = 1
lbmonitor_metric_binding	<lbmonitor_metric_binding[]>	Read-only	metric that can be bound to lbmonitor.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor_binding/monitorname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lbmonitor_binding": [ {
      "monitorname":<String_value>,
      "lbmonitor_metric_binding":<lbmonitor_metric_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmonitor_metric_binding

Jul 07, 2017

Binding object showing the metric that can be bound to lbmonitor.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
metric	<String>	Read-write	Metric name in the metric table, whose setting is changed. A value zero disables the metric and it will not be used for load calculation. Minimum length = 1 Maximum length = 37
metricthreshold	<Double>	Read-write	Threshold to be used for that metric.
metricweight	<Double>	Read-write	The weight for the specified service metric with respect to others. Minimum value = 1 Maximum value = 100
monitorname	<String>	Read-write	Name of the monitor. Minimum length = 1
metric_unit	<String>	Read-only	Giving the unit of the metric. Possible values = Bytes/s, ms, pkts/s, users
metrictable	<String>	Read-only	Metric table to which to bind metrics. Minimum length = 1 Maximum length = 99
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbmonitor_metric_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbmonitor_metric_binding":{
    "monitorname":<String_value>,
    "metric":<String_value>,
    "metricthreshold":<Double_value>,
    "metricweight":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbmonitor_metric_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbmonitor_metric_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "lbmonitor_metric_binding": [ {
    "metric":<String_value>,
    "metricthreshold":<Double_value>,
    "metricweight":<Double_value>,
    "monitorname":<String_value>,
    "metric_unit":<String_value>,
    "metrictable":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor_metric_binding/monitorname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbmonitor_metric_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmonitor_servicegroup_binding

May 20, 2016

Binding object showing the servicegroup that can be bound to lbmonitor.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
dup_state	<String>	Read-write	State of the monitor. The state setting for a monitor of a given type affects all monitors of that type. For example, if an HTTP monitor is enabled, all HTTP monitors on the appliance are (or remain) enabled. If an HTTP monitor is disabled, all HTTP monitors on the appliance are disabled. Default value: ENABLED Possible values = ENABLED, DISABLED
servicename	<String>	Read-write	Name of the service or service group. Minimum length = 1
state	<String>	Read-write	State of the monitor. The state setting for a monitor of a given type affects all monitors of that type. For example, if an HTTP monitor is enabled, all HTTP monitors on the appliance are (or remain) enabled. If an HTTP monitor is disabled, all HTTP monitors on the appliance are disabled. Default value: ENABLED Possible values = ENABLED, DISABLED
dup_weight	<Double>	Read-write	Weight to assign to the binding between the monitor and service. Default value: 1 Minimum value = 1 Maximum value = 100
monitorname	<String>	Read-write	Name of the monitor. Minimum length = 1
weight	<Double>	Read-write	Weight to assign to the binding between the monitor and service. Minimum value = 1 Maximum value = 100

Operations (click to see Properties)

ADD: | DELETE:

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbmonitor_servicegroup_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbmonitor_servicegroup_binding":{
    "monitorname":<String_value>,
    "servicename":<String_value>,
    "dup_state":<String_value>,
    "dup_weight":<Double_value>,
    "servicegroupname":<String_value>,
    "state":<String_value>,
    "weight":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor_servicegroup_binding/monitorname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbparameter

May 20, 2016

Configuration for LB parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
httponlycookieflag	<String>	Read-write	Include the HttpOnly attribute in persistence cookies. The HttpOnly attribute limits the scope of a cookie to HTTP requests and helps mitigate the risk of cross-site scripting attacks. Default value: ENABLED Possible values = ENABLED, DISABLED
usesecuredpersistencecookie	<String>	Read-write	Encode persistence cookie values using SHA2 hash. Default value: DISABLED Possible values = ENABLED, DISABLED
cookiepassphrase	<String>	Read-write	Use this parameter to specify the passphrase used to generate secured persistence cookie value. It specifies the passphrase with a maximum of 31 characters.
consolidatedlconn	<String>	Read-write	To find the service with the fewest connections, the virtual server uses the consolidated connection statistics from all the packet engines. The NO setting allows consideration of only the number of connections on the packet engine that received the new connection. Default value: YES Possible values = YES, NO
useportforhashlb	<String>	Read-write	Include the port number of the service when creating a hash for hash based load balancing methods. With the NO setting, only the IP address of the service is considered when creating a hash. Default value: YES Possible values = YES, NO
preferdirectroute	<String>	Read-write	Perform route lookup for traffic received by the NetScaler appliance, and forward the traffic according to configured routes. Do not set this parameter if you want a wildcard virtual server to direct packets received by the appliance to an intermediary device, such as a firewall, even if their destination is directly connected to the appliance. Route lookup is performed after the packets have been processed and returned by the intermediary device. Default value: YES Possible values = YES, NO
			Number of requests, per service, for which to apply the round robin load balancing method before switching to the configured load balancing

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			specified number of requests is distributed, the NetScaler appliance is said to be implementing the slow start mode (or startup round robin).
startuprrfactor	<Double>	Read-write	Implemented for a virtual server when one of the following is true: * The virtual server is newly created. * One or more services are newly bound to the virtual server. * One or more services bound to the virtual server are enabled. * The load balancing method is changed. This parameter applies to all the load balancing virtual servers configured on the NetScaler appliance, except for those virtual servers for which the virtual server-level slow start parameters (New Service Startup Request Rate and Increment Interval) are configured. If the global slow start parameter and the slow start parameters for a given virtual server are not set, the appliance implements a default slow start for the virtual server, as follows: * For a newly configured virtual server, the appliance implements slow start for the first 100 requests received by the virtual server. * For an existing virtual server, if one or more services are newly bound or newly enabled, or if the load balancing method is changed, the appliance dynamically computes the number of requests for which to implement startup round robin. It obtains this number by multiplying the request rate by the number of bound services (it includes services that are marked as DOWN). For example, if the current request rate is 20 requests/s and ten services are bound to the virtual server, the appliance performs startup round robin for 200 requests. Not applicable to a virtual server for which a hash based load balancing method is configured.
monitorskipmaxclient	<String>	Read-write	When a monitor initiates a connection to a service, do not check to determine whether the number of connections to the service has reached the limit specified by the service's Max Clients setting. Enables monitoring to continue even if the service has reached its connection limit. Default value: DISABLED Possible values = ENABLED, DISABLED
monitorconnectionclose	<String>	Read-write	Close monitoring connections by sending the service a connection termination message with the specified bit set. Default value: FIN Possible values = RESET, FIN
vserverspecificmac	<String>	Read-write	Allow a MAC-mode virtual server to accept traffic returned by an intermediary device, such as a firewall, to which the traffic was previously forwarded by another MAC-mode virtual server. The second virtual server can then distribute that traffic across the destination server farm. Also useful when load balancing Branch Repeater appliances. Note: The second virtual server can also send the traffic to another set of intermediary devices, such as another set of firewalls. If necessary, you can configure multiple MAC-mode virtual servers to pass traffic successively through multiple sets of intermediary devices. Default value: DISABLED Possible values = ENABLED, DISABLED
			This option is used to retain the original state of service or servicegroup member when an enable server command is issued.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = ON, OFF
sessionsthreshold	<Double>	Read-only	This option is used to get the upper-limit on the number of persistent sessions set by the administrator for this system. Minimum value = 1
builtin	<String[]>	Read-only	. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lbparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"lbparameter":{
    "httponlycookieflag":<String_value>,
    "usesecuredpersistencecookie":<String_value>,
    "cookiepassphrase":<String_value>,
    "consolidatedlconn":<String_value>,
    "useportforhashlb":<String_value>,
    "preferdirectroute":<String_value>,
    "startuprrfactor":<Double_value>,
    "monitorskipmaxclient":<String_value>,
    "monitorconnectionclose":<String_value>,
    "vserverspecificmac":<String_value>,
    "retainservicestate":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lbparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbparameter":{
    "httponlycookieflag":true,
    "usesecuredpersistencecookie":true,
    "cookiepassphrase":true,
    "consolidatedlconn":true,
    "useportforhashlb":true,
    "preferdirectroute":true,
    "startuprrfactor":true,
    "monitorskipmaxclient":true,
    "monitorconnectionclose":true,
    "vserverspecificmac":true,
    "retainservicestate":true
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbparameter": [ {
      "httponlycookieflag":<String_value>,
      "usesecuredpersistencecookie":<String_value>,
      "cookiepassphrase":<String_value>,
      "consolidatedlconn":<String_value>,
      "useportforhashlb":<String_value>,
      "preferdirectroute":<String_value>,
      "startuprrfactor":<Double_value>,
      "monitorskipmaxclient":<String_value>,
      "monitorconnectionclose":<String_value>,
      "vserverspecificmac":<String_value>,
      "sessionsthreshold":<Double_value>,
      "builtin":<String[]_value>,
      "retainservicestate":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbmonitor_service_binding

May 20, 2016

Binding object showing the service that can be bound to lbmonitor.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
dup_state	<String>	Read-write	State of the monitor. The state setting for a monitor of a given type affects all monitors of that type. For example, if an HTTP monitor is enabled, all HTTP monitors on the appliance are (or remain) enabled. If an HTTP monitor is disabled, all HTTP monitors on the appliance are disabled. Default value: ENABLED Possible values = ENABLED, DISABLED
servicename	<String>	Read-write	Name of the service or service group. Minimum length = 1
state	<String>	Read-write	State of the monitor. The state setting for a monitor of a given type affects all monitors of that type. For example, if an HTTP monitor is enabled, all HTTP monitors on the appliance are (or remain) enabled. If an HTTP monitor is disabled, all HTTP monitors on the appliance are disabled. Default value: ENABLED Possible values = ENABLED, DISABLED
dup_weight	<Double>	Read-write	Weight to assign to the binding between the monitor and service. Default value: 1 Minimum value = 1 Maximum value = 100
monitorname	<String>	Read-write	Name of the monitor. Minimum length = 1
weight	<Double>	Read-write	Weight to assign to the binding between the monitor and service. Minimum value = 1 Maximum value = 100

Operations (click to see Properties)

ADD: | DELETE:

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbmonitor_service_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbmonitor_service_binding":{
    "monitorname":<String_value>,
    "servicename":<String_value>,
    "dup_state":<String_value>,
    "dup_weight":<Double_value>,
    "servicegroupname":<String_value>,
    "state":<String_value>,
    "weight":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbmonitor_service_binding/monitorname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbpersistentsessions

Jul 07, 2017

Configuration for persistence session resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<String>	Read-write	The name of the virtual server.
persistenceparameter	<String>	Read-write	The persistence parameter whose persistence sessions are to be flushed.
type	<Double>	Read-only	Type of Persistence.
typestring	<String>	Read-only	Type of Persistence as String.
srcip	<String>	Read-only	SOURCE IP.
srcipv6	<String>	Read-only	SOURCE IPv6 ADDRESS.
destip	<String>	Read-only	DESTINATION IP.
destipv6	<String>	Read-only	DESTINATION IPv6 ADDRESS.
flags	<Boolean>	Read-only	IPv6 FLAGS.
destport	<Integer>	Read-only	Destination port. Range 1 - 65535
vservername	<String>	Read-only	Virtual server name.
timeout	<Double>	Read-only	Persistent Session timeout.
referencecount	<Double>	Read-only	Reference Count.
persistenceparam	<String>	Read-only	Specific persistence information . Callid in case of SIP_CALLID persistence entry

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sitepersparam	<String>	Read-only	This parameter that is selected incase of gslb service.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

CLEAR | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/lbpersistentsessions?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"lbpersistentsessions":{
    "vserver":<String_value>,
    "persistenceparameter":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbpersistentsessions

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/lbpersistentsessions?**args=vserver:<String_value>**

Use this query-parameter to get lbpersistentsessions resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbpersistentsessions?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbpersistentsessions?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbpersistentsessions resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lbpersistentsessions?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbpersistentsessions?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbpersistentsessions resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbpersistentsessions": [ {
vserver:<String_value>      "type":<Double_value>,
      "typestring":<String_value>,
      "srcip":<String_value>,
      "srcipv6":<String_value>,
      "destip":<String_value>,
      "destipv6":<String_value>,
      "flags":<Boolean_value>,
      "destport":<Integer_value>,
      "vservername":<String_value>,
      "timeout":<Double_value>,
      "referencecount":<Double_value>,
      "sipcallid":<String_value>,
      "persistenceparam":<String_value>,
      "cnamepersparam":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbpersistentsessions?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{ "lbpersistentsessions": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbroute

May 20, 2016

Configuration for LB route resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
network	<String>	Read-write	The IP address of the network to which the route belongs.
netmask	<String>	Read-write	The netmask to which the route belongs.
gatewayname	<String>	Read-write	The name of the route. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Default value: 0 Minimum value = 0 Maximum value = 4094
flags	<String>	Read-only	State of the configured gateway. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lbroute

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbroute":{
    "network":<String_value>,
    "netmask":<String_value>,
    "gatewayname":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lbroute/network_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbroute

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbroute?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbroute?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbroute resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lbroute?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbroute?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbroute resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbroute": [ {
      "network":<String_value>,
      "netmask":<String_value>,
      "td":<Double_value>,
      "gatewayname":<String_value>,
      "flags":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbroute?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbroute": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbroute6

May 20, 2016

Configuration for LB route6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
network	<String>	Read-write	The destination network.
gatewayname	<String>	Read-write	The name of the route. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Default value: 0 Minimum value = 0 Maximum value = 4094
flags	<String>	Read-only	State of the configured gateway. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lbroute6

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbroute6":{
    "network":<String_value>,
    "gatewayname":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lbroute6/network_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbroute6

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbroute6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbroute6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbroute6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lbroute6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbroute6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbroute6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "lbroute6": [ {
      "network":<String_value>,
      "td":<Double_value>,
      "gatewayname":<String_value>,
      "flags":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbroute6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbroute6": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbsipparameters

May 20, 2016

Configuration for SIP parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
rnatsrcport	<Integer>	Read-write	Port number with which to match the source port in server-initiated SIP traffic. The rport parameter is added, without a value, to SIP packets that have a matching source port number, and CALL-ID based persistence is implemented for the responses received by the virtual server. Default value: 0
rnatdstport	<Integer>	Read-write	Port number with which to match the destination port in server-initiated SIP traffic. The rport parameter is added, without a value, to SIP packets that have a matching destination port number, and CALL-ID based persistence is implemented for the responses received by the virtual server. Default value: 0
retrydur	<Integer>	Read-write	Time, in seconds, for which a client must wait before initiating a connection after receiving a 503 Service Unavailable response from the SIP server. The time value is sent in the "Retry-After" header in the 503 response. Default value: 120 Minimum value = 1
addrportvip	<String>	Read-write	Add the rport parameter to the VIA headers of SIP requests that virtual servers receive from clients or servers. Default value: ENABLED Possible values = ENABLED, DISABLED
sip503ratethreshold	<Double>	Read-write	Maximum number of 503 Service Unavailable responses to generate, once every 10 milliseconds, when a SIP virtual server becomes unavailable. Default value: 100
rnatsecuresrcport	<Integer>	Read-write	Port number with which to match the source port in server-initiated SIP over SSL traffic. The rport parameter is added, without a value, to SIP packets that have a matching source port number, and CALL-ID based persistence is implemented for the responses received by the virtual server. Default value: 0 Range 1 - 65535
rnatsecuredstport	<Integer>	Read-write	Port number with which to match the destination port in server-initiated SIP over SSL traffic. The rport parameter is added, without a value, to SIP packets that have a matching destination port number, and CALL-ID based persistence is implemented for the responses received by the virtual server. Default value: 0

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lbsipparameters

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbsipparameters":{
    "rnatsrcport":<Integer_value>,
    "rnatdstport":<Integer_value>,
    "retrydur":<Integer_value>,
    "addrportvip":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "sip503ratethreshold":<Double_value>,
    "rnatsecuresrcport":<Integer_value>,
    "rnatsecuredstport":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lbsipparameters?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbsipparameters":{
    "rnatsrcport":true,
    "rnatdstport":true,
    "retrydur":true,
    "addrportvip":true,
    "sip503ratethreshold":true,
    "rnatsecuresrcport":true,
    "rnatsecuredstport":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbsipparameters

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbsipparameters": [ {
      "rnatsrcport":<Integer_value>,
      "rnatdstport":<Integer_value>,
      "retrydur":<Integer_value>,
      "addrportvip":<String_value>,
      "sip503ratethreshold":<Double_value>,
      "rnatsecuresrcport":<Integer_value>,
      "rnatsecuredstport":<Integer_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver

May 20, 2016

Configuration for Load Balancing Virtual Server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
servicetype	<String>	Read-write	Protocol used by the service (also called the service type). Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP, SYSLOGUDP, FIX
ipv46	<String>	Read-write	IPv4 or IPv6 address to assign to the virtual server.
ippattern	<String>	Read-write	IP address pattern, in dotted decimal notation, for identifying packets to be accepted by the virtual server. The IP Mask parameter specifies which part of the destination IP address is matched against the pattern. Mutually exclusive with the IP Address parameter. For example, if the IP pattern assigned to the virtual server is 198.51.100.0 and the IP mask is 255.255.240.0 (a forward mask), the first 20 bits in the destination IP addresses are matched with the first 20 bits in the pattern. The virtual server accepts requests with IP addresses that range from 198.51.96.1 to 198.51.111.254. You can also use a pattern such as 0.0.2.2 and a mask such as 0.0.255.255 (a reverse mask). If a destination IP address matches more than one IP pattern, the pattern with the longest match is selected, and the associated virtual server processes the request. For example, if virtual servers vs1 and vs2 have the same IP pattern, 0.0.100.128, but different IP masks of 0.0.255.255 and 0.0.224.255, a destination IP address of 198.51.100.128 has the longest match with the IP pattern of vs1. If a destination IP address matches two or more virtual servers to the same extent, the request is processed by the virtual server whose port number matches the port number in the request.
ipmask	<String>	Read-write	IP mask, in dotted decimal notation, for the IP Pattern parameter. Can have leading or trailing non-zero octets (for example, 255.255.240.0 or 0.0.255.255). Accordingly, the mask specifies whether the first n bits or the last n bits of the destination IP address in a client request are to be matched with the corresponding bits in the IP pattern. The former is called a forward mask. The latter is called a reverse mask.
port	<Integer>	Read-write	Port number for the virtual server. Range 1 - 65535
			Number of IP addresses that the appliance must generate and assign to the virtual server. The virtual server then functions as a network virtual server, accepting traffic

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			as follows: For a range of n, the last octet of the address specified by the IP Address parameter increments n-1 times.
range	<Double>	Read-write	* If the last octet exceeds 255, it rolls over to 0 and the third octet increments by 1. Note: The Range parameter assigns multiple IP addresses to one virtual server. To generate an array of virtual servers, each of which owns only one IP address, use brackets in the IP Address and Name parameters to specify the range. For example: add lb vserver my_vserver[1-3] HTTP 192.0.2.[1-3] 80. Default value: 1 Minimum value = 1 Maximum value = 254
persistencetype	<String>	Read-write	Type of persistence for the virtual server. Available settings function as follows: * SOURCEIP - Connections from the same client IP address belong to the same persistence session. * COOKIEINSERT - Connections that have the same HTTP Cookie, inserted by a Set-Cookie directive from a server, belong to the same persistence session. * SSLSESSION - Connections that have the same SSL Session ID belong to the same persistence session. * CUSTOMSERVERID - Connections with the same server ID form part of the same session. For this persistence type, set the Server ID (CustomServerID) parameter for each service and configure the Rule parameter to identify the server ID in a request. * RULE - All connections that match a user defined rule belong to the same persistence session. * URLPASSIVE - Requests that have the same server ID in the URL query belong to the same persistence session. The server ID is the hexadecimal representation of the IP address and port of the service to which the request must be forwarded. This persistence type requires a rule to identify the server ID in the request. * DESTIP - Connections to the same destination IP address belong to the same persistence session. * SRCIPDESTIP - Connections that have the same source IP address and destination IP address belong to the same persistence session. * CALLID - Connections that have the same CALL-ID SIP header belong to the same persistence session. * RTSPSID - Connections that have the same RTSP Session ID belong to the same persistence session. * FIXSESSION - Connections that have the same SenderCompID and TargetCompID values belong to the same persistence session. Possible values = SOURCEIP, COOKIEINSERT, SSLSESSION, RULE, URLPASSIVE, CUSTOMSERVERID, DESTIP, SRCIPDESTIP, CALLID, RTSPSID, DIAMETER, FIXSESSION, NONE
timeout	<Double>	Read-write	Time period for which a persistence session is in effect. Default value: 2 Minimum value = 0 Maximum value = 1440
persistencebackup	<String>	Read-write	Backup persistence type for the virtual server. Becomes operational if the primary persistence mechanism fails. Possible values = SOURCEIP, NONE
backuppersistencetimeout	<Double>	Read-write	Time period for which backup persistence is in effect. Default value: 2 Minimum value = 2 Maximum value = 1440
			Load balancing method. The available settings function as follows: * ROUNDROBIN - Distribute requests in rotation, regardless of the load. Weights can be assigned to services to enforce weighted round robin distribution. * LEASTCONNECTION (default) - Select the service with the fewest connections. * LEASTRESPONSETIME - Select the service with the lowest average response time. * LEASTBANDWIDTH - Select the service currently handling the least traffic.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			per second. * CUSTOMLOAD - Base service selection on the SNMP metrics obtained by custom load monitors.
lbmethod	<String>	Read-write	* LRTM - Select the service with the lowest response time. Response times are learned through monitoring probes. This method also takes the number of active connections into account. Also available are a number of hashing methods, in which the appliance extracts a predetermined portion of the request, creates a hash of the portion, and then checks whether any previous requests had the same hash value. If it finds a match, it forwards the request to the service that served those previous requests. Following are the hashing methods: * URLHASH - Create a hash of the request URL (or part of the URL). * DOMAINHASH - Create a hash of the domain name in the request (or part of the domain name). The domain name is taken from either the URL or the Host header. If the domain name appears in both locations, the URL is preferred. If the request does not contain a domain name, the load balancing method defaults to LEASTCONNECTION. * DESTINATIONIPHASH - Create a hash of the destination IP address in the IP header. * SOURCEIPHASH - Create a hash of the source IP address in the IP header. * TOKEN - Extract a token from the request, create a hash of the token, and then select the service to which any previous requests with the same token hash value were sent. * SRCIPDESTIPHASH - Create a hash of the string obtained by concatenating the source IP address and destination IP address in the IP header. * SRCIPSRCPORTHASH - Create a hash of the source IP address and source port in the IP header. * CALLIDHASH - Create a hash of the SIP Call-ID header. Default value: LEASTCONNECTION Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, URLHASH, DOMAINHASH, DESTINATIONIPHASH, SOURCEIPHASH, SRCIPDESTIPHASH, LEASTBANDWIDTH, LEASTPACKETS, TOKEN, SRCIPSRCPORTHASH, LRTM, CALLIDHASH, CUSTOMLOAD, LEASTREQUEST, AUDITLOGHASH, STATICPROXIMITY
hashlength	<Double>	Read-write	Number of bytes to consider for the hash value used in the URLHASH and DOMAINHASH load balancing methods. Default value: 80 Minimum value = 1 Maximum value = 4096
netmask	<String>	Read-write	IPv4 subnet mask to apply to the destination IP address or source IP address when the load balancing method is DESTINATIONIPHASH or SOURCEIPHASH. Minimum length = 1
v6netmasklen	<Double>	Read-write	Number of bits to consider in an IPv6 destination or source IP address, for creating the hash that is required by the DESTINATIONIPHASH and SOURCEIPHASH load balancing methods. Default value: 128 Minimum value = 1 Maximum value = 128
backuplbmethod	<String>	Read-write	Backup load balancing method. Becomes operational if the primary load balancing method fails or cannot be used. Valid only if the primary method is based on static proximity. Default value: ROUNDROBIN Possible values = ROUNDROBIN, LEASTCONNECTION, LEASTRESPONSETIME, SOURCEIPHASH, LEASTBANDWIDTH, LEASTPACKETS, CUSTOMLOAD
cookiename	<String>	Read-write	Use this parameter to specify the cookie name for COOKIE peristence type. It specifies the name of cookie with a maximum of 32 characters. If not specified, cookie name is internally generated.
			Expression, or name of a named expression, against which traffic is evaluated.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
rule	<String>	Read-write	Note: Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks. Default value: "none"
listenpolicy	<String>	Read-write	Default syntax expression identifying traffic accepted by the virtual server. Can be either an expression (for example, CLIENT.IP.DST.IN_SUBNET(192.0.2.0/24) or the name of a named expression. In the above example, the virtual server accepts all requests whose destination IP address is in the 192.0.2.0/24 subnet. Default value: "NONE"
listenpriority	<Double>	Read-write	Integer specifying the priority of the listen policy. A higher number specifies a lower priority. If a request matches the listen policies of more than one virtual server the virtual server whose listen policy has the highest priority (the lowest priority number) accepts the request. Default value: 101 Minimum value = 0 Maximum value = 101
resrule	<String>	Read-write	Default syntax expression specifying which part of a server's response to use for creating rule based persistence sessions (persistence type RULE). Can be either an expression or the name of a named expression. Example: HTTP.RES.HEADER("setcookie").VALUE(0).TYPECAST_NVLIST_T('=',';').VALUE("server1"). Default value: "none"
persistmask	<String>	Read-write	Persistence mask for IP based persistence types, for IPv4 virtual servers. Minimum length = 1
v6persistmasklen	<Double>	Read-write	Persistence mask for IP based persistence types, for IPv6 virtual servers. Default value: 128 Minimum value = 1 Maximum value = 128
pq	<String>	Read-write	Use priority queuing on the virtual server. based persistence types, for IPv6 virtual servers. Default value: OFF Possible values = ON, OFF
sc	<String>	Read-write	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
rtspnat	<String>	Read-write	Use network address translation (NAT) for RTSP data connections. Default value: OFF Possible values = ON, OFF
			Redirection mode for load balancing. Available settings function as follows: * IP - Before forwarding a request to a server, change the destination IP address to the server's IP address. * MAC - Before forwarding a request to a server, change the destination MAC address to the server's MAC address. The destination IP address is not changed. MAC-based

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
m	<String>	Read-write	* IPTUNNEL - Perform IP-in-IP encapsulation for client IP packets. In the outer IP headers, set the destination IP address to the IP address of the server and the source IP address to the subnet IP (SNIP). The client IP packets are not modified. Applicable to both IPv4 and IPv6 packets. * TOS - Encode the virtual server's TOS ID in the TOS field of the IP header. You can use either the IPTUNNEL or the TOS option to implement Direct Server Return (DSR). Default value: IP Possible values = IP, MAC, IPTUNNEL, TOS
tosid	<Double>	Read-write	TOS ID of the virtual server. Applicable only when the load balancing redirection mode is set to TOS. Minimum value = 1 Maximum value = 63
datalength	<Double>	Read-write	Length of the token to be extracted from the data segment of an incoming packet, for use in the token method of load balancing. The length of the token, specified in bytes, must not be greater than 24 KB. Applicable to virtual servers of type TCP. Minimum value = 1 Maximum value = 100
dataoffset	<Double>	Read-write	Offset to be considered when extracting a token from the TCP payload. Applicable to virtual servers, of type TCP, using the token method of load balancing. Must be within the first 24 KB of the TCP payload. Minimum value = 0 Maximum value = 25400
sessionless	<String>	Read-write	Perform load balancing on a per-packet basis, without establishing sessions. Recommended for load balancing of intrusion detection system (IDS) servers and scenarios involving direct server return (DSR), where session information is unnecessary. Default value: DISABLED Possible values = ENABLED, DISABLED
state	<String>	Read-write	State of the load balancing virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
connfailover	<String>	Read-write	Mode in which the connection failover feature must operate for the virtual server. After a failover, established TCP connections and UDP packet flows are kept active and resumed on the secondary appliance. Clients remain connected to the same servers. Available settings function as follows: * STATEFUL - The primary appliance shares state information with the secondary appliance, in real time, resulting in some runtime processing overhead. * STATELESS - State information is not shared, and the new primary appliance tries to re-create the packet flow on the basis of the information contained in the packets it receives. * DISABLED - Connection failover does not occur. Default value: DISABLED Possible values = DISABLED, STATEFUL, STATELESS
redirurl	<String>	Read-write	URL to which to redirect traffic if the virtual server becomes unavailable. WARNING! Make sure that the domain in the URL does not match the domain specified for a content switching policy. If it does, requests are continuously redirected to the unavailable virtual server. Minimum length = 1
cacheable	<String>	Read-write	Route cacheable requests to a cache redirection virtual server. The load balancing virtual server can forward requests only to a transparent cache redirection virtual server that has an IP address and port combination of *:80, so such a cache redirection virtual server must be configured on the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = YES, NO
clttimeout	<Double>	Read-write	Idle time, in seconds, after which a client connection is terminated. Minimum value = 0 Maximum value = 31536000
somethod	<String>	Read-write	Type of threshold that, when exceeded, triggers spillover. Available settings function as follows: * CONNECTION - Spillover occurs when the number of client connections exceeds the threshold. * DYNAMICCONNECTION - Spillover occurs when the number of client connections at the virtual server exceeds the sum of the maximum client (Max Clients) settings for bound services. Do not specify a spillover threshold for this setting, because the threshold is implied by the Max Clients settings of bound services. * BANDWIDTH - Spillover occurs when the bandwidth consumed by the virtual server's incoming and outgoing traffic exceeds the threshold. * HEALTH - Spillover occurs when the percentage of weights of the services that are UP drops below the threshold. For example, if services svc1, svc2, and svc3 are bound to a virtual server, with weights 1, 2, and 3, and the spillover threshold is 50%, spillover occurs if svc1 and svc3 or svc2 and svc3 transition to DOWN. * NONE - Spillover does not occur. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sopersistence	<String>	Read-write	If spillover occurs, maintain source IP address based persistence for both primary and backup virtual servers. Default value: DISABLED Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-write	Timeout for spillover persistence, in minutes. Default value: 2 Minimum value = 2 Maximum value = 1440
healththreshold	<Double>	Read-write	Threshold in percent of active services below which vserver state is made down. If this threshold is 0, vserver state will be up even if one bound service is up. Default value: 0 Minimum value = 0 Maximum value = 100
sothreshold	<Double>	Read-write	Threshold at which spillover occurs. Specify an integer for the CONNECTION spillover method, a bandwidth value in kilobits per second for the BANDWIDTH method (do not enter the units), or a percentage for the HEALTH method (do not enter the percentage symbol). Minimum value = 1 Maximum value = 4294967287
sobackupaction	<String>	Read-write	Action to be performed if spillover is to take effect, but no backup chain to spillover is usable or exists. Possible values = DROP, ACCEPT, REDIRECT
redirectportrewrite	<String>	Read-write	Rewrite the port and change the protocol to ensure successful HTTP redirects from services. Default value: DISABLED Possible values = ENABLED, DISABLED
downstateflush	<String>	Read-write	Flush all active transactions associated with a virtual server whose state transitions from UP to DOWN. Do not enable this option for applications that must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED
			Name of the backup virtual server to which to forward requests if the primary virtual

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
disableprimaryondown	<String>	Read-write	If the primary virtual server goes down, do not allow it to return to primary status until manually enabled. Default value: DISABLED Possible values = ENABLED, DISABLED
insertvserveripport	<String>	Read-write	Insert an HTTP header, whose value is the IP address and port number of the virtual server, before forwarding a request to the server. The format of the header is <vipHeader>: <virtual server IP address>_<port number >, where vipHeader is the name that you specify for the header. If the virtual server has an IPv6 address, the address in the header is enclosed in brackets ([and]) to separate it from the port number. If you have mapped an IPv4 address to a virtual server's IPv6 address, the value of this parameter determines which IP address is inserted in the header, as follows: * VIPADDR - Insert the IP address of the virtual server in the HTTP header regardless of whether the virtual server has an IPv4 address or an IPv6 address. A mapped IPv4 address, if configured, is ignored. * V6TOV4MAPPING - Insert the IPv4 address that is mapped to the virtual server's IPv6 address. If a mapped IPv4 address is not configured, insert the IPv6 address. * OFF - Disable header insertion. Possible values = OFF, VIPADDR, V6TOV4MAPPING
vipheader	<String>	Read-write	Name for the inserted header. The default name is vip-header. Minimum length = 1
authenticationhost	<String>	Read-write	Fully qualified domain name (FQDN) of the authentication virtual server to which the user must be redirected for authentication. Make sure that the Authentication parameter is set to ENABLED. Minimum length = 3 Maximum length = 252
authentication	<String>	Read-write	Enable or disable user authentication. Default value: OFF Possible values = ON, OFF
authn401	<String>	Read-write	Enable or disable user authentication with HTTP 401 responses. Default value: OFF Possible values = ON, OFF
authnvsname	<String>	Read-write	Name of an authentication virtual server with which to authenticate users. Minimum length = 1 Maximum length = 252
push	<String>	Read-write	Process traffic with the push virtual server that is bound to this load balancing virtual server. Default value: DISABLED Possible values = ENABLED, DISABLED
pushvserver	<String>	Read-write	Name of the load balancing virtual server, of type PUSH or SSL_PUSH, to which the server pushes updates received on the load balancing virtual server that you are configuring. Minimum length = 1
pushlabel	<String>	Read-write	Expression for extracting a label from the server's response. Can be either an expression or the name of a named expression. Default value: "none"
pushmulticlients	<String>	Read-write	Allow multiple Web 2.0 connections from the same client to connect to the virtual server and expect updates. Default value: NO

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
tcpprofilename	<String>	Read-write	Name of the TCP profile whose settings are to be applied to the virtual server. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the HTTP profile whose settings are to be applied to the virtual server. Minimum length = 1 Maximum length = 127
dbprofilename	<String>	Read-write	Name of the DB profile whose settings are to be applied to the virtual server. Minimum length = 1 Maximum length = 127
comment	<String>	Read-write	Any comments that you might want to associate with the virtual server.
l2conn	<String>	Read-write	Use Layer 2 parameters (channel number, MAC address, and VLAN ID) in addition to the 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) that is used to identify a connection. Allows multiple TCP and non-TCP connections with the same 4-tuple to co-exist on the NetScaler appliance. Possible values = ON, OFF
oracleserverversion	<String>	Read-write	Oracle server version. Default value: 10G Possible values = 10G, 11G
mssqlserverversion	<String>	Read-write	For a load balancing virtual server of type MSSQL, the Microsoft SQL Server version. Set this parameter if you expect some clients to run a version different from the version of the database. This setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version. Default value: 2008R2 Possible values = 70, 2000, 2000SP1, 2005, 2008, 2008R2, 2012, 2014
mysqlprotocolversion	<Double>	Read-write	MySQL protocol version that the virtual server advertises to clients.
mysqlserverversion	<String>	Read-write	MySQL server version string that the virtual server advertises to clients. Minimum length = 1 Maximum length = 31
mysqlcharacterset	<Double>	Read-write	Character set that the virtual server advertises to clients.
mysqlservercapabilities	<Double>	Read-write	Server capabilities that the virtual server advertises to clients.
appflowlog	<String>	Read-write	Apply AppFlow logging to the virtual server. Default value: ENABLED Possible values = ENABLED, DISABLED
netprofile	<String>	Read-write	Name of the network profile to associate with the virtual server. If you set this parameter, the virtual server uses only the IP addresses in the network profile as source IP addresses when initiating connections with servers. Minimum length = 1 Maximum length = 127
			How the NetScaler appliance responds to ping requests received for an IP address that is common to one or more virtual servers. Available settings function as follows: * If set to PASSIVE on all the virtual servers that share the IP address, the appliance always responds to the ping requests. * If set to ACTIVE on all the virtual servers that share the IP address, the appliance

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
icmpvsrresponse	<String>	Read-write	appliance does not respond. * If set to ACTIVE on some virtual servers and PASSIVE on the others, the appliance responds if at least one virtual server with the ACTIVE setting is UP. Otherwise, the appliance does not respond. Note: This parameter is available at the virtual server level. A similar parameter, ICMP Response, is available at the IP address level, for IPv4 addresses of type VIP. To set that parameter, use the add ip command in the CLI or the Create IP dialog box in the GUI. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
rhistate	<String>	Read-write	Route Health Injection (RHI) functionality of the NetSaler appliance for advertising the route of the VIP address associated with the virtual server. When Vserver RHI Level (RHI) parameter is set to VSVR_CNTRLD, the following are different RHI behaviors for the VIP address on the basis of RHIstate (RHISTATE) settings on the virtual servers associated with the VIP address: * If you set RHISTATE to PASSIVE on all virtual servers, the NetScaler ADC always advertises the route for the VIP address. * If you set RHISTATE to ACTIVE on all virtual servers, the NetScaler ADC advertises the route for the VIP address if at least one of the associated virtual servers is in UP state. * If you set RHISTATE to ACTIVE on some and PASSIVE on others, the NetScaler ADC advertises the route for the VIP address if at least one of the associated virtual servers, whose RHISTATE set to ACTIVE, is in UP state. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
newservicerequest	<Double>	Read-write	Number of requests, or percentage of the load on existing services, by which to increase the load on a new service at each interval in slow-start mode. A non-zero value indicates that slow-start is applicable. A zero value indicates that the global RR startup parameter is applied. Changing the value to zero will cause services currently in slow start to take the full traffic as determined by the LB method. Subsequently, any new services added will use the global RR factor. Default value: 0
newservicerequestunit	<String>	Read-write	Units in which to increment load at each interval in slow-start mode. Default value: PER_SECOND Possible values = PER_SECOND, PERCENT
newservicerequestincrementinterval	<Double>	Read-write	Interval, in seconds, between successive increments in the load on a new service or a service whose state has just changed from DOWN to UP. A value of 0 (zero) specifies manual slow start. Default value: 0 Minimum value = 0 Maximum value = 3600
minautoscalemembers	<Double>	Read-write	Minimum number of members expected to be present when vserver is used in Autoscale. Default value: 0 Minimum value = 0 Maximum value = 5000
maxautoscalemembers	<Double>	Read-write	Maximum number of members expected to be present when vserver is used in Autoscale. Default value: 0 Minimum value = 0 Maximum value = 5000
persistavpno	<Double[]>	Read-write	Persist AVP number for Diameter Persistency. In case this AVP is not defined in Base RFC 3588 and it is nested inside a Grouped AVP, define a sequence of AVP numbers (max 3) in order of parent to child. So say persist

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 1
skippersistency	<String>	Read-write	This argument decides the behavior incase the service which is selected from an existing persistence session has reached threshold. Default value: None Possible values = Bypass, ReLb, None
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
authnprofile	<String>	Read-write	Name of the authentication profile to be used when authentication is turned on.
macmoderetainvlan	<String>	Read-write	This option is used to retain vlan information of incoming packet when macmode is enabled. Default value: DISABLED Possible values = ENABLED, DISABLED
dbslb	<String>	Read-write	Enable database specific load balancing for MySQL and MSSQL service types. Default value: DISABLED Possible values = ENABLED, DISABLED
dns64	<String>	Read-write	This argument is for enabling/disabling the dns64 on lbvserver. Possible values = ENABLED, DISABLED
bypassaaaa	<String>	Read-write	If this option is enabled while resolving DNS64 query AAAA queries are not sent to back end dns server. Default value: NO Possible values = YES, NO
recursionavailable	<String>	Read-write	When set to YES, this option causes the DNS replies from this vserver to have the RA bit turned on. Typically one would set this option to YES, when the vserver is load balancing a set of DNS servers thatsupport recursive queries. Default value: NO Possible values = YES, NO
processlocal	<String>	Read-write	By turning on this option packets destined to a vserver in a cluster will not under go any steering. Turn this option for single packet request response mode or when the upstream device is performing a proper RSS for connection based distribution. Default value: DISABLED Possible values = ENABLED, DISABLED
dnsprofilename	<String>	Read-write	Name of the DNS profile to be associated with the VServer. DNS profile properties will be applied to the transactions processed by a VServer. This parameter is valid only for DNS and DNS-TCP VServers. Minimum length = 1 Maximum length = 127
weight	<Double>	Read-write	Weight to assign to the specified service. Minimum value = 1 Maximum value = 100
servicename	<String>	Read-write	Service to bind to the virtual server. Minimum length = 1
redirurlflags	<Boolean>	Read-write	The redirect URL to be unset.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
value	<String>	Read-only	SSL status. Possible values = Certkey not bound, SSL feature disabled
ipmapping	<String>	Read-only	The permanent mapping for the V6 Address.
ngname	<String>	Read-only	Nodegroup name to which this lbvsever belongs to.
type	<String>	Read-only	Type of LB vserver. Possible values = CONTENT, ADDRESS
curstate	<String>	Read-only	Current LB vserver state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
effectivestate	<String>	Read-only	Effective state of the LB vserver , based on the state of backup vservers. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
status	<Integer>	Read-only	Current status of the lb vserver. During the initial phase if the configured lb method is not round robin , the vserver will adopt round robin to distribute traffic for a predefined number of requests.
lbrrreason	<Integer>	Read-only	Reason why a vserver is in RR. The following are the reasons: 1 - MEP is DOWN (GSLB) 2 - LB method has changed 3 - Bound service's state changed to UP 4 - A new service is bound 5 - Startup RR factor has changed 6 - LB feature is enabled 7 - Load monitor is not active on a service 8 - Vserver is Enabled 9 - SSL feature is Enabled 10 - All bound services have reached threshold. Using effective state to load balance (GSLB) 11 - Primary state of bound services are not UP. Using effective state to load balance (GSLB) 12 - No LB decision can be made as all bound services have either reached threshold or are not UP (GSLB) 13 - All load monitors are active .
redirect	<String>	Read-only	Cache redirect type. Possible values = CACHE, POLICY, ORIGIN
precedence	<String>	Read-only	Precedence. Possible values = RULE, URL
homepage	<String>	Read-only	Home page.
dnsvservername	<String>	Read-only	DNS vserver name.
domain	<String>	Read-only	Domain.
policyname	<String>	Read-only	Name of the policy bound to the LB vserver.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cachevserver	<String>	Read-only	Cache virtual server.
health	<Double>	Read-only	Health of vserver based on percentage of weights of active svcs/all svcs. This does not consider administratively disabled svcs.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
ruletype	<Double>	Read-only	Rule type.
groupname	<String>	Read-only	LB group to which the lb vserver is to be bound.
cookiedomain	<String>	Read-only	Domain name to be used in the set cookie header in case of cookie persistence.
map	<String>	Read-only	Map. Possible values = ON, OFF
gt2gb	<String>	Read-only	Allow for greater than 2 GB transactions on this vserver. Default value: DISABLED Possible values = ENABLED, DISABLED
consolidatedlconn	<String>	Read-only	Use consolidated stats for LeastConnection. Default value: GLOBAL Possible values = GLOBAL, NO, YES
consolidatedlconngbl	<String>	Read-only	Fetches Global setting. Possible values = YES, NO
thresholdvalue	<Integer>	Read-only	Tells whether threshold exceeded for this service participating in CUSTOMLB.
bindpoint	<String>	Read-only	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
invoke	<Boolean>	Read-only	Invoke policies bound to a virtual server or policy label.
labeltype	<String>	Read-only	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-only	Name of the label invoked.
version	<Integer>	Read-only	Cookie version.
totalservices	<Double>	Read-only	Total number of services bound to the vserver.
activeservices	<Double>	Read-only	Total number of active services bound to the vserver.
statechangetimesec	<String>	Read-only	Time when last state change happened. Seconds part.
statechangetimeseconds	<Double>	Read-only	Time when last state change happened. Seconds part.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
statechangetimemsec	<Double>	Read-only	Time at which last state change happened. Milliseconds part.
tickssincelaststatechange	<Double>	Read-only	Time in 10 millisecond ticks since the last state change.
isgslb	<Boolean>	Read-only	This field is set to true if it is a GSLBVserver.
vsvrdynconnsothreshold	<Double>	Read-only	Spillover threshold for dynamic connection.
backupvserverstatus	<String>	Read-only	Staus of BackUp Vserver .
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"lbvserver":{
    "name":<String_value>,
    "servicetype":<String_value>,
    "ipv46":<String_value>,
    "ippattern":<String_value>,
    "ipmask":<String_value>,
    "port":<Integer_value>,
    "range":<Double_value>,
    "persistencetype":<String_value>,
    "timeout":<Double_value>,
    "persistencebackup":<String_value>,
    "backuppersistencetimeout":<Double_value>,
    "lbmethod":<String_value>,
    "hashlength":<Double_value>,
    "netmask":<String_value>,
    "v6netmasklen":<Double_value>,
    "backuplbmethod":<String_value>,
    "cookiename":<String_value>,
    "rule":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "resrule":<String_value>,
    "persistmask":<String_value>,
    "v6persistmasklen":<Double_value>,
    "pq":<String_value>,
    "sc":<String_value>,
    "rtspnat":<String_value>,
    "m":<String_value>,
    "tosid":<Double_value>,
    "datalength":<Double_value>,
    "dataoffset":<Double_value>,
    "sessionless":<String_value>,
    "state":<String_value>,
    "connfailover":<String_value>,
    "redirurl":<String_value>,
    "cacheable":<String_value>,
    "clttimeout":<Double_value>,
    "somethod":<String_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "healththreshold":<Double_value>,
    "sothreshold":<Double_value>,
    "sobackupaction":<String_value>,
    "redirectportrewrite":<String_value>,
    "downstateflush":<String_value>,
    "backupvserver":<String_value>,
    "disableprimaryondown":<String_value>,
    "insertvserveripport":<String_value>,
    "vipheader":<String_value>,
    "authenticationhost":<String_value>,
    "authentication":<String_value>,
    "authn401":<String_value>,
    "authnvsname":<String_value>,
    "push":<String_value>,
    "pushvserver":<String_value>,
    "pushlabel":<String_value>,
    "pushmulticlients":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "dbprofilename":<String_value>,
    "comment":<String_value>,
    "l2conn":<String_value>,
    "oracleserverversion":<String_value>,
    "mssqlserverversion":<String_value>,
    "mysqlprotocolversion":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "mysqlserverversion":<String_value>,
    "mysqlcharacterset":<Double_value>,
    "mysqlservercapabilities":<Double_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "newservicerequest":<Double_value>,
    "newservicerequestunit":<String_value>,
    "newservicerequestincrementinterval":<Double_value>,
    "minautoscalemembers":<Double_value>,
    "maxautoscalemembers":<Double_value>,
    "persistavpno":<Double[]_value>,
    "skippersistency":<String_value>,
    "td":<Double_value>,
    "authnprofile":<String_value>,
    "macmoderetainvlan":<String_value>,
    "dbslb":<String_value>,
    "dns64":<String_value>,
    "bypassaaaa":<String_value>,
    "recursionavailable":<String_value>,
    "processlocal":<String_value>,
    "dnsprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"lbvserver":{
    "name":<String_value>,
    "ipv46":<String_value>,
    "ippattern":<String_value>,
    "ipmask":<String_value>,
    "weight":<Double_value>,
    "servicename":<String_value>,
    "persistencetype":<String_value>,
    "timeout":<Double_value>,
    "persistencebackup":<String_value>,
    "backuppersistencetimeout":<Double_value>,
    "lbmethod":<String_value>,
    "hashlength":<Double_value>,
    "netmask":<String_value>,
    "v6netmasklen":<Double_value>,
    "backuplbmethod":<String_value>,
    "rule":<String_value>,
    "cookiename":<String_value>,
    "resrule":<String_value>,
    "persistmask":<String_value>,
    "v6persistmasklen":<Double_value>,
    "pq":<String_value>,
    "sc":<String_value>,
    "rtspnat":<String_value>,
    "m":<String_value>,
    "tosid":<Double_value>,
    "datalength":<Double_value>,
    "dataoffset":<Double_value>,
    "sessionless":<String_value>,
    "connfailover":<String_value>,
    "backupvserver":<String_value>,
    "redirurl":<String_value>,
    "cacheable":<String_value>,
    "clttimeout":<Double_value>,
    "somethod":<String_value>,
    "sothreshold":<Double_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "healththreshold":<Double_value>,
    "sobackupaction":<String_value>,
    "redirectportrewrite":<String_value>,
    "downstateflush":<String_value>,
    "insertvserveripport":<String_value>,
    "vipheader":<String_value>,
    "disableprimaryondown":<String_value>,
    "authenticationhost":<String_value>,
    "authentication":<String_value>,
    "authn401":<String_value>,
    "authnvsname":<String_value>,
    "push":<String_value>,
    "pushvserver":<String_value>,
    "pushlabel":<String_value>,
    "pushmulticlients":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "dbprofilename":<String_value>,
    "comment":<String_value>,
    "l2conn":<String_value>,
    "oracleserverversion":<String_value>,
    "mssqlserverversion":<String_value>,
    "mysqlprotocolversion":<Double_value>,
    "mysqlserverversion":<String_value>,
    "mysqlcharacterset":<Double_value>,
    "mysqlservercapabilities":<Double_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "icmpvsrresponse":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "rhistate":<String_value>,
    "newservicerequest":<Double_value>,
    "newservicerequestunit":<String_value>,
    "newservicerequestincrementinterval":<Double_value>,
    "minautoscalemembers":<Double_value>,
    "maxautoscalemembers":<Double_value>,
    "persistavpno":<Double[]_value>,
    "skippersistency":<String_value>,
    "authnprofile":<String_value>,
    "macmoderetainvlan":<String_value>,
    "dbslb":<String_value>,
    "dns64":<String_value>,
    "bypassaaaa":<String_value>,
    "recursionavailable":<String_value>,
    "processlocal":<String_value>,
    "dnsprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbvserver":{
    "name":<String_value>,
    "backupvserver":true,
    "clttimeout":true,
    "redirurl":true,
    "redirurlflags":true,
    "authn401":true,
    "authentication":true,
    "authenticationhost":true,
    "authnvsname":true,
    "pushvserver":true,
    "pushlabel":true,
    "tcpprofilename":true,
    "httpprofilename":true,
    "dbprofilename":true,
    "rule":true,
    "sothreshold":true,
    "l2conn":true,
    "mysqlprotocolversion":true,
    "mysqlserverversion":true,
    "mysqlcharacterset":true,
    "mysqlservercapabilities":true,
    "appflowlog":true,
    "netprofile":true,
    "icmpvsrresponse":true,
    "skippersistency":true,
    "minautoscalemembers":true,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "maxautoscalemembers":true,
        "authnprofile":true,
        "macmoderetainvlan":true,
        "dbslb":true,
        "dnsprofilename":true,
        "servicename":true,
        "persistencetype":true,
        "timeout":true,
        "persistencebackup":true,
        "backuppersistencetimeout":true,
        "lbmethod":true,
        "hashlength":true,
        "netmask":true,
        "v6netmasklen":true,
        "backuplbmethod":true,
        "cookiename":true,
        "resrule":true,
        "persistmask":true,
        "v6persistmasklen":true,
        "pq":true,
        "sc":true,
        "rtspnat":true,
        "m":true,
        "tosid":true,
        "datalength":true,
        "dataoffset":true,
        "sessionless":true,
        "connfailover":true,
        "cacheable":true,
        "somethod":true,
        "sopersistence":true,
        "sopersistencetimeout":true,
        "healththreshold":true,
        "sobackupaction":true,
        "redirectportrewrite":true,
        "downstateflush":true,
        "insertvserveripport":true,
        "vipheader":true,
        "disableprimaryondown":true,
        "push":true,
        "pushmulticlients":true,
        "listenpolicy":true,
        "listenpriority":true,
        "comment":true,
        "oracleserverversion":true,
        "mssqlserverversion":true,
        "rhistate":true,
        "newservicerequest":true,
        "newservicerequestunit":true,
        "newservicerequestincrementinterval":true,
        "persistavpno":true,
        "recursionavailable":true
    }}
```


Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver?**action=enable**

HTTP Method: POST

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbvserver":{
    "name":<String_value>,
    "newname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbvserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lbvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver": [ {
    "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

"insertvserveripport":<String_value>,
"vipheader":<String_value>,
"value":<String_value>,
"ip":<String_value>,
"ipv46":<String_value>,
"ippattern":<String_value>,
"ipmask":<String_value>,
"listenpolicy":<String_value>,
"listenpriority":<Double_value>,
"ipmapping":<String_value>,
"port":<Integer_value>,
"range":<Double_value>,
"servicetype":<String_value>,
"ngname":<String_value>,
"type":<String_value>,
"curstate":<String_value>,
"effectivestate":<String_value>,
"status":<Integer_value>,
"lbrrreason":<Integer_value>,
"redirect":<String_value>,
"precedence":<String_value>,
"redirurl":<String_value>,
"authentication":<String_value>,
"authn401":<String_value>,
"authnvsname":<String_value>,
"homepage":<String_value>,
"dnsvservername":<String_value>,
"domain":<String_value>,
"policyname":<String_value>,
"servicename":<String_value>,
"weight":<Double_value>,
"cachevserver":<String_value>,
"backupvserver":<String_value>,
"clttimeout":<Double_value>,
"somethod":<String_value>,
"sopersistence":<String_value>,
"sopersistencetimeout":<Double_value>,
"healththreshold":<Double_value>,
"sothreshold":<Double_value>,
"sobackupaction":<String_value>,
"lbmethod":<String_value>,
"backuplbmethod":<String_value>,
"hashlength":<Double_value>,
"dataoffset":<Double_value>,
"health":<Double_value>,
"datalength":<Double_value>,
"netmask":<String_value>,
"v6netmasklen":<Double_value>,
"rule":<String_value>,
"resrule":<String_value>,
"gotopriorityexpression":<String_value>,
"ruletype":<Double_value>,
"groupname":<String_value>,
"m":<String_value>,
"tosid":<Double_value>,
"persistencetype":<String_value>,
"timeout":<Double_value>,
"cookiedomain":<String_value>,
"persistmask":<String_value>,
"v6persistmasklen":<Double_value>,
"persistencebackup":<String_value>,
"backuppersistencetimeout":<Double_value>,
"cacheable":<String_value>,
"pq":<String_value>,
"sc":<String_value>,
"rtspnat":<String_value>,
"sessionless":<String_value>,
"map":<String_value>,
"connfailover":<String_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "redirectportrewrite":<String_value>,
    "downstateflush":<String_value>,
    "disableprimaryondown":<String_value>,
    "gt2gb":<String_value>,
    "consolidatedlconn":<String_value>,
    "consolidatedlconngbl":<String_value>,
    "thresholdvalue":<Integer_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "cookiename":<String_value>,
    "version":<Integer_value>,
    "totalservices":<Double_value>,
    "activeservices":<Double_value>,
    "statechangetimesec":<String_value>,
    "statechangetimeseconds":<Double_value>,
    "statechangetimemsec":<Double_value>,
    "tickssincelaststatechange":<Double_value>,
    "authenticationhost":<String_value>,
    "push":<String_value>,
    "pushvserver":<String_value>,
    "pushlabel":<String_value>,
    "pushmulticlients":<String_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "dbprofilename":<String_value>,
    "comment":<String_value>,
    "l2conn":<String_value>,
    "oracleserverversion":<String_value>,
    "mssqlserverversion":<String_value>,
    "mysqlprotocolversion":<Double_value>,
    "mysqlserverversion":<String_value>,
    "mysqlcharacterset":<Double_value>,
    "mysqlservercapabilities":<Double_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "isgslb":<Boolean_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "newservicerequest":<Double_value>,
    "newservicerequestunit":<String_value>,
    "newservicerequestincrementinterval":<Double_value>,
    "persistavpno":<Double[]_value>,
    "skippersistency":<String_value>,
    "td":<Double_value>,
    "minautoscalemembers":<Double_value>,
    "maxautoscalemembers":<Double_value>,
    "authnprofile":<String_value>,
    "macmoderetainvlan":<String_value>,
    "dbslb":<String_value>,
    "dns64":<String_value>,
    "bypassaaaa":<String_value>,
    "recursionavailable":<String_value>,
    "processlocal":<String_value>,
    "vsvrdynconnsothreshold":<Double_value>,
    "backupvserverstatus":<String_value>,
    "dnsprofilename":<String_value>
}]}
```


get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lbvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver": [ {
    "name":<String_value>,
    "insertvserveripport":<String_value>,
    "vipheader":<String_value>,
    "value":<String_value>,
    "ip":<String_value>,
    "ipv46":<String_value>,
    "ippattern":<String_value>,
    "ipmask":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "ipmapping":<String_value>,
    "port":<Integer_value>,
    "range":<Double_value>,
    "servicetype":<String_value>,
    "ngname":<String_value>,
    "type":<String_value>,
    "curstate":<String_value>,
    "effectivestate":<String_value>,
    "status":<Integer_value>,
    "lbrrreason":<Integer_value>,
    "redirect":<String_value>,
    "precedence":<String_value>,
    "redirurl":<String_value>,
    "authentication":<String_value>,
    "authn401":<String_value>,
    "authnvsname":<String_value>,
    "homepage":<String_value>,
    "dnsvservername":<String_value>,
    "domain":<String_value>,
    "policyname":<String_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "cachevserver":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

"backupvserver":<String_value>,
"clttimeout":<Double_value>,
"somethod":<String_value>,
"sopersistence":<String_value>,
"sopersistencetimeout":<Double_value>,
"healththreshold":<Double_value>,
"sothreshold":<Double_value>,
"sobackupaction":<String_value>,
"lbmethod":<String_value>,
"backuplbmethod":<String_value>,
"hashlength":<Double_value>,
"dataoffset":<Double_value>,
"health":<Double_value>,
"datalength":<Double_value>,
"netmask":<String_value>,
"v6netmasklen":<Double_value>,
"rule":<String_value>,
"resrule":<String_value>,
"gotopriorityexpression":<String_value>,
"ruletype":<Double_value>,
"groupname":<String_value>,
"m":<String_value>,
"tosid":<Double_value>,
"persistencetype":<String_value>,
"timeout":<Double_value>,
"cookiedomain":<String_value>,
"persistmask":<String_value>,
"v6persistmasklen":<Double_value>,
"persistencebackup":<String_value>,
"backuppersistencetimeout":<Double_value>,
"cacheable":<String_value>,
"pq":<String_value>,
"sc":<String_value>,
"rtspnat":<String_value>,
"sessionless":<String_value>,
"map":<String_value>,
"connfailover":<String_value>,
"redirectportrewrite":<String_value>,
"downstateflush":<String_value>,
"disableprimaryondown":<String_value>,
"gt2gb":<String_value>,
"consolidatedlconn":<String_value>,
"consolidatedlconngbl":<String_value>,
"thresholdvalue":<Integer_value>,
"bindpoint":<String_value>,
"invoke":<Boolean_value>,
"labeltype":<String_value>,
"labelname":<String_value>,
"cookiename":<String_value>,
"version":<Integer_value>,
"totalservices":<Double_value>,
"activeservices":<Double_value>,
"statechangetimesec":<String_value>,
"statechangetimeseconds":<Double_value>,
"statechangetimemsec":<Double_value>,
"tickssincelaststatechange":<Double_value>,
"authenticationhost":<String_value>,
"push":<String_value>,
"pushvserver":<String_value>,
"pushlabel":<String_value>,
"pushmulticlients":<String_value>,
"tcpprofilename":<String_value>,
"httpprofilename":<String_value>,
"dbprofilename":<String_value>,
"comment":<String_value>,
"l2conn":<String_value>,
"oracleserverversion":<String_value>,
"mssqlserverversion":<String_value>,
"mysqlprotocolversion":<Double_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "mysqlserverversion":<String_value>,
    "mysqlcharacterset":<Double_value>,
    "mysqlservercapabilities":<Double_value>,
    "appflowlog":<String_value>,
    "netprofile":<String_value>,
    "isgslb":<Boolean_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "newservicerequest":<Double_value>,
    "newservicerequestunit":<String_value>,
    "newservicerequestincrementinterval":<Double_value>,
    "persistavpno":<Double[]_value>,
    "skippersistency":<String_value>,
    "td":<Double_value>,
    "minautoscalemembers":<Double_value>,
    "maxautoscalemembers":<Double_value>,
    "authnprofile":<String_value>,
    "macmoderetainvlan":<String_value>,
    "dbslb":<String_value>,
    "dns64":<String_value>,
    "bypassaaaa":<String_value>,
    "recursionavailable":<String_value>,
    "processlocal":<String_value>,
    "vsvrdynconnsothreshold":<Double_value>,
    "backupvserverstatus":<String_value>,
    "dnsprofilename":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_appflowpolicy_binding

May 20, 2016

Binding object showing the appflowpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_appflowpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_appflowpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_appflowpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_appflowpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_appflowpolicy_binding": [ {
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "policyname":<String_value>,
      "name":<String_value>,
      "bindpoint":<String_value>,
      "labeltype":<String_value>,
      "labelname":<String_value>,
      "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appflowpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_appflowpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_appfwpolicy_binding

Jul 07, 2017

Binding object showing the appfwpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_appfwpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_appfwpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_appfwpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_appfwpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_appfwpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "sc":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appfwpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_appfwpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_appqoepolicy_binding

May 20, 2016

Binding object showing the appqoepolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_appqoepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_appqoepolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_appqoepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_appqoepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_appqoepolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_appqoepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_appqoepolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_auditnslogpolicy_binding

May 20, 2016

Binding object showing the auditnslogpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, and cache policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_auditnslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_auditnslogpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_auditnslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_auditnslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_auditnslogpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "sc":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_auditnslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbvserver_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_auditsyslogpolicy_binding

Jul 07, 2017

Binding object showing the auditsyslogpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, and cache policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_auditsyslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_auditsyslogpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_auditsyslogpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "sc":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_auditsyslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbvserver_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbwlm

May 20, 2016

Configuration for web log manager resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
wlmname	<String>	Read-write	The name of the Work Load Manager. Minimum length = 1
ipaddress	<String>	Read-write	The IP address of the WLM.
port	<Integer>	Read-write	The port of the WLM. Range 1 - 65535
lbuid	<String>	Read-write	The LBUID for the Load Balancer to communicate to the Work Load Manager.
katimeout	<Double>	Read-write	The idle time period after which NS would probe the WLM. The value ranges from 1 to 1440 minutes. Default value: 2 Minimum value = 0 Maximum value = 1440
secure	<String>	Read-only	Use this parameter to enable secure mode of communication with WLM. Possible values = YES, NO
state	<String>	Read-only	State of the WLM. Possible values = ACTIVE, INACTIVE, UNKNOWN
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbwlm":{
    "wlmname":<String_value>,
    "ipaddress":<String_value>,
    "port":<Integer_value>,
    "lbuid":<String_value>,
    "katimeout":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm/wlmname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lbwlm":{
    "wlmname":<String_value>,
    "katimeout":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"lbwlm":{
    "wlmname":<String_value>,
    "katimeout":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbwlm?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbwlm?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lbwlm resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lbwlm?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbwlm?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbwlm resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbwlm": [ {
      "wlmname":<String_value>,
      "ipaddress":<String_value>,
      "port":<Integer_value>,
      "secure":<String_value>,
      "katimeout":<Double_value>,
      "lbuid":<String_value>,
      "state":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm/wlmname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lbwlm/wlmname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lbwlm/wlmname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbwlm": [ {
    "wlmname":<String_value>,
    "ipaddress":<String_value>,
    "port":<Integer_value>,
    "secure":<String_value>,
    "katimeout":<Double_value>,
    "lbuid":<String_value>,
    "state":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbwlm": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_authorizationpolicy_binding

May 20, 2016

Binding object showing the authorizationpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_authorizationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_authorizationpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_authorizationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_authorizationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_authorizationpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "sc":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_authorizationpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"lbvserver_authorizationpolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_binding

May 20, 2016

Binding object which returns the resources bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the virtual server. If no name is provided, statistical data of all configured virtual servers is displayed. Minimum length = 1
lbvserver_cmppolicy_binding	<lbvserver_cmppolicy_binding[]>	Read-only	cmppolicy that can be bound to lbvserver.
lbvserver_authorizationpolicy_binding	<lbvserver_authorizationpolicy_binding[]>	Read-only	authorizationpolicy that can be bound to lbvserver.
lbvserver_rewritepolicy_binding	<lbvserver_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to lbvserver.
lbvserver_pqpolicy_binding	<lbvserver_pqpolicy_binding[]>	Read-only	pqpolicy that can be bound to lbvserver.
lbvserver_service_binding	<lbvserver_service_binding[]>	Read-only	service that can be bound to lbvserver.
lbvserver_tmtrafficpolicy_binding	<lbvserver_tmtrafficpolicy_binding[]>	Read-only	tmtrafficpolicy that can be bound to lbvserver.
lbvserver_auditsyslogpolicy_binding	<lbvserver_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to lbvserver.
lbvserver_appfwpolicy_binding	<lbvserver_appfwpolicy_binding[]>	Read-only	appfwpolicy that can be bound to lbvserver.
lbvserver_spilloverpolicy_binding	<lbvserver_spilloverpolicy_binding[]>	Read-only	spilloverpolicy that can be bound to lbvserver.
lbvserver_csvserver_binding	<lbvserver_csvserver_binding[]>	Read-only	csvserver that can be bound to lbvserver.
lbvserver_capolicy_binding	<lbvserver_capolicy_binding[]>	Read-only	capolicy that can be bound to lbvserver.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			be bound to lbvserver.
lbvserver_filterpolicy_binding	<lbvserver_filterpolicy_binding[]>	Read-only	filterpolicy that can be bound to lbvserver.
lbvserver_appflowpolicy_binding	<lbvserver_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to lbvserver.
lbvserver_responderpolicy_binding	<lbvserver_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to lbvserver.
lbvserver_transformpolicy_binding	<lbvserver_transformpolicy_binding[]>	Read-only	transformpolicy that can be bound to lbvserver.
lbvserver_dospolicy_binding	<lbvserver_dospolicy_binding[]>	Read-only	dospolicy that can be bound to lbvserver.
lbvserver_feopolicy_binding	<lbvserver_feopolicy_binding[]>	Read-only	feopolicy that can be bound to lbvserver.
lbvserver_servicegroupmember_binding	<lbvserver_servicegroupmember_binding[]>	Read-only	servicegroupmember that can be bound to lbvserver.
lbvserver_dnspolicy64_binding	<lbvserver_dnspolicy64_binding[]>	Read-only	dnspolicy64 that can be bound to lbvserver.
lbvserver_cachepolicy_binding	<lbvserver_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to lbvserver.
lbvserver_scpolicy_binding	<lbvserver_scpolicy_binding[]>	Read-only	scpolicy that can be bound to lbvserver.
lbvserver_servicegroup_binding	<lbvserver_servicegroup_binding[]>	Read-only	servicegroup that can be bound to lbvserver.
lbvserver_appqoepolicy_binding	<lbvserver_appqoepolicy_binding[]>	Read-only	appqoepolicy that can be bound to lbvserver.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lbvserver_binding": [ {
    "name":<String_value>,
    "lbvserver_appfwpolicy_binding":<lbvserver_appfwpolicy_binding[]_value>,
    "lbvserver_spilloverpolicy_binding":<lbvserver_spilloverpolicy_binding[]_value>,
    "lbvserver_appqoepolicy_binding":<lbvserver_appqoepolicy_binding[]_value>,
    "lbvserver_feopolicy_binding":<lbvserver_feopolicy_binding[]_value>,
    "lbvserver_csvserver_binding":<lbvserver_csvserver_binding[]_value>,
    "lbvserver_capolicy_binding":<lbvserver_capolicy_binding[]_value>,
    "lbvserver_authorizationpolicy_binding":<lbvserver_authorizationpolicy_binding[]_value>,
    "lbvserver_servicegroupmember_binding":<lbvserver_servicegroupmember_binding[]_value>,
    "lbvserver_dospolicy_binding":<lbvserver_dospolicy_binding[]_value>,
    "lbvserver_tmtrafficpolicy_binding":<lbvserver_tmtrafficpolicy_binding[]_value>,
    "lbvserver_cmppolicy_binding":<lbvserver_cmppolicy_binding[]_value>,
    "lbvserver_auditsyslogpolicy_binding":<lbvserver_auditsyslogpolicy_binding[]_value>,
    "lbvserver_pqpolicy_binding":<lbvserver_pqpolicy_binding[]_value>,
    "lbvserver_dnspolicy64_binding":<lbvserver_dnspolicy64_binding[]_value>,
    "lbvserver_rewritepolicy_binding":<lbvserver_rewritepolicy_binding[]_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "lbvserver_auditnslogpolicy_binding":<lbvserver_auditnslogpolicy_binding[]_value>,
    "lbvserver_transformpolicy_binding":<lbvserver_transformpolicy_binding[]_value>,
    "lbvserver_filterpolicy_binding":<lbvserver_filterpolicy_binding[]_value>,
    "lbvserver_scpolicy_binding":<lbvserver_scpolicy_binding[]_value>,
    "lbvserver_appflowpolicy_binding":<lbvserver_appflowpolicy_binding[]_value>,
    "lbvserver_servicegroup_binding":<lbvserver_servicegroup_binding[]_value>,
    "lbvserver_cachepolicy_binding":<lbvserver_cachepolicy_binding[]_value>,
    "lbvserver_service_binding":<lbvserver_service_binding[]_value>,
    "lbvserver_responderpolicy_binding":<lbvserver_responderpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_cachepolicy_binding

Jul 07, 2017

Binding object showing the cachepolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_cachepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_cachepolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_cachepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_cachepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_cachepolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_cachepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_cachepolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbwlm_binding

May 20, 2016

Binding object which returns the resources bound to lbwlm.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
wlmname	<String>	Read-write	The name of the work load manager. Minimum length = 1
lbwlm_lbvserver_binding	<lbwlm_lbvserver_binding[]>	Read-only	lbvserver that can be bound to lbwlm.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm_binding/wlmname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lbwlm_binding": [ {
      "wlmname":<String_value>,
      "lbwlm_lbvserver_binding":<lbwlm_lbvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_capolicy_binding

May 20, 2016

Binding object showing the capolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_capolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_capolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_capolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_capolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_capolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_capolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_capolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_cmppolicy_binding

May 20, 2016

Binding object showing the cmppolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_cmppolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_cmppolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_cmppolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_cmppolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_cmppolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "sc":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_cmppolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_cmppolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
cachevserver	<String>	Read-write	Cache virtual server.
cachetype	<String>	Read-only	Cache type. Possible values = TRANSPARENT, REVERSE, FORWARD
hits	<Double>	Read-only	Number of hits.
pipolicyhits	<Double>	Read-only	Number of hits.
policysubtype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_csvserver_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "cachevserver":<String_value>,
    "cachetype":<String_value>,
    "hits":<Double_value>,
    "pipolicyhits":<Double_value>,
    "policysubtype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_csvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_dnspolicy64_binding

May 20, 2016

Binding object showing the dnspolicy64 that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, and cache policies. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_dnspolicy64_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_dnspolicy64_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_dnspolicy64_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_dnspolicy64_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_dnspolicy64_binding": [ {
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "policyname":<String_value>,
      "name":<String_value>,
      "bindpoint":<String_value>,
      "labeltype":<String_value>,
      "labelname":<String_value>,
      "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_dnspolicy64_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"lbvserver_dnspolicy64_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_dospolicy_binding

May 20, 2016

Binding object showing the dospolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_dospolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_dospolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_dospolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_dospolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_dospolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_dospolicy_binding": [ {
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "policyname":<String_value>,
    "name":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_dospolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_dospolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_feopolicy_binding

May 20, 2016

Binding object showing the feopolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_feopolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_feopolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_feopolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_feopolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_feopolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_feopolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_feopolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_filterpolicy_binding

May 20, 2016

Binding object showing the filterpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, and cache policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
sc	<String>	Read-only	Use SureConnect on the virtual server. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_filterpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_filterpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_filterpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_filterpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_filterpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "sc":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_filterpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbvserver_filterpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_pqpolicy_binding

Jul 07, 2017

Binding object showing the pqpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Integer specifying the policy's priority. The lower the priority number, the higher the policy's priority. Minimum value = 1 Maximum value = 2147483647
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			and cache policies. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_pqpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_pqpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_pqpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_pqpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_pqpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_pqpolicy_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbvserver_pqpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_scpolicy_binding

May 20, 2016

Binding object showing the scpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, and cache policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_scpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_scpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_scpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_scpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_scpolicy_binding": [ {
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "policyname":<String_value>,
      "name":<String_value>,
      "bindpoint":<String_value>,
      "labeltype":<String_value>,
      "labelname":<String_value>,
      "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_scpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbvserver_scpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_responderpolicy_binding

May 20, 2016

Binding object showing the responderpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, and cache policies. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_responderpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_responderpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_responderpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_responderpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_responderpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_responderpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_responderpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_rewritepolicy_binding

Jul 07, 2017

Binding object showing the rewritepolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_rewritepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_rewritepolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_rewritepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_rewritepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_rewritepolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_rewritepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_rewritepolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_servicegroupmember_binding

Jul 07, 2017

Binding object showing the servicegroupmember that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
weight	<Double>	Read-write	Weight to assign to the specified service. Minimum value = 1 Maximum value = 100
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
servicegroupname	<String>	Read-write	The service group name bound to the selected load balancing virtual server.
preferredlocation	<String>	Read-only	Used for displaying the location of bound services.
vserverid	<String>	Read-only	Vserver Id.
cookiename	<String>	Read-only	Use this parameter to specify the cookie name for COOKIE peristence type. It specifies the name of cookie with a maximum of 32 characters. If not specified, cookie name is internally generated.
servicetype	<String>	Read-only	Protocol used by the service (also called the service type). Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP, SYSLOGUDP, FIX
cookieipport	<String>	Read-only	Encryped Ip address and port of the service that is inserted into the set-cookie http header.
port	<Integer>	Read-only	Port number for the virtual server. Range 1 - 65535
curstate	<String>	Read-only	Current LB vserver state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			DISABLED
ipv46	<String>	Read-only	IPv4 or IPv6 address to assign to the virtual server.
dynamicweight	<Double>	Read-only	Dynamic weight.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_servicegroupmember_binding/name_value<String>

Query-parameters:

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_servicegroupmember_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_servicegroupmember_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_servicegroupmember_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_servicegroupmember_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_servicegroupmember_binding": [ {
    "weight":<Double_value>,
    "name":<String_value>,
    "servicegroupname":<String_value>,
    "preferredlocation":<String_value>,
    "vserverid":<String_value>,
    "cookiename":<String_value>,
    "servicetype":<String_value>,
    "cookieipport":<String_value>,
    "port":<Integer_value>,
    "curstate":<String_value>,
    "ipv46":<String_value>,
    "dynamicweight":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_servicegroupmember_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbvserver_servicegroupmember_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_servicegroup_binding

May 20, 2016

Binding object showing the servicegroup that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
weight	<Double>	Read-write	Integer specifying the weight of the service. A larger number specifies a greater weight. Defines the capacity of the service relative to the other services in the load balancing configuration. Determines the priority given to the service in load balancing decisions. Default value: 1 Minimum value = 1 Maximum value = 100
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
servicename	<String>	Read-write	Service to bind to the virtual server. Minimum length = 1
servicegroupname	<String>	Read-write	The service group name bound to the selected load balancing virtual server.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_servicegroup_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_servicegroup_binding":{
    "name":<String_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "servicegroupname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_servicegroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_servicegroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "lbvserver_servicegroup_binding": [ {
      "weight":<Double_value>,
      "name":<String_value>,
      "servicename":<String_value>,
      "servicegroupname":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_servicegroup_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_servicegroup_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_service_binding

May 20, 2016

Binding object showing the service that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
weight	<Double>	Read-write	Weight to assign to the specified service. Minimum value = 1 Maximum value = 100
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
servicename	<String>	Read-write	Service to bind to the virtual server. Minimum length = 1
servicegroupname	<String>	Read-write	Name of the service group. Minimum length = 1
preferredlocation	<String>	Read-only	Used for displaying the location of bound services.
vserverid	<String>	Read-only	Vserver Id.
vsvrbindsvcip	<String>	Read-only	used for showing the ip of bound entities.
servicetype	<String>	Read-only	Protocol used by the service (also called the service type). Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, DNS, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, RTSP, PUSH, SSL_PUSH, RADIUS, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, SMPP, SYSLOGTCP, SYSLOGUDP, FIX
cookieipport	<String>	Read-only	Encryped Ip address and port of the service that is inserted into the set-cookie http header.
port	<Integer>	Read-only	Port number for the virtual server. Range 1 - 65535

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Range 1 - 65535
curstate	<String>	Read-only	Current LB vserver state. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
ipv46	<String>	Read-only	IPv4 or IPv6 address to assign to the virtual server.
dynamicweight	<Double>	Read-only	Dynamic weight.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_service_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_service_binding":{
    "name":<String_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "servicegroupname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_service_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_service_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_service_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of lbvserver_service_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_service_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_service_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_service_binding": [ {
    "weight":<Double_value>,
    "name":<String_value>,
    "servicename":<String_value>,
    "servicegroupname":<String_value>,
    "preferredlocation":<String_value>,
    "vserverid":<String_value>,
    "vsvrbindsvcip":<String_value>,
    "servicetype":<String_value>,
    "cookieipport":<String_value>,
    "port":<Integer_value>,
    "vsvrbindsvcport":<Integer_value>,
    "curstate":<String_value>,
    "ipv46":<String_value>,
    "dynamicweight":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_service_binding/name_value<String>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbvserver_service_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_spilloverpolicy_binding

Jul 07, 2017

Binding object showing the spilloverpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_spilloverpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_spilloverpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_spilloverpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_spilloverpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_spilloverpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_spilloverpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_spilloverpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_tmtrafficpolicy_binding

May 20, 2016

Binding object showing the tmtrafficpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applicable only to compression, rewrite, and cache policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
labeltype	<String>	Read-write	Type of policy label to invoke. Applicable only to rewrite and cache policies. Available settings function as follows: * reqvserver - Evaluate the request against the request-based policies bound to the specified virtual server. * resvserver - Evaluate the response against the response-based policies bound to the specified virtual server. * policylabel - invoke the request or response against the specified user-defined policy label. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_tmtrafficpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_tmtrafficpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_tmtrafficpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_tmtrafficpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_tmtrafficpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_tmtrafficpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbvserver_tmtrafficpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver_transformpolicy_binding

May 20, 2016

Binding object showing the transformpolicy that can be bound to lbvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the policy bound to the LB vserver.
name	<String>	Read-write	Name for the virtual server. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Can be changed after the virtual server is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my vserver" or 'my vserver'). . Minimum length = 1
bindpoint	<String>	Read-write	The bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE
labeltype	<String>	Read-write	The invocation type. Possible values = reqvserver, resvserver, policylabel
labelname	<String>	Read-write	Name of the label invoked.
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server or policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbvserver_transformpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbvserver_transformpolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbvserver_transformpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbvserver_transformpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver_transformpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "bindpoint":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbvserver_transformpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lbvserver_transformpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbwlm_lbvserver_binding

Jul 07, 2017

Binding object showing the lbvserver that can be bound to lbwlm.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
wlmname	<String>	Read-write	The name of the Work Load Manager. Minimum length = 1
vservername	<String>	Read-write	Name of the virtual server which is to be bound to the WLM.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lbwlm_lbvserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lbwlm_lbvserver_binding":{
    "wlmname":<String_value>,
    "vservername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lbwlm_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lbwlm_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbwlm_lbvserver_binding": [ {
    "wlmname":<String_value>,
    "vservername":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lbwlm_lbvserver_binding/wlmname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lbwlm_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

LLDP

May 20, 2016

Link Layer Discovery Protocol.

Name	Description
lldpneighbors	Configuration for lldp neighbors
lldpparam	Configuration for lldp params

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lldpneighbors

Jul 07, 2017

Configuration for lldp neighbors resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ifnum	<String>	Read-write	Interface Name.
chassisidsubtype	<String>	Read-only	Chassis id sub type. Possible values = NONE, Component, Interface Alias, Port Component, MAC Address, Network Address, Interface Name, Locally Assigned
chassisid	<String>	Read-only	Chassis Id.
portidsubtype	<String>	Read-only	Port id subtype. Possible values = NONE, Interface Alias, Port Component, MAC Address, Network Address, Interface Name, Agent Circuit ID, Locally Assigned
portid	<String>	Read-only	Port Id.
ttl	<Double>	Read-only	Time to Live.
portdescription	<String>	Read-only	Port Description.
sys	<String>	Read-only	System Name.
sysdesc	<String>	Read-only	System Description.
mgmtaddresssubtype	<String>	Read-only	Management Address Type. Possible values = OTHER, IPv4, IPv6
mgmtaddress	<String>	Read-only	Management Address.
iftype	<String>	Read-only	Interface subtype. Possible values = UNKNOWN, ifIndex, system port number
ifnumber	<Double>	Read-only	Interface Number.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
vlan	<String>	Read-only	vlan name.
vlanid	<Double>	Read-only	Vlan Id.
portprotosupported	<Double>	Read-only	Flag to show Port Protocol Support.
portprotoenabled	<Double>	Read-only	Flag to show Port Protocol support enabled.
portprotoid	<Double>	Read-only	Port Protocol ID.
portvlanid	<Double>	Read-only	Port Vlan Id.
protocolid	<String>	Read-only	port vlan id name.
linkaggrcapable	<String>	Read-only	Is neighbor Link Aggregation Capable. Possible values = NO, YES
linkaggrenabled	<String>	Read-only	Is link aggregation Enabled. Possible values = NO, YES
linkaggrid	<Double>	Read-only	Link Aggregation Id.
flag	<Double>	Read-only	.
syscapabilities	<String>	Read-only	Acronyms for remote system capabilities: OT : Other. RE : Repeater(IETF RFC 2108). BR : MAC Bridge(IEEE Std 802.1D). WL : WLAN Access Point(IEEE Std 802.11 MIB). RO : ROuter(IETF RFC 1812). TE : Telephone(IETF RFC 4293). DO : DOCSIS cable device(IETF RFC 4639 and IETF RFC 4546). ST : STation only(IETF RFC 4293). CV : C-VLAN component of a VLAN Bridge(IEEE Std 802.1Q). SV : S-VLAN component of a VLAN Bridge(IEEE Std 802.1Q). MR : Two port MAC Relay(IEEE Std 802.1Q).
syscapenabled	<String>	Read-only	Acronyms for remote system enabled capabilities: OT : Other. RE : Repeater(IETF RFC 2108). BR : MAC Bridge(IEEE Std 802.1D). WL : WLAN Access Point(IEEE Std 802.11 MIB). RO : ROuter(IETF RFC 1812). TE : Telephone(IETF RFC 4293). DO : DOCSIS cable device(IETF RFC 4639 and IETF RFC 4546).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			CV : C-VLAN component of a VLAN Bridge(IEEE Std 802.1Q). SV : S-VLAN component of a VLAN Bridge(IEEE Std 802.1Q). MR : Two port MAC Relay(IEEE Std 802.1Q).
autonegsupport	<String>	Read-only	MAC/PHY autonegotiation support. Possible values = NO, YES
autonegenabled	<String>	Read-only	MAC/PHY autonegotiation enabled. Possible values = NO, YES
autonegadvertised	<String>	Read-only	MAC/PHY autonegotiation advetised.
autonegmautype	<String>	Read-only	MAC/PHY Medium Attachment Unit (MAU) type of the port. Description listed below: AUI - no internal MAU, view from AUI 10Base5 - thick coax MAU Foirl - FOIRL MAU 10Base2 - thin coax MAU 10BaseT - UTP MAU 10BaseFP - passive fiber MAU 10BaseFB - sync fiber MAU 10BaseFL - async fiber MAU 10Broad36 - broadband DTE MAU 10BaseTHD - UTP MAU, half duplex mode 10BaseTFD - UTP MAU, full duplex mode 10BaseFLHD - async fiber MAU, half duplex mode 10BaseFLDF - async fiber MAU, full duplex mode 10BaseT4 - 4 pair category 3 UTP 100BaseTXHD - 2 pair category 5 UTP, half duplex mode 100BaseTXFD - 2 pair category 5 UTP, full duplex mode 100BaseFXHD - X fiber over PMT, half duplex mode 100BaseFXFD - X fiber over PMT, full duplex mode 100BaseT2HD - 2 pair category 3 UTP, half duplex mode 100BaseT2DF - 2 pair category 3 UTP, full duplex mode 1000BaseXHD - PCS/PMA, unknown PMD, half duplex mode 1000BaseXFD - PCS/PMA, unknown PMD, full duplex mode 1000BaseLXHD - Fiber over long-wavelength laser, half duplex mode 1000BaseLXFD - Fiber over long-wavelength laser, full duplex mode 1000BaseSXHD - Fiber over short-wavelength laser, half duplex mode 1000BaseSXFD - Fiber over short-wavelength laser, full duplex mode 1000BaseCXHD - Copper over 150-Ohm balanced cable, half duplex mode 1000BaseCXFD - Copper over 150-Ohm balanced cable, full duplex mode 1000BaseTHD - Four-pair Category 5 UTP, half duplex mode 1000BaseTFD - Four-pair Category 5 UTP, full duplex mode 10GigBaseX - X PCS/PMA, unknown PMD 10GigBaseLX4 - X fiber over WWDM optics 10GigBaseR - R PCS/PMA, unknown PMD 10GigBaseER - R fiber over 1550 nm optics 10GigBaseLR - R fiber over 1310 nm optics 10GigBaseSR - R fiber over 850 nm optics 10GigBaseW - W PCS/PMA, unknown PMD 10GigBaseEW - W fiber over 1550 nm optics 10GigBaseLW - W fiber over 1310 nm optics

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = Not Recieved, AUI, 10BaseS, FOIrl, 10Base2, 10BaseT, 10BaseFP, 10BaseFB, 10BaseFL, 10Broad36, 10BaseTHD, 10BaseTFD, 10BaseFLHD, 10BaseFLDF, 10BaseT4, 100BaseTXHD, 100BaseTXFD, 100BaseFXHD, 100BaseFXFD, 100BaseT2HD, 100BaseT2DF, 1000BaseXHD, 1000BaseXFD, 1000BaseLXHD, 1000BaseLXFD, 1000BaseSXHD, 1000BaseSXFD, 1000BaseCXHD, 1000BaseCXFD, 1000BaseTHD, 1000BaseTFD, 10GigBaseX, 10GigBaseLX4, 10GigBaseR, 10GigBaseER, 10GigBaseLR, 10GigBaseSR, 10GigBaseW, 10GigBaseEW, 10GigBaseLW, 10GigBaseSW
mtu	<Double>	Read-only	MTU of Remote Device.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

CLEAR | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/lldpneighbors?**action=clear**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lldpneighbors":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lldpneighbors

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lldpneighbors?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lldpneighbors?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lldpneighbors resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lldpneighbors?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lldpneighbors?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lldpneighbors resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lldpneighbors": [ {
    "ifnum":<String_value>,
    "chassisidsubtype":<String_value>,
    "chassisid":<String_value>,
    "portidsubtype":<String_value>,
    "portid":<String_value>,
    "ttl":<Double_value>,
    "portdescription":<String_value>,
    "sys":<String_value>,
    "sysdesc":<String_value>,
    "mgmtaddresssubtype":<String_value>,
    "mgmtaddress":<String_value>,
    "iftype":<String_value>,
    "ifnumber":<Double_value>,
    "vlan":<String_value>,
    "vlanid":<Double_value>,
    "portprotosupported":<Double_value>,
    "portprotoenabled":<Double_value>,
    "portprotoid":<Double_value>,
    "portvlanid":<Double_value>,
    "protocolid":<String_value>,
    "linkaggrcapable":<String_value>,
    "linkaggrenabled":<String_value>,
    "linkaggrid":<Double_value>,
    "flag":<Double_value>,
    "syscapabilities":<String_value>,
    "syscapenabled":<String_value>,
    "autonegsupport":<String_value>,
    "autonegenabled":<String_value>,
    "autonegadvertised":<String_value>,
    "autonegmautype":<String_value>,
    "mtu":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lldpneighbors/ifnum_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lldpneighbors/ifnum_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lldpneighbors/ifnum_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lldpneighbors": [ {
    "ifnum":<String_value>,
    "chassisidsubtype":<String_value>,
    "chassisid":<String_value>,
    "portidsubtype":<String_value>,
    "portid":<String_value>,
    "ttl":<Double_value>,
    "portdescription":<String_value>,
    "sys":<String_value>,
    "sysdesc":<String_value>,
    "mgmtaddresssubtype":<String_value>,
    "mgmtaddress":<String_value>,
    "iftype":<String_value>,
    "ifnumber":<Double_value>,
    "vlan":<String_value>,
    "vlanid":<Double_value>,
    "portprotosupported":<Double_value>,
    "portprotoenabled":<Double_value>,
    "portprotoid":<Double_value>,
    "portvlanid":<Double_value>,
    "protocolid":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "linkaggrcapable":<String_value>,
    "linkaggrenabled":<String_value>,
    "linkaggrid":<Double_value>,
    "flag":<Double_value>,
    "syscapabilities":<String_value>,
    "syscapenabled":<String_value>,
    "autonegsupport":<String_value>,
    "autonegenabled":<String_value>,
    "autonegadvertised":<String_value>,
    "autonegmautype":<String_value>,
    "mtu":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lldpneighbors?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lldpneighbors": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lldpparam

May 20, 2016

Configuration for lldp params resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
holdtimetxmult	<Double>	Read-write	A multiplier for calculating the duration for which the receiving device stores the LLDP information in its database before discarding or removing it. The duration is calculated as the holdtimeTxMult (Holdtime Multiplier) parameter value multiplied by the timer (Timer) parameter value. Default value: 4 Minimum value = 1 Maximum value = 20
timer	<Double>	Read-write	Interval, in seconds, between LLDP packet data units (LLDPDUs). that the NetScaler ADC sends to a directly connected device. Default value: 30 Minimum value = 1 Maximum value = 3000
mode	<String>	Read-write	Global mode of Link Layer Discovery Protocol (LLDP) on the NetScaler ADC. The resultant LLDP mode of an interface depends on the LLDP mode configured at the global and the interface levels. Possible values = NONE, TRANSMITTER, RECEIVER, TRANSCEIVER

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lldpparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lldpparam":{
    "holdtimetxmult":<Double_value>,
    "timer":<Double_value>,
    "mode":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lldpparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"lldpparam":{
    "holdtimetxmult":true,
    "timer":true,
    "mode":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lldpparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lldpparam": [ {
    "holdtimetxmult":<Double_value>,
    "timer":<Double_value>,
    "mode":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Lsn

May 20, 2016

Large Scale NAT commands

Name	Description
lsnappsprofile	Configuration for LSN Application Profile
lsnappsprofile_binding	Binding object showing the resources that can be bound to lsnappsprofile
lsnappsprofile_port_binding	Binding object showing the port that can be bound to lsnappsprofile
lsnclient	Configuration for lsn client
lsnclient_binding	Binding object showing the resources that can be bound to lsnclient
lsnclient_network6_binding	Binding object showing the network6 that can be bound to lsnclient
lsnclient_network_binding	Binding object showing the network that can be bound to lsnclient
lsnclient_nsacl6_binding	Binding object showing the nsacl6 that can be bound to lsnclient
lsnclient_nsacl_binding	Binding object showing the nsacl that can be bound to lsnclient
lsndeterministicnat	Configuration for deterministic NAT
lsngroup	Configuration for LSN group
lsngroup_binding	Binding object showing the resources that can be bound to lsngroup
lsngroup_lsnappsprofile_binding	Binding object showing the lsnappsprofile that can be bound to lsngroup
lsngroup_lsnhttphdrlogprofile_binding	Binding object showing the lsnhttphdrlogprofile that can be bound to lsngroup
lsngroup_lsnlogprofile_binding	Binding object showing the lsnlogprofile that can be bound to lsngroup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup_lsnpool_binding	
Name	**Description**
lsngroup_lsnrtspalgprofile_binding	Binding object showing the lsnrtspalgprofile that can be bound to lsngroup
lsngroup_lsnsipalgprofile_binding	Binding object showing the lsnsipalgprofile that can be bound to lsngroup
lsngroup_lsntransportprofile_binding	Binding object showing the lsntransportprofile that can be bound to lsngroup
lsnhttphdrlogprofile	Configuration for LSN HTTP header logging Profile
lsnip6profile	Configuration for LSN ip6 Profile
lsnlogprofile	Configuration for LSN logging Profile
lsnparameter	Configuration for LSN parameter
lsnpool	Configuration for LSN pool
lsnpool_binding	Binding object showing the resources that can be bound to lsnpool
lsnpool_lsnip_binding	Binding object showing the lsnip that can be bound to lsnpool
lsnrtspalgprofile	Configuration for LSN RTSPALG Profile
lsnrtspalgsession	Configuration for LSN RTSPALG session
lsnrtspalgsession_binding	Binding object showing the resources that can be bound to lsnrtspalgsession
lsnrtspalgsession_datachannel_binding	Binding object showing the datachannel that can be bound to lsnrtspalgsession
lsnsession	Configuration for lsn session
lsnsipalgcall	Configuration for LSN SIPALG call
lsnsipalgcall_binding	Binding object showing the resources that can be bound to lsnsipalgcall
lsnsipalgcall_controlchannel_binding	Binding object showing the controlchannel that can be bound to lsnsipalgcall

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnsipalgcall_datachannel_binding	
Name	**Description**
lsnsipalgprofile	Configuration for LSN SIPALG Profile
lsnstatic	Configuration for static mapping
lsntransportprofile	Configuration for LSN Transport Profile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnappsprofile

May 20, 2016

Configuration for LSN Application Profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
appsprofilename	<String>	Read-write	Name for the LSN application profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN application profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn application profile1" or 'lsn application profile1'). Minimum length = 1 Maximum length = 127
transportprotocol	<String>	Read-write	Name of the protocol for which the parameters of this LSN application profile applies. Possible values = TCP, UDP, ICMP
ippooling	<String>	Read-write	NAT IP address allocation options for sessions associated with the same subscriber. Available options function as follows: * Paired - The NetScaler ADC allocates the same NAT IP address for all sessions associated with the same subscriber. When all the ports of a NAT IP address are used in LSN sessions (for same or multiple subscribers), the NetScaler ADC drops any new connection from the subscriber. * Random - The NetScaler ADC allocates random NAT IP addresses, from the pool, for different sessions associated with the same subscriber. This parameter is applicable to dynamic NAT allocation only. Default value: RANDOM Possible values = PAIRED, RANDOM
mapping	<String>	Read-write	Type of LSN mapping to apply to subsequent packets originating from the same subscriber IP address and port. Consider an example of an LSN mapping that includes the mapping of the subscriber IP:port (X:x), NAT IP:port (N:n), and external host IP:port (Y:y). Available options function as follows: * ENDPOINT-INDEPENDENT - Reuse the LSN mapping for subsequent packets sent from the same subscriber IP address and port (X:x) to any external IP address and port.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			the same subscriber IP address and port (X:x) to the same external IP address (Y), regardless of the external port. * ADDRESS-PORT-DEPENDENT - Reuse the LSN mapping for subsequent packets sent from the same internal IP address and port (X:x) to the same external IP address and port (Y:y) while the mapping is still active. Default value: ADDRESS-PORT-DEPENDENT Possible values = ENDPOINT-INDEPENDENT, ADDRESS-DEPENDENT, ADDRESS-PORT-DEPENDENT
filtering	<String>	Read-write	Type of filter to apply to packets originating from external hosts. Consider an example of an LSN mapping that includes the mapping of subscriber IP:port (X:x), NAT IP:port (N:n), and external host IP:port (Y:y). Available options function as follows: * ENDPOINT INDEPENDENT - Filters out only packets not destined to the subscriber IP address and port X:x, regardless of the external host IP address and port source (Z:z). The NetScaler ADC forwards any packets destined to X:x. In other words, sending packets from the subscriber to any external IP address is sufficient to allow packets from any external hosts to the subscriber. * ADDRESS DEPENDENT - Filters out packets not destined to subscriber IP address and port X:x. In addition, the ADC filters out packets from Y:y destined for the subscriber (X:x) if the client has not previously sent packets to Y:anyport (external port independent). In other words, receiving packets from a specific external host requires that the subscriber first send packets to that specific external host's IP address. * ADDRESS PORT DEPENDENT (the default) - Filters out packets not destined to subscriber IP address and port (X:x). In addition, the NetScaler ADC filters out packets from Y:y destined for the subscriber (X:x) if the subscriber has not previously sent packets to Y:y. In other words, receiving packets from a specific external host requires that the subscriber first send packets first to that external IP address and port. Default value: ADDRESS-PORT-DEPENDENT Possible values = ENDPOINT-INDEPENDENT, ADDRESS-DEPENDENT, ADDRESS-PORT-DEPENDENT
tcpproxy	<String>	Read-write	Enable TCP proxy, which enables the NetScaler appliance to optimize the TCP traffic by using Layer 4 features. Default value: DISABLED Possible values = ENABLED, DISABLED
td	<Double>	Read-write	ID of the traffic domain through which the NetScaler ADC sends the outbound traffic after performing LSN. If you do not specify an ID, the ADC sends the outbound traffic through the default traffic domain, which has an ID of 0. Default value: 65535
l2info	<String>	Read-write	Enable l2info by creating natpcbs for LSN, which enables the NetScaler appliance to use L2CONN/MBF with LSN. Default value: DISABLED Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnappsprofile":{
    "appsprofilename":<String_value>,
    "transportprotocol":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ippooling":<String_value>,
    "mapping":<String_value>,
    "filtering":<String_value>,
    "tcpproxy":<String_value>,
    "td":<Double_value>,
    "l2info":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile/appsprofilename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnappsprofile":{
    "appsprofilename":<String_value>,
    "ippooling":<String_value>,
    "mapping":<String_value>,
    "filtering":<String_value>,
    "tcpproxy":<String_value>,
    "td":<Double_value>,
    "l2info":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnappsprofile":{
    "appsprofilename":<String_value>,
    "ippooling":true,
    "mapping":true,
    "filtering":true,
    "tcpproxy":true,
    "td":true,
    "l2info":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnappsprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnappsprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnappsprofile": [ {
      "appsprofilename":<String_value>,
      "transportprotocol":<String_value>,
      "ippooling":<String_value>,
      "mapping":<String_value>,
      "filtering":<String_value>,
      "tcpproxy":<String_value>,
      "td":<Double_value>,
      "l2info":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile/appsprofilename_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile/appsprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile/appsprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnappsprofile": [ {
      "appsprofilename":<String_value>,
      "transportprotocol":<String_value>,
      "ippooling":<String_value>,
      "mapping":<String_value>,
      "filtering":<String_value>,
      "tcpproxy":<String_value>,
      "td":<Double_value>,
      "l2info":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "lsnappsprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnappsprofile_binding

Jul 07, 2017

Binding object which returns the resources bound to lsnappsprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
appsprofilename	<String>	Read-write	Name for the LSN application profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN application profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn application profile1" or 'lsn application profile1'). Minimum length = 1 Maximum length = 127
lsnappsprofile_port_binding	<lsnappsprofile_port_binding[]>	Read-only	port that can be bound to lsnappsprofile.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile_binding/appsprofilename_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lsnappsprofile_binding": [ {
    "appsprofilename":<String_value>,
    "lsnappsprofile_port_binding":<lsnappsprofile_port_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnappsprofile_port_binding

May 20, 2016

Binding object showing the port that can be bound to lsnappsprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
appsprofilename	<String>	Read-write	Name for the LSN application profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN application profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn application profile1" or 'lsn application profile1'). Minimum length = 1 Maximum length = 127
lsnport	<String>	Read-write	Port numbers or range of port numbers to match against the destination port of the incoming packet from a subscriber. When the destination port is matched, the LSN application profile is applied for the LSN session. Separate a range of ports with a hyphen. For example, 40-90.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsnappsprofile_port_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsnappsprofile_port_binding":{
    "appsprofilename":<String_value>,
    "lsnport":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnappsprofile_port_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnappsprofile_port_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnappsprofile_port_binding": [ {
    "appsprofilename":<String_value>,
    "lsnport":<String_value>
}]}
```

count

URL: http://<netscaler-ip-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

address>/nitro/v1/config/lsnappsprofile_port_binding/appsprofilename_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lsnappsprofile_port_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnclient

Jul 07, 2017

Configuration for lsn client resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
td	<Double>	Read-only	ID of the traffic domain on which this subscriber or the subscriber network (as specified by the network parameter) belongs. If you do not specify an ID, the subscriber or the subscriber network becomes part of the default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnclient":{
    "clientname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient/clientname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnclient?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnclient?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnclient resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnclient?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnclient?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnclient resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "lsnclient": [ {
      "clientname":<String_value>,
      "td":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient/clientname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnclient/clientname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnclient/clientname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnclient": [ {
      "clientname":<String_value>,
      "td":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnclient": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnclient_binding

May 20, 2016

Binding object which returns the resources bound to lsnclient.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
lsnclient_network6_binding	<lsnclient_network6_binding[]>	Read-only	network6 that can be bound to lsnclient.
lsnclient_nsacl6_binding	<lsnclient_nsacl6_binding[]>	Read-only	nsacl6 that can be bound to lsnclient.
lsnclient_network_binding	<lsnclient_network_binding[]>	Read-only	network that can be bound to lsnclient.
lsnclient_nsacl_binding	<lsnclient_nsacl_binding[]>	Read-only	nsacl that can be bound to lsnclient.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_binding/clientname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lsnclient_binding": [ {
    "clientname":<String_value>,
    "lsnclient_nsacl6_binding":<lsnclient_nsacl6_binding[]_value>,
    "lsnclient_network_binding":<lsnclient_network_binding[]_value>,
    "lsnclient_nsacl_binding":<lsnclient_nsacl_binding[]_value>,
    "lsnclient_network6_binding":<lsnclient_network6_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnclient_network6_binding

Jul 07, 2017

Binding object showing the network6 that can be bound to lsnclient.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
network	<String>	Read-write	IPv4 address(es) of the LSN subscriber(s) or subscriber network(s) on whose traffic you want the NetScaler ADC to perform Large Scale NAT. Minimum length = 1
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
network6	<String>	Read-write	IPv6 address(es) of the LSN subscriber(s) or subscriber network(s) on whose traffic you want the NetScaler ADC to perform Large Scale NAT. Minimum length = 1
td	<Double>	Read-write	ID of the traffic domain on which this subscriber or the subscriber network (as specified by the network parameter) belongs. If you do not specify an ID, the subscriber or the subscriber network becomes part of the default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
netmask	<String>	Read-write	Subnet mask for the IPv4 address specified in the Network parameter.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsnclient_network6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsnclient_network6_binding":{
    "clientname":<String_value>,
    "network":<String_value>,
    "netmask":<String_value>,
    "network6":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnclient_network6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnclient_network6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnclient_network6_binding": [ {
      "network":<String_value>,
      "clientname":<String_value>,
      "network6":<String_value>,
      "td":<Double_value>,
      "netmask":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_network6_binding/clientname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lsnclient_network6_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnclient_network_binding

Jul 07, 2017

Binding object showing the network that can be bound to lsnclient.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
network	<String>	Read-write	IPv4 address(es) of the LSN subscriber(s) or subscriber network(s) on whose traffic you want the NetScaler ADC to perform Large Scale NAT. Minimum length = 1
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
td	<Double>	Read-write	ID of the traffic domain on which this subscriber or the subscriber network (as specified by the network parameter) belongs. If you do not specify an ID, the subscriber or the subscriber network becomes part of the default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
netmask	<String>	Read-write	Subnet mask for the IPv4 address specified in the Network parameter.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsnclient_network_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsnclient_network_binding":{
    "clientname":<String_value>,
    "network":<String_value>,
    "netmask":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnclient_network_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnclient_network_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "lsnclient_network_binding": [ {
      "network":<String_value>,
      "clientname":<String_value>,
      "td":<Double_value>,
      "netmask":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_network_binding/clientname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lsnclient_network_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnclient_nsacl6_binding

Jul 07, 2017

Binding object showing the nsacl6 that can be bound to lsnclient.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
acl6name	<String>	Read-write	Name of any configured extended ACL6 whose action is ALLOW. The condition specified in the extended ACL6 rule is used as the condition for the LSN client. Minimum length = 1
td	<Double>	Read-write	ID of the traffic domain on which this subscriber or the subscriber network (as specified by the network parameter) belongs. If you do not specify an ID, the subscriber or the subscriber network becomes part of the default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsnclient_nsacl6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsnclient_nsacl6_binding":{
    "clientname":<String_value>,
    "acl6name":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnclient_nsacl6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnclient_nsacl6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnclient_nsacl6_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "clientname":<String_value>,
    "acl6name":<String_value>,
    "td":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_nsacl6_binding/clientname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lsnclient_nsacl6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnclient_nsacl_binding

May 20, 2016

Binding object showing the nsacl that can be bound to lsnclient.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clientname	<String>	Read-write	Name for the LSN client entity. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN client is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn client1" or 'lsn client1'). . Minimum length = 1 Maximum length = 127
aclname	<String>	Read-write	Name(s) of any configured extended ACL(s) whose action is ALLOW. The condition specified in the extended ACL rule identifies the traffic from an LSN subscriber for which the NetScaler ADC is to perform large scale NAT. . Minimum length = 1
td	<Double>	Read-write	ID of the traffic domain on which this subscriber or the subscriber network (as specified by the network parameter) belongs. If you do not specify an ID, the subscriber or the subscriber network becomes part of the default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsnclient_nsacl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsnclient_nsacl_binding":{
    "clientname":<String_value>,
    "aclname":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnclient_nsacl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnclient_nsacl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnclient_nsacl_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "clientname":<String_value>,
    "aclname":<String_value>,
    "td":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnclient_nsacl_binding/clientname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lsnclient_nsacl_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsndeterministicnat

Jul 07, 2017

Configuration for deterministic NAT resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clientname	<String>	Read-write	The name of the LSN Client.
network6	<String>	Read-write	IPv6 address of the LSN subscriber or B4 device.
subscrip	<String>	Read-write	The Client IP address.
td	<Double>	Read-write	The LSN client TD. Default value: 0 Minimum value = 0 Maximum value = 4094
natip	<String>	Read-write	The NAT IP address. Minimum length = 1
natprefix	<String>	Read-only	IPv6 address of the LSN subscriber network(B4-Device) on whose traffic the NetScaler ADC perform Large Scale NAT.
subscrip2	<String>	Read-only	The Subscriber IP address.
natip2	<String>	Read-only	The NAT IP address.
firstport	<Integer>	Read-only	The Application Protocol Port of the Profile. Minimum value = 1 Maximum value = 65535
lastport	<Integer>	Read-only	The Application Protocol Ending Port of the Profile. Minimum value = 1 Maximum value = 65535
srctd	<Double>	Read-only	The LSN client TD. Minimum value = 0 Maximum value = 4094
nattype	<String>	Read-only	Type of subscriber info(ex: nat44 or ds-lite). Default value: NAT44 Possible values = NAT44, DS-Lite
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsndeterministicnat

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/lsndeterministicnat?**args=clientname:<String_value>,network6:<String_value>,subscrip:<String_value>,td:<Double_value>,natip:<String_value>**

Use this query-parameter to get lsndeterministicnat resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsndeterministicnat?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsndeterministicnat?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsndeterministicnat resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsndeterministicnat?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsndeterministicnat?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsndeterministicnat resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsndeterministicnat": [ {
clientname:<String_value>,network6:<String_value>,subscrip:<String_value>,td:<Double_value>,natip:<String_value>      "natprefix":<String_value>,
      "subscrip2":<String_value>,
      "natip2":<String_value>,
      "firstport":<Integer_value>,
      "lastport":<Integer_value>,
      "srctd":<Double_value>,
      "nattype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsndeterministicnat?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "lsndeterministicnat": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup

May 20, 2016

Configuration for LSN group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
clientname	<String>	Read-write	Name of the LSN client entity to be associated with the LSN group. You can associate only one LSN client entity with an LSN group.You cannot remove this association or replace with another LSN client entity once the LSN group is created.
nattype	<String>	Read-write	Type of NAT IP address and port allocation (from the bound LSN pools) for subscribers: Available options function as follows: * Deterministic - Allocate a NAT IP address and a block of ports to each subscriber (of the LSN client bound to the LSN group). The NetScaler ADC sequentially allocates NAT resources to these subscribers. The NetScaler ADC assigns the first block of ports (block size determined by the port block size parameter of the LSN group) on the beginning NAT IP address to the beginning subscriber IP address. The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. In this case, the first port block on the next NAT address is used for the subscriber, and so on. Because each subscriber now receives a deterministic NAT IP address and a block of ports, a subscriber can be identified without any need for logging. For a connection, a subscriber can be identified based only on the NAT IP address and port, and the destination IP address and port. The maximum number of LSN subscribers allowed, globally, is 1 million. * Dynamic - Allocate a random NAT IP address and a port from the LSN NAT pool for a subscriber\?s connection. If port block allocation is enabled (in LSN pool) and a port block size is specified (in the LSN group), the NetScaler ADC allocates a random NAT IP address and a block of ports for a subscriber when it initiates a connection for the first time. The ADC allocates this NAT IP address and a port (from the allocated block of ports) for different connections from this subscriber. If all the ports are allocated (for different subscriber\?s connections) from the subscriber\?s allocated port block, the ADC allocates a new random port block for the subscriber. Default value: DYNAMIC

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
portblocksize	<Double>	Read-write	Size of the NAT port block to be allocated for each subscriber. To set this parameter for Dynamic NAT, you must enable the port block allocation parameter in the bound LSN pool. For Deterministic NAT, the port block allocation parameter is always enabled, and you cannot disable it. In Dynamic NAT, the NetScaler ADC allocates a random NAT port block, from the available NAT port pool of an NAT IP address, for each subscriber. For a subscriber, if all the ports are allocated from the subscriber\?s allocated port block, the ADC allocates a new random port block for the subscriber. The default port block size is 256 for Deterministic NAT, and 0 for Dynamic NAT. Default value: 0 Minimum value = 256 Maximum value = 65536
logging	<String>	Read-write	Log mapping entries and sessions created or deleted for this LSN group. The NetScaler ADC logs LSN sessions for this LSN group only when both logging and session logging parameters are enabled. The ADC uses its existing syslog and audit log framework to log LSN information. You must enable global level LSN logging by enabling the LSN parameter in the related NSLOG action and SYLOG action entities. When the Logging parameter is enabled, the NetScaler ADC generates log messages related to LSN mappings and LSN sessions of this LSN group. The ADC then sends these log messages to servers associated with the NSLOG action and SYSLOG actions entities. A log message for an LSN mapping entry consists of the following information: * NSIP address of the NetScaler ADC * Time stamp * Entry type (MAPPING or SESSION) * Whether the LSN mapping entry is created or deleted * Subscriber's IP address, port, and traffic domain ID * NAT IP address and port * Protocol name * Destination IP address, port, and traffic domain ID might be present, depending on the following conditions: ** Destination IP address and port are not logged for Endpoint-Independent mapping ** Only Destination IP address (and not port) is logged for Address-Dependent mapping ** Destination IP address and port are logged for Address-Port-Dependent mapping. Default value: DISABLED Possible values = ENABLED, DISABLED
sessionlogging	<String>	Read-write	Log sessions created or deleted for the LSN group. The NetScaler ADC logs LSN sessions for this LSN group only when both logging and session logging parameters are enabled. A log message for an LSN session consists of the following information: * NSIP address of the NetScaler ADC * Time stamp * Entry type (MAPPING or SESSION) * Whether the LSN session is created or removed * Subscriber's IP address, port, and traffic domain ID

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			* Protocol name * Destination IP address, port, and traffic domain ID. Default value: DISABLED Possible values = ENABLED, DISABLED
sessionsync	<String>	Read-write	In a high availability (HA) deployment, synchronize information of all LSN sessions related to this LSN group with the secondary node. After a failover, established TCP connections and UDP packet flows are kept active and resumed on the secondary node (new primary). For this setting to work, you must enable the global session synchronization parameter. Default value: ENABLED Possible values = ENABLED, DISABLED
snmptraplimit	<Double>	Read-write	Maximum number of SNMP Trap messages that can be generated for the LSN group in one minute. Default value: 100 Minimum value = 0 Maximum value = 10000
ftp	<String>	Read-write	Enable Application Layer Gateway (ALG) for the FTP protocol. For some application-layer protocols, the IP addresses and protocol port numbers are usually communicated in the packet?s payload. When acting as an ALG, the NetScaler changes the packet?s payload to ensure that the protocol continues to work over LSN. Note: The NetScaler ADC also includes ALG for ICMP and TFTP protocols. ALG for the ICMP protocol is enabled by default, and there is no provision to disable it. ALG for the TFTP protocol is disabled by default. ALG is enabled automatically for an LSN group when you bind a UDP LSN application profile, with endpoint-independent-mapping, endpoint-independent filtering, and destination port as 69 (well-known port for TFTP), to the LSN group. Default value: ENABLED Possible values = ENABLED, DISABLED
pptp	<String>	Read-write	Enable the PPTP Application Layer Gateway. Default value: ENABLED Possible values = ENABLED, DISABLED
sipalg	<String>	Read-write	Enable the SIP ALG. Default value: DISABLED Possible values = ENABLED, DISABLED
rtspalg	<String>	Read-write	Enable the RTSP ALG. Default value: DISABLED Possible values = ENABLED, DISABLED
ip6profile	<String>	Read-write	Name of the LSN ip6 profile to associate with the specified LSN group. An ip6 profile can be associated with a group only during group creation. By default, no LSN ip6 profile is associated with an LSN group during its creation. Only one ip6profile can be associated with a group. Minimum length = 1 Maximum length = 127

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Minimum value = 1
Name	**Data Type**	**Permissions**	**Description**
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsngroup":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "groupname":<String_value>,
    "clientname":<String_value>,
    "nattype":<String_value>,
    "portblocksize":<Double_value>,
    "logging":<String_value>,
    "sessionlogging":<String_value>,
    "sessionsync":<String_value>,
    "snmptraplimit":<Double_value>,
    "ftp":<String_value>,
    "pptp":<String_value>,
    "sipalg":<String_value>,
    "rtspalg":<String_value>,
    "ip6profile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsngroup":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "groupname":<String_value>,
    "portblocksize":<Double_value>,
    "logging":<String_value>,
    "sessionlogging":<String_value>,
    "sessionsync":<String_value>,
    "snmptraplimit":<Double_value>,
    "ftp":<String_value>,
    "pptp":<String_value>,
    "sipalg":<String_value>,
    "rtspalg":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsngroup":{
    "groupname":<String_value>,
    "portblocksize":true,
    "logging":true,
    "sessionlogging":true,
    "sessionsync":true,
    "snmptraplimit":true,
    "ftp":true,
    "pptp":true,
    "sipalg":true,
    "rtspalg":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsngroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsngroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsngroup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsngroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsngroup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsngroup resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup": [ {
      "groupname":<String_value>,
      "clientname":<String_value>,
      "nattype":<String_value>,
      "portblocksize":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "logging":<String_value>,
    "sessionlogging":<String_value>,
    "sessionsync":<String_value>,
    "snmptraplimit":<Double_value>,
    "ftp":<String_value>,
    "pptp":<String_value>,
    "groupid":<Double_value>,
    "sipalg":<String_value>,
    "rtspalg":<String_value>,
    "ip6profile":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup/groupname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsngroup/groupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsngroup/groupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup": [ {
    "groupname":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "clientname":<String_value>,
    "nattype":<String_value>,
    "portblocksize":<Double_value>,
    "logging":<String_value>,
    "sessionlogging":<String_value>,
    "sessionsync":<String_value>,
    "snmptraplimit":<Double_value>,
    "ftp":<String_value>,
    "pptp":<String_value>,
    "groupid":<Double_value>,
    "sipalg":<String_value>,
    "rtspalg":<String_value>,
    "ip6profile":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup_binding

May 20, 2016

Binding object which returns the resources bound to lsngroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
lsngroup_lsnsipalgprofile_binding	<lsngroup_lsnsipalgprofile_binding[]>	Read-only	lsnsipalgprofile that can be bound to lsngroup.
lsngroup_lsntransportprofile_binding	<lsngroup_lsntransportprofile_binding[]>	Read-only	lsntransportprofile that can be bound to lsngroup.
lsngroup_lsnlogprofile_binding	<lsngroup_lsnlogprofile_binding[]>	Read-only	lsnlogprofile that can be bound to lsngroup.
lsngroup_lsnhttphdrlogprofile_binding	<lsngroup_lsnhttphdrlogprofile_binding[]>	Read-only	lsnhttphdrlogprofile that can be bound to lsngroup.
lsngroup_lsnpool_binding	<lsngroup_lsnpool_binding[]>	Read-only	lsnpool that can be bound to lsngroup.
lsngroup_lsnrtspalgprofile_binding	<lsngroup_lsnrtspalgprofile_binding[]>	Read-only	lsnrtspalgprofile that can be bound to lsngroup.
lsngroup_lsnappsprofile_binding	<lsngroup_lsnappsprofile_binding[]>	Read-only	lsnappsprofile that can be bound to lsngroup.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

Name	Data Type	Permissions	Description

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_binding/groupname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "lsngroup_binding": [ {
    "groupname":<String_value>,
    "lsngroup_lsnhttphdrlogprofile_binding":<lsngroup_lsnhttphdrlogprofile_binding[]_value>,
    "lsngroup_lsnrtspalgprofile_binding":<lsngroup_lsnrtspalgprofile_binding[]_value>,
    "lsngroup_lsnsipalgprofile_binding":<lsngroup_lsnsipalgprofile_binding[]_value>,
    "lsngroup_lsnlogprofile_binding":<lsngroup_lsnlogprofile_binding[]_value>,
    "lsngroup_lsnappsprofile_binding":<lsngroup_lsnappsprofile_binding[]_value>,
    "lsngroup_lsnpool_binding":<lsngroup_lsnpool_binding[]_value>,
    "lsngroup_lsntransportprofile_binding":<lsngroup_lsntransportprofile_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup_lsnappsprofile_binding

May 20, 2016

Binding object showing the lsnappsprofile that can be bound to lsngroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
appsprofilename	<String>	Read-write	Name of the LSN application profile to bind to the specified LSN group. For each set of destination ports, bind a profile for each protocol for which you want to specify settings. By default, one LSN application profile with default settings for TCP, UDP, and ICMP protocols for all destination ports is bound to an LSN group during its creation. This profile is called a default application profile. When you bind an LSN application profile, with a specified set of destination ports, to an LSN group, the bound profile overrides the default LSN application profile for that protocol at that set of destination ports.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsngroup_lsnappsprofile_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsngroup_lsnappsprofile_binding":{
    "groupname":<String_value>,
    "appsprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnappsprofile_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsngroup_lsnappsprofile_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup_lsnappsprofile_binding": [ {
    "appsprofilename":<String_value>,
    "groupname":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnappsprofile_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lsngroup_lsnappsprofile_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup_lsnhttphdrlogprofile_binding

Jul 07, 2017

Binding object showing the lsnhttphdrlogprofile that can be bound to lsngroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
httphdrlogprofilename	<String>	Read-write	The name of the LSN HTTP header logging Profile.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsngroup_lsnhttphdrlogprofile_binding":{
    "groupname":<String_value>,
    "httphdrlogprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnhttphdrlogprofile_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsngroup_lsnhttphdrlogprofile_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup_lsnhttphdrlogprofile_binding": [ {
      "httphdrlogprofilename":<String_value>,
      "groupname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnhttphdrlogprofile_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lsngroup_lsnhttphdrlogprofile_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnpool

Jul 07, 2017

Configuration for LSN pool resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
poolname	<String>	Read-write	Name for the LSN pool. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN pool is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn pool1" or 'lsn pool1'). Minimum length = 1 Maximum length = 127
nattype	<String>	Read-write	Type of NAT IP address and port allocation (from the LSN pools bound to an LSN group) for subscribers (of the LSN client entity bound to the LSN group): Available options function as follows: * Deterministic - Allocate a NAT IP address and a block of ports to each subscriber (of the LSN client bound to the LSN group). The NetScaler ADC sequentially allocates NAT resources to these subscribers. The NetScaler ADC assigns the first block of ports (block size determined by the port block size parameter of the LSN group) on the beginning NAT IP address to the beginning subscriber IP address. The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. In this case, the first port block on the next NAT address is used for the subscriber, and so on. Because each subscriber now receives a deterministic NAT IP address and a block of ports, a subscriber can be identified without any need for logging. For a connection, a subscriber can be identified based only on the NAT IP address and port, and the destination IP address and port. * Dynamic - Allocate a random NAT IP address and a port from the LSN NAT pool for a subscriber\?s connection. If port block allocation is enabled (in LSN pool) and a port block size is specified (in the LSN group), the NetScaler ADC allocates a random NAT IP address and a block of ports for a subscriber when it initiates a connection for the first time. The ADC allocates this NAT IP address and a port (from the allocated block of ports) for different connections from this subscriber. If all the ports are allocated (for different subscriber\?s connections) from the subscriber\?s allocated port block, the ADC allocates a new random port block for the subscriber. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group. A maximum of 16 LSN pools can be bound to an LSN group. .

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = DYNAMIC, DETERMINISTIC
portblockallocation	<String>	Read-write	Allocate a random NAT port block, from the available NAT port pool of an NAT IP address, for each subscriber when the NAT allocation is set as Dynamic NAT. For any connection initiated from a subscriber, the NetScaler ADC allocates a NAT port from the subscriber\?s allocated NAT port block to create the LSN session. You must set the port block size in the bound LSN group. For a subscriber, if all the ports are allocated from the subscriber\?s allocated port block, the NetScaler ADC allocates a new random port block for the subscriber. For Deterministic NAT, this parameter is enabled by default, and you cannot disable it. Default value: DISABLED Possible values = ENABLED, DISABLED
portrealloctimeout	<Double>	Read-write	The waiting time, in seconds, between deallocating LSN NAT ports (when an LSN mapping is removed) and reallocating them for a new LSN session. This parameter is necessary in order to prevent collisions between old and new mappings and sessions. It ensures that all established sessions are broken instead of redirected to a different subscriber. This is not applicable for ports used in: * Deterministic NAT * Address-Dependent filtering and Address-Port-Dependent filtering * Dynamic NAT with port block allocation In these cases, ports are immediately reallocated. Default value: 0 Minimum value = 0 Maximum value = 600
maxportrealloctmq	<Double>	Read-write	Maximum number of ports for which the port reallocation timeout applies for each NAT IP address. In other words, the maximum deallocated-port queue size for which the reallocation timeout applies for each NAT IP address. When the queue size is full, the next port deallocated is reallocated immediately for a new LSN session. Default value: 65536 Minimum value = 0 Maximum value = 65536
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnpool":{
    "poolname":<String_value>,
    "nattype":<String_value>,
    "portblockallocation":<String_value>,
    "portrealloctimeout":<Double_value>,
    "maxportrealloctmq":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool/poolname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnpool":{
    "poolname":<String_value>,
    "portrealloctimeout":<Double_value>,
    "maxportrealloctmq":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"lsnpool":{
    "poolname":<String_value>,
    "portrealloctimeout":true,
    "maxportrealloctmq":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnpool?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnpool?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnpool resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnpool?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnpool?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnpool resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnpool": [ {
      "poolname":<String_value>,
      "nattype":<String_value>,
      "portrealloctimeout":<Double_value>,
      "maxportrealloctmq":<Double_value>,
      "portblockallocation":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool/poolname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnpool/poolname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnpool/poolname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnpool": [ {
      "poolname":<String_value>,
      "nattype":<String_value>,
      "portrealloctimeout":<Double_value>,
      "maxportrealloctmq":<Double_value>,
      "portblockallocation":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnpool": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup_lsnlogprofile_binding

May 20, 2016

Binding object showing the lsnlogprofile that can be bound to lsngroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
logprofilename	<String>	Read-write	The name of the LSN logging Profile.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsngroup_lsnlogprofile_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsngroup_lsnlogprofile_binding":{
    "groupname":<String_value>,
    "logprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnlogprofile_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsngroup_lsnlogprofile_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup_lsnlogprofile_binding": [ {
    "logprofilename":<String_value>,
    "groupname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnlogprofile_binding/groupname_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lsngroup_lsnlogprofile_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup_lsnpool_binding

May 20, 2016

Binding object showing the lsnpool that can be bound to lsngroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
poolname	<String>	Read-write	Name of the LSN pool to bind to the specified LSN group. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group. For Deterministic NAT, pools bound to an LSN group cannot be bound to other LSN groups. For Dynamic NAT, pools bound to an LSN group can be bound to multiple LSN groups.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsngroup_lsnpool_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsngroup_lsnpool_binding":{
    "groupname":<String_value>,
    "poolname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnpool_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsngroup_lsnpool_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup_lsnpool_binding": [ {
      "poolname":<String_value>,
      "groupname":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnpool_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lsngroup_lsnpool_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup_lsnrtspalgprofile_binding

May 20, 2016

Binding object showing the lsnrtspalgprofile that can be bound to lsngroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
rtspalgprofilename	<String>	Read-write	The name of the LSN RTSP ALG Profile.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsngroup_lsnrtspalgprofile_binding":{
    "groupname":<String_value>,
    "rtspalgprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnrtspalgprofile_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsngroup_lsnrtspalgprofile_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup_lsnrtspalgprofile_binding": [ {
      "rtspalgprofilename":<String_value>,
      "groupname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnrtspalgprofile_binding/groupname_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lsngroup_lsnrtspalgprofile_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup_lsnsipalgprofile_binding

May 20, 2016

Binding object showing the lsnsipalgprofile that can be bound to lsngroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sipalgprofilename	<String>	Read-write	The name of the LSN SIP ALG Profile.
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsngroup_lsnsipalgprofile_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsngroup_lsnsipalgprofile_binding":{
    "groupname":<String_value>,
    "sipalgprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsnsipalgprofile_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsngroup_lsnsipalgprofile_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup_lsnsipalgprofile_binding": [ {
    "sipalgprofilename":<String_value>,
    "groupname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsnsipalgprofile_binding/groupname_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lsngroup_lsnsipalgprofile_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup_lsntransportprofile_binding

May 20, 2016

Binding object showing the lsntransportprofile that can be bound to lsngroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
groupname	<String>	Read-write	Name for the LSN group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn group1" or 'lsn group1'). Minimum length = 1 Maximum length = 127
transportprofilename	<String>	Read-write	Name of the LSN transport profile to bind to the specified LSN group. Bind a profile for each protocol for which you want to specify settings. By default, one LSN transport profile with default settings for TCP, UDP, and ICMP protocols is bound to an LSN group during its creation. This profile is called a default transport. An LSN transport profile that you bind to an LSN group overrides the default LSN transport profile for that protocol.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsngroup_lsntransportprofile_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsngroup_lsntransportprofile_binding":{
    "groupname":<String_value>,
    "transportprofilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsngroup_lsntransportprofile_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsngroup_lsntransportprofile_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup_lsntransportprofile_binding": [ {
      "groupname":<String_value>,
      "transportprofilename":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsngroup_lsntransportprofile_binding/groupname_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lsngroup_lsntransportprofile_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnhttphdrlogprofile

May 20, 2016

Configuration for LSN HTTP header logging Profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
httphdrlogprofilename	<String>	Read-write	The name of the HTTP header logging Profile. Minimum length = 1 Maximum length = 127
logurl	<String>	Read-write	URL information is logged if option is enabled. Default value: ENABLED Possible values = ENABLED, DISABLED
logmethod	<String>	Read-write	HTTP method information is logged if option is enabled. Default value: ENABLED Possible values = ENABLED, DISABLED
logversion	<String>	Read-write	Version information is logged if option is enabled. Default value: ENABLED Possible values = ENABLED, DISABLED
loghost	<String>	Read-write	Host information is logged if option is enabled. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnhttphdrlogprofile":{
    "httphdrlogprofilename":<String_value>,
    "logurl":<String_value>,
    "logmethod":<String_value>,
    "logversion":<String_value>,
    "loghost":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile/httphdrlogprofilename_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnhttphdrlogprofile":{
    "httphdrlogprofilename":<String_value>,
    "logurl":<String_value>,
    "logmethod":<String_value>,
    "logversion":<String_value>,
    "loghost":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnhttphdrlogprofile":{
    "httphdrlogprofilename":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "logurl":true,
        "logmethod":true,
        "logversion":true,
        "loghost":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnhttphdrlogprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnhttphdrlogprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnhttphdrlogprofile": [ {
      "httphdrlogprofilename":<String_value>,
      "logurl":<String_value>,
      "logmethod":<String_value>,
      "logversion":<String_value>,
      "loghost":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile/httphdrlogprofilename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile/httphdrlogprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile/httphdrlogprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnhttphdrlogprofile": [ {
      "httphdrlogprofilename":<String_value>,
      "logurl":<String_value>,
      "logmethod":<String_value>,
      "logversion":<String_value>,
      "loghost":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnhttphdrlogprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnhttphdrlogprofile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnip6profile

May 20, 2016

Configuration for LSN ip6 Profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the LSN ip6 profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN ip6 profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn ip6 profile1" or 'lsn ip6 profile1'). Minimum length = 1 Maximum length = 127
type	<String>	Read-write	IPv6 translation type for which to set the LSN IP6 profile parameters. Possible values = DS-Lite
network6	<String>	Read-write	IPv6 address of the NetScaler ADC AFTR device.
natprefix	<String>	Read-only	IPv6 address(es) of the LSN subscriber(s) or subscriber network(s) on whose traffic you want the NetScaler ADC to perform Large Scale NAT.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnip6profile":{
    "name":<String_value>,
    "type":<String_value>,
    "network6":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnip6profile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnip6profile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnip6profile": [ {
      "name":<String_value>,
      "type":<String_value>,
      "network6":<String_value>,
      "natprefix":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnip6profile": [ {
      "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "type":<String_value>,
    "network6":<String_value>,
    "natprefix":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnip6profile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnip6profile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnlogprofile

May 20, 2016

Configuration for LSN logging Profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
logprofilename	<String>	Read-write	The name of the logging Profile. Minimum length = 1 Maximum length = 127
logsubscrinfo	<String>	Read-write	Subscriber ID information is logged if option is enabled. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnlogprofile":{
    "logprofilename":<String_value>,
    "logsubscrinfo":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile/logprofilename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnlogprofile":{
    "logprofilename":<String_value>,
    "logsubscrinfo":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnlogprofile":{
    "logprofilename":<String_value>,
    "logsubscrinfo":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnlogprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnlogprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnlogprofile": [ {
      "logprofilename":<String_value>,
      "logsubscrinfo":<String_value>
}]}
```

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile/logprofilename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile/logprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile/logprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnlogprofile": [ {
      "logprofilename":<String_value>,
      "logsubscrinfo":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnlogprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "lsnlogprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnparameter

Jul 07, 2017

Configuration for LSN parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
memlimit	<Double>	Read-write	Amount of NetScaler memory to reserve for the LSN feature, in multiples of 2MB. Note: If you later reduce the value of this parameter, the amount of active memory is not reduced. Changing the configured memory limit can only increase the amount of active memory. This command is deprecated, use 'set extendedmemoryparam -memlimit' instead.
sessionsync	<String>	Read-write	Synchronize all LSN sessions with the secondary node in a high availability (HA) deployment (global synchronization). After a failover, established TCP connections and UDP packet flows are kept active and resumed on the secondary node (new primary). The global session synchronization parameter and session synchronization parameters (group level) of all LSN groups are enabled by default. For a group, when both the global level and the group level LSN session synchronization parameters are enabled, the primary node synchronizes information of all LSN sessions related to this LSN group with the secondary node. Default value: ENABLED Possible values = ENABLED, DISABLED
memlimitactive	<Double>	Read-only	Amount of actual memory reserved for the LSN feature. The amount of active memory for the LSN feature might be less than the configured memory, because the available memory is shared across features.
maxmemlimit	<Double>	Read-only	Maximum amount of NetScaler memory that can be reserved for the LSN feature.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnparameter":{
    "memlimit":<Double_value>,
    "sessionsync":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnparameter?**action=unset**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnparameter":{
    "memlimit":true,
    "sessionsync":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnparameter": [ {
    "memlimitactive":<Double_value>,
    "maxmemlimit":<Double_value>,
    "memlimit":<Double_value>,
    "sessionsync":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnpool_binding

May 20, 2016

Binding object which returns the resources bound to lsnpool.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
poolname	<String>	Read-write	Name for the LSN pool. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN pool is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn pool1" or 'lsn pool1'). Minimum length = 1 Maximum length = 127
lsnpool_lsnip_binding	<lsnpool_lsnip_binding[]>	Read-only	lsnip that can be bound to lsnpool.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool_binding/poolname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lsnpool_binding": [ {
      "poolname":<String_value>,
      "lsnpool_lsnip_binding":<lsnpool_lsnip_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnpool_lsnip_binding

May 20, 2016

Binding object showing the lsnip that can be bound to lsnpool.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
poolname	<String>	Read-write	Name for the LSN pool. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN pool is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn pool1" or 'lsn pool1'). Minimum length = 1 Maximum length = 127
lsnip	<String>	Read-write	IPv4 address or a range of IPv4 addresses to be used as NAT IP address(es) for LSN. After the pool is created, these IPv4 addresses are added to the NetScaler ADC as NetScaler owned IP address of type LSN. A maximum of 4096 IP addresses can be bound to an LSN pool. An LSN IP address associated with an LSN pool cannot be shared with other LSN pools. IP addresses specified for this parameter must not already exist on the NetScaler ADC as any NetScaler owned IP addresses. In the command line interface, separate the range with a hyphen. For example: 10.102.29.30-10.102.29.189. You can later remove some or all the LSN IP addresses from the pool, and add IP addresses to the LSN pool. . Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/lsnpool_lsnip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"lsnpool_lsnip_binding":{
    "poolname":<String_value>,
    "lsnip":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnpool_lsnip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnpool_lsnip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnpool_lsnip_binding": [ {
      "poolname":<String_value>,
      "lsnip":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnpool_lsnip_binding/poolname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"lsnpool_lsnip_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnrtspalgprofile

May 20, 2016

Configuration for LSN RTSPALG Profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
rtspalgprofilename	<String>	Read-write	The name of the RTSPALG Profile. Minimum length = 1 Maximum length = 127
rtspidletimeout	<Double>	Read-write	Idle timeout for the rtsp sessions in seconds. Default value: 120
rtspportrange	<String>	Read-write	port for the RTSP.
rtsptransportprotocol	<String>	Read-write	RTSP ALG Profile transport protocol type. Default value: TCP Possible values = TCP, UDP
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnrtspalgprofile":{
    "rtspalgprofilename":<String_value>,
    "rtspidletimeout":<Double_value>,
    "rtspportrange":<String_value>,
    "rtsptransportprotocol":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnrtspalgprofile":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "rtspalgprofilename":<String_value>,
    "rtspidletimeout":<Double_value>,
    "rtspportrange":<String_value>,
    "rtsptransportprotocol":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnrtspalgprofile":{
    "rtspalgprofilename":<String_value>,
    "rtspidletimeout":true,
    "rtspportrange":true,
    "rtsptransportprotocol":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile/rtspalgprofilename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnrtspalgprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnrtspalgprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnrtspalgprofile": [ {
      "rtspalgprofilename":<String_value>,
      "rtspidletimeout":<Double_value>,
      "rtspportrange":<String_value>,
      "rtsptransportprotocol":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile/rtspalgprofilename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile/rtspalgprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile/rtspalgprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "lsnrtspalgprofile": [ {
      "rtspalgprofilename":<String_value>,
      "rtspidletimeout":<Double_value>,
      "rtspportrange":<String_value>,
      "rtsptransportprotocol":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnrtspalgprofile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnrtspalgsession

Jul 07, 2017

Configuration for LSN RTSPALG session resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sessionid	<String>	Read-write	Session ID for the RTSP call.
callflags	<Double>	Read-only	Flags for the call entry.
callrefcount	<Double>	Read-only	Reference count for the call entry.
calltimer	<Double>	Read-only	Timer for the call entry.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

FLUSH | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

flush

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession?**action=flush**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnrtspalgsession":{
    "sessionid":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnrtspalgsession resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnrtspalgsession resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnrtspalgsession": [ {
    "sessionid":<String_value>,
    "callflags":<Double_value>,
    "callrefcount":<Double_value>,
    "calltimer":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession/sessionid_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession/sessionid_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession/sessionid_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnrtspalgsession": [ {
    "sessionid":<String_value>,
    "callflags":<Double_value>,
    "callrefcount":<Double_value>,
    "calltimer":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "lsnrtspalgsession": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnrtspalgsession_binding

May 20, 2016

Binding object which returns the resources bound to lsnrtspalgsession.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sessionid	<String>	Read-write	Session ID for the RTSP call.
lsnrtspalgsession_datachannel_binding	<lsnrtspalgsession_datachannel_binding[]>	Read-only	datachannel that can be bound to lsnrtspalgsession.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession_binding/sessionid_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lsnrtspalgsession_binding": [ {
      "sessionid":<String_value>,
      "lsnrtspalgsession_datachannel_binding":<lsnrtspalgsession_datachannel_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnrtspalgsession_datachannel_binding

May 20, 2016

Binding object showing the datachannel that can be bound to lsnrtspalgsession.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
channelip	<String>	Read-write	IP address of the channel.
sessionid	<String>	Read-write	Session ID for the RTSP call.
channelnatip	<String>	Read-only	Natted IP address of the channel.
channelport	<Integer>	Read-only	port for the channel.
channelflags	<Double>	Read-only	Flags for the call entry.
channeltimeout	<Double>	Read-only	Timeout for the channel.
channelnatport	<Integer>	Read-only	Natted port for the channel.
channelprotocol	<String>	Read-only	Channel transport protocol. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, HTTPSVR, HTTPCLIENT, NAT, HA, AAA, SINCTCP, VPNAFTP, MONITORS, SSLVPNUDP, SINCUDP, RIP, UDP CLNT, SASP, RPCCLNTS, ROUTE, AUDIT, STA HTTP, STA SSL, DNS RESOLVE, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, ICA, RADIUS, RADIUSListener, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, TFTP_DATA
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession_datachannel_binding/sessionid_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession_datachannel_binding/sessionid_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnrtspalgsession_datachannel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession_datachannel_binding/sessionid_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnrtspalgsession_datachannel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnrtspalgsession_datachannel_binding": [ {
    "channelip":<String_value>,
    "sessionid":<String_value>,
    "channelnatip":<String_value>,
    "channelport":<Integer_value>,
    "channelflags":<Double_value>,
    "channeltimeout":<Double_value>,
    "channelnatport":<Integer_value>,
    "channelprotocol":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnrtspalgsession_datachannel_binding/sessionid_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lsnrtspalgsession_datachannel_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnsession

May 20, 2016

Configuration for lsn session resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
nattype	<String>	Read-write	Type of sessions to be displayed. Default value: NAT44 Possible values = NAT44, DS-Lite
clientname	<String>	Read-write	Name of the LSN Client entity.
network	<String>	Read-write	IP address or network address of subscriber(s).
netmask	<String>	Read-write	Subnet mask for the IP address specified by the network parameter.
network6	<String>	Read-write	IPv6 address of the LSN subscriber or B4 device.
td	<Double>	Read-write	Traffic domain ID of the LSN client entity. Default value: 0 Minimum value = 0 Maximum value = 4094
natip	<String>	Read-write	Mapped NAT IP address used in LSN sessions.
natport2	<Integer>	Read-write	Mapped NAT port used in the LSN sessions.
natprefix	<String>	Read-only	IPv6 address of the LSN subscriber(s) or subscriber network(B4-Device) on whose traffic the NetScaler ADC perform Large Scale NAT.
subscrip	<String>	Read-only	The Source IP address.
subscrport	<Integer>	Read-only	The Source Port.
destip	<String>	Read-only	The Destination IP address.
destport	<Integer>	Read-only	The Destination Port.
natport	<Integer>	Read-only	The NAT Port.
transportprotocol	<String>	Read-only	The Transport Protocol for the session. Possible values = TCP, UDP, ICMP
sessionestdir	<String>	Read-only	The Session establishment direction, session was established for outbound or inbound packet. Possible values = OUT, IN
dsttd	<Double>	Read-only	.
srctd	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

FLUSH | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

flush

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsession?**action=flush**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnsession":{
    "nattype":<String_value>,
    "clientname":<String_value>,
    "network":<String_value>,
    "netmask":<String_value>,
    "network6":<String_value>,
    "td":<Double_value>,
    "natip":<String_value>,
    "natport2":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsession

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/lsnsession?**args=nattype:<String_value>,clientname:<String_value>,network:<String_value>,netmask:<String_value>,network6:<String_value>,td:<Double**

Use this query-parameter to get lsnsession resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnsession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnsession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnsession resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnsession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnsession?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnsession resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnsession": [ {
nattype:<String_value>,clientname:<String_value>,network:<String_value>,netmask:<String_value>,network6:<String_value>,td:<Double_value>,natip:<String_value>      "natprefix":<String_value>,
      "subscrip":<String_value>,
      "subscrport":<Integer_value>,
      "destip":<String_value>,
      "destport":<Integer_value>,
      "natport":<Integer_value>,
      "transportprotocol":<String_value>,
      "sessionestdir":<String_value>,
      "dsttd":<Double_value>,
      "srctd":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsession?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnsession": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnsipalgcall

May 20, 2016

Configuration for LSN SIPALG call resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
callid	<String>	Read-write	Call ID for the SIP call.
callflags	<Double>	Read-only	Flags for the call entry.
callrefcount	<Double>	Read-only	Reference count for the call entry.
calltimer	<Double>	Read-only	Timer for the call entry.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

FLUSH | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

flush

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall?**action=flush**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnsipalgcall":{
    "callid":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnsipalgcall resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnsipalgcall resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnsipalgcall": [ {
      "callid":<String_value>,
      "callflags":<Double_value>,
      "callrefcount":<Double_value>,
      "calltimer":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall/callid_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall/callid_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall/callid_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnsipalgcall": [ {
      "callid":<String_value>,
      "callflags":<Double_value>,
      "callrefcount":<Double_value>,
      "calltimer":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "lsnsipalgcall": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnsipalgcall_binding

May 20, 2016

Binding object which returns the resources bound to lsnsipalgcall.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
callid	<String>	Read-write	Call ID for the SIP call.
lsnsipalgcall_datachannel_binding	<lsnsipalgcall_datachannel_binding[]>	Read-only	datachannel that can be bound to lsnsipalgcall.
lsnsipalgcall_controlchannel_binding	<lsnsipalgcall_controlchannel_binding[]>	Read-only	controlchannel that can be bound to lsnsipalgcall.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall_binding/callid_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "lsnsipalgcall_binding": [ {
      "callid":<String_value>,
      "lsnsipalgcall_controlchannel_binding":<lsnsipalgcall_controlchannel_binding[]_value>,
      "lsnsipalgcall_datachannel_binding":<lsnsipalgcall_datachannel_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnsipalgcall_controlchannel_binding

May 20, 2016

Binding object showing the controlchannel that can be bound to lsnsipalgcall.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
channelip	<String>	Read-write	IP address of the channel.
callid	<String>	Read-write	Call ID for the SIP call.
channelnatip	<String>	Read-only	Natted IP address of the channel.
channelport	<Integer>	Read-only	port for the channel.
channelflags	<Double>	Read-only	Flags for the call entry.
channeltimeout	<Double>	Read-only	Timeout for the channel.
channelnatport	<Integer>	Read-only	Natted port for the channel.
channelprotocol	<String>	Read-only	Channel transport protocol. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, HTTPSVR, HTTPCLIENT, NAT, HA, AAA, SINCTCP, VPNAFTP, MONITORS, SSLVPNUDP, SINCUDP, RIP, UDP CLNT, SASP, RPCCLNTS, ROUTE, AUDIT, STA HTTP, STA SSL, DNS RESOLVE, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, ICA, RADIUS, RADIUSListener, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, TFTP_DATA
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall_controlchannel_binding/callid_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall_controlchannel_binding/callid_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnsipalgcall_controlchannel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall_controlchannel_binding/callid_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnsipalgcall_controlchannel_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnsipalgcall_controlchannel_binding": [ {
    "channelip":<String_value>,
    "callid":<String_value>,
    "channelnatip":<String_value>,
    "channelport":<Integer_value>,
    "channelflags":<Double_value>,
    "channeltimeout":<Double_value>,
    "channelnatport":<Integer_value>,
    "channelprotocol":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall_controlchannel_binding/callid_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lsnsipalgcall_controlchannel_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnsipalgcall_datachannel_binding

May 20, 2016

Binding object showing the datachannel that can be bound to lsnsipalgcall.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
channelip	<String>	Read-write	IP address of the channel.
callid	<String>	Read-write	Call ID for the SIP call.
channelnatip	<String>	Read-only	Natted IP address of the channel.
channelport	<Integer>	Read-only	port for the channel.
channelflags	<Double>	Read-only	Flags for the call entry.
channeltimeout	<Double>	Read-only	Timeout for the channel.
channelnatport	<Integer>	Read-only	Natted port for the channel.
channelprotocol	<String>	Read-only	Channel transport protocol. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, HTTPSVR, HTTPCLIENT, NAT, HA, AAA, SINCTCP, VPNAFTP, MONITORS, SSLVPNUDP, SINCUDP, RIP, UDP CLNT, SASP, RPCCLNTS, ROUTE, AUDIT, STA HTTP, STA SSL, DNS RESOLVE, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, ICA, RADIUS, RADIUSListener, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, TFTP_DATA
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall_datachannel_binding/callid_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall_datachannel_binding/callid_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of lsnsipalgcall_datachannel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall_datachannel_binding/callid_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnsipalgcall_datachannel_binding resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnsipalgcall_datachannel_binding": [ {
    "channelip":<String_value>,
    "callid":<String_value>,
    "channelnatip":<String_value>,
    "channelport":<Integer_value>,
    "channelflags":<Double_value>,
    "channeltimeout":<Double_value>,
    "channelnatport":<Integer_value>,
    "channelprotocol":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgcall_datachannel_binding/callid_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"lsnsipalgcall_datachannel_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnsipalgprofile

May 20, 2016

Configuration for LSN SIPALG Profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sipalgprofilename	<String>	Read-write	The name of the SIPALG Profile. Minimum length = 1 Maximum length = 127
datasessionidletimeout	<Double>	Read-write	Idle timeout for the data channel sessions in seconds. Default value: 120
sipsessiontimeout	<Double>	Read-write	SIP control channel session timeout in seconds. Default value: 600
registrationtimeout	<Double>	Read-write	SIP registration timeout in seconds. Default value: 60
sipsrcportrange	<String>	Read-write	Source port range for SIP_UDP and SIP_TCP.
sipdstportrange	<String>	Read-write	Destination port range for SIP_UDP and SIP_TCP.
openregisterpinhole	<String>	Read-write	ENABLE/DISABLE RegisterPinhole creation. Default value: ENABLED Possible values = ENABLED, DISABLED
opencontactpinhole	<String>	Read-write	ENABLE/DISABLE ContactPinhole creation. Default value: ENABLED Possible values = ENABLED, DISABLED
openviapinhole	<String>	Read-write	ENABLE/DISABLE ViaPinhole creation. Default value: ENABLED Possible values = ENABLED, DISABLED
openrecordroutepinhole	<String>	Read-write	ENABLE/DISABLE RecordRoutePinhole creation. Default value: ENABLED Possible values = ENABLED, DISABLED
siptransportprotocol	<String>	Read-write	SIP ALG Profile transport protocol type. Possible values = TCP, UDP
openroutepinhole	<String>	Read-write	ENABLE/DISABLE RoutePinhole creation. Default value: ENABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
rport	<String>	Read-write	ENABLE/DISABLE rport. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"lsnsipalgprofile":{
    "sipalgprofilename":<String_value>,
    "datasessionidletimeout":<Double_value>,
    "sipsessiontimeout":<Double_value>,
    "registrationtimeout":<Double_value>,
    "sipsrcportrange":<String_value>,
    "sipdstportrange":<String_value>,
    "openregisterpinhole":<String_value>,
    "opencontactpinhole":<String_value>,
    "openviapinhole":<String_value>,
    "openrecordroutepinhole":<String_value>,
    "siptransportprotocol":<String_value>,
    "openroutepinhole":<String_value>,
    "rport":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnsipalgprofile":{
    "sipalgprofilename":<String_value>,
    "datasessionidletimeout":<Double_value>,
    "sipsessiontimeout":<Double_value>,
    "registrationtimeout":<Double_value>,
    "sipsrcportrange":<String_value>,
    "sipdstportrange":<String_value>,
    "openregisterpinhole":<String_value>,
    "opencontactpinhole":<String_value>,
    "openviapinhole":<String_value>,
    "openrecordroutepinhole":<String_value>,
    "siptransportprotocol":<String_value>,
    "openroutepinhole":<String_value>,
    "rport":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnsipalgprofile":{
    "sipalgprofilename":<String_value>,
    "datasessionidletimeout":true,
    "sipsessiontimeout":true,
    "registrationtimeout":true,
    "sipsrcportrange":true,
    "sipdstportrange":true,
    "openregisterpinhole":true,
    "opencontactpinhole":true,
    "openviapinhole":true,
    "openrecordroutepinhole":true,
    "siptransportprotocol":true,
    "openroutepinhole":true,
    "rport":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile/sipalgprofilename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnsipalgprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnsipalgprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnsipalgprofile": [ {
    "sipalgprofilename":<String_value>,
    "datasessionidletimeout":<Double_value>,
    "sipsessiontimeout":<Double_value>,
    "registrationtimeout":<Double_value>,
    "sipsrcportrange":<String_value>,
    "sipdstportrange":<String_value>,
    "openregisterpinhole":<String_value>,
    "opencontactpinhole":<String_value>,
    "openviapinhole":<String_value>,
    "openrecordroutepinhole":<String_value>,
    "siptransportprotocol":<String_value>,
    "openroutepinhole":<String_value>,
    "rport":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile/sipalgprofilename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile/sipalgprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile/sipalgprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnsipalgprofile": [ {
    "sipalgprofilename":<String_value>,
    "datasessionidletimeout":<Double_value>,
    "sipsessiontimeout":<Double_value>,
    "registrationtimeout":<Double_value>,
    "sipsrcportrange":<String_value>,
    "sipdstportrange":<String_value>,
    "openregisterpinhole":<String_value>,
    "opencontactpinhole":<String_value>,
    "openviapinhole":<String_value>,
    "openrecordroutepinhole":<String_value>,
    "siptransportprotocol":<String_value>,
    "openroutepinhole":<String_value>,
    "rport":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnsipalgprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnsipalgprofile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsnstatic

May 20, 2016

Configuration for static mapping resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the LSN static mapping entry. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn static1" or 'lsn static1'). Minimum length = 1 Maximum length = 127
transportprotocol	<String>	Read-write	Protocol for the LSN mapping entry. Possible values = TCP, UDP, ICMP
subscrip	<String>	Read-write	IPv4(NAT44 & DS-Lite)/IPv6(NAT64) address of an LSN subscriber for the LSN static mapping entry.
subscrport	<Integer>	Read-write	Port of the LSN subscriber for the LSN mapping entry. Minimum value = 0 Maximum value = 65535
network6	<String>	Read-write	B4 address in DS-Lite setup. Minimum length = 1
td	<Double>	Read-write	ID of the traffic domain to which the subscriber belongs. If you do not specify an ID, the subscriber is assumed to be a part of the default traffic domain. Default value: 0 Minimum value = 0 Maximum value = 4094
natip	<String>	Read-write	IPv4 address, already existing on the NetScaler ADC as type LSN, to be used as NAT IP address for this mapping entry.
natport	<Integer>	Read-write	NAT port for this LSN mapping entry.
destip	<String>	Read-write	Destination IP address for the LSN mapping entry.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			mapping entry is reachable from the NetScaler ADC.
dsttd	<Double>	Read-write	If you do not specify an ID, the destination IP address is assumed to be reachable through the default traffic domain, which has an ID of 0. Default value: 0 Minimum value = 0 Maximum value = 4094
nattype	<String>	Read-write	Type of sessions to be displayed. Possible values = NAT44, DS-Lite
status	<String>	Read-only	The status of the Mapping. Status could be Inactive, if mapping addition failed due to already existing dynamic/static mapping, port allocation failure. Possible values = ACTIVE, INACTIVE
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnstatic

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsnstatic":{
    "name":<String_value>,
    "transportprotocol":<String_value>,
    "subscrip":<String_value>,
    "subscrport":<Integer_value>,
    "network6":<String_value>,
    "td":<Double_value>,
    "natip":<String_value>,
    "natport":<Integer_value>,
    "destip":<String_value>,
    "dsttd":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnstatic/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnstatic

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/lsnstatic?**args=name:<String_value>,nattype:<String_value>**

Use this query-parameter to get lsnstatic resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnstatic?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsnstatic?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsnstatic resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnstatic?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsnstatic?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsnstatic resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnstatic": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
name:<String_value>,nattype:<String_value>      "transportprotocol":<String_value>,
      "subscrip":<String_value>,
      "subscrport":<Integer_value>,
      "network6":<String_value>,
      "natip":<String_value>,
      "natport":<Integer_value>,
      "td":<Double_value>,
      "destip":<String_value>,
      "dsttd":<Double_value>,
      "status":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnstatic/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsnstatic/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsnstatic/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnstatic": [ {
name:<String_value>,nattype:<String_value>      "transportprotocol":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "subscrip":<String_value>,
    "subscrport":<Integer_value>,
    "network6":<String_value>,
    "natip":<String_value>,
    "natport":<Integer_value>,
    "td":<Double_value>,
    "destip":<String_value>,
    "dsttd":<Double_value>,
    "status":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsnstatic?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsnstatic": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsntransportprofile

Jul 07, 2017

Configuration for LSN Transport Profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
transportprofilename	<String>	Read-write	Name for the LSN transport profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the LSN transport profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "lsn transport profile1" or 'lsn transport profile1'). Minimum length = 1 Maximum length = 127
transportprotocol	<String>	Read-write	Protocol for which to set the LSN transport profile parameters. Possible values = TCP, UDP, ICMP
sessiontimeout	<Double>	Read-write	Timeout, in seconds, for an idle LSN session. If an LSN session is idle for a time that exceeds this value, the NetScaler ADC removes the session. This timeout does not apply for a TCP LSN session when a FIN or RST message is received from either of the endpoints. . Default value: 120 Minimum value = 60
finrsttimeout	<Double>	Read-write	Timeout, in seconds, for a TCP LSN session after a FIN or RST message is received from one of the endpoints. If a TCP LSN session is idle (after the NetScaler ADC receives a FIN or RST message) for a time that exceeds this value, the NetScaler ADC removes the session. Since the LSN feature of the NetScaler ADC does not maintain state information of any TCP LSN sessions, this timeout accommodates the transmission of the FIN or RST, and ACK messages from the other endpoint so that both endpoints can properly close the connection. Default value: 30
stuntimeout	<Double>	Read-write	STUN protocol timeout. Default value: 600 Minimum value = 120 Maximum value = 1200

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
synidletimeout	<Double>	Read-write	Default value: 60 Minimum value = 30 Maximum value = 120
portquota	<Double>	Read-write	Maximum number of LSN NAT ports to be used at a time by each subscriber for the specified protocol. For example, each subscriber can be limited to a maximum of 500 TCP NAT ports. When the LSN NAT mappings for a subscriber reach the limit, the NetScaler ADC does not allocate additional NAT ports for that subscriber. Default value: 0 Minimum value = 0 Maximum value = 65535
sessionquota	<Double>	Read-write	Maximum number of concurrent LSN sessions allowed for each subscriber for the specified protocol. When the number of LSN sessions reaches the limit for a subscriber, the NetScaler ADC does not allow the subscriber to open additional sessions. Default value: 0 Minimum value = 0 Maximum value = 65535
groupsessionlimit	<Double>	Read-write	Maximum number of concurrent LSN sessions(for the specified protocol) allowed for all subscriber of a group to which this profile has bound. This limit will get split across the netscalers packet engines and rounded down. When the number of LSN sessions reaches the limit for a group in packet engine, the NetScaler ADC does not allow the subscriber of that group to open additional sessions through that packet engine. Default value: 0
portpreserveparity	<String>	Read-write	Enable port parity between a subscriber port and its mapped LSN NAT port. For example, if a subscriber initiates a connection from an odd numbered port, the NetScaler ADC allocates an odd numbered LSN NAT port for this connection. You must set this parameter for proper functioning of protocols that require the source port to be even or odd numbered, for example, in peer-to-peer applications that use RTP or RTCP protocol. Default value: DISABLED Possible values = ENABLED, DISABLED
portpreserverange	<String>	Read-write	If a subscriber initiates a connection from a well-known port (0-1023), allocate a NAT port from the well-known port range (0-1023) for this connection. For example, if a subscriber initiates a connection from port 80, the NetScaler ADC can allocate port 100 as the NAT port for this connection. This parameter applies to dynamic NAT without port block allocation. It also applies to Deterministic NAT if the range of ports allocated includes well-known ports. When all the well-known ports of all the available NAT IP addresses are used in different subscribers\? connections (LSN sessions), and a subscriber initiates a connection from a well-known port, the NetScaler ADC drops this connection. Default value: DISABLED Possible values = ENABLED, DISABLED
			Silently drop any non-SYN packets for connections for which there is no LSN-NAT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			If you disable this parameter, the NetScaler ADC accepts any non-SYN packets and creates a new LSN session entry for this connection.
syncheck	<String>	Read-write	Following are some reasons for the NetScaler ADC to receive such packets: * LSN session for a connection existed but the NetScaler ADC removed this session because the LSN session was idle for a time that exceeded the configured session timeout. * Such packets can be a part of a DoS attack. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsntransportprofile":{
    "transportprofilename":<String_value>,
    "transportprotocol":<String_value>,
    "sessiontimeout":<Double_value>,
    "finrsttimeout":<Double_value>,
    "stuntimeout":<Double_value>,
    "synidletimeout":<Double_value>,
    "portquota":<Double_value>,
    "sessionquota":<Double_value>,
    "groupsessionlimit":<Double_value>,
    "portpreserveparity":<String_value>,
    "portpreserverange":<String_value>,
    "syncheck":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile/transportprofilename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsntransportprofile":{
    "transportprofilename":<String_value>,
    "sessiontimeout":<Double_value>,
    "finrsttimeout":<Double_value>,
    "stuntimeout":<Double_value>,
    "synidletimeout":<Double_value>,
    "portquota":<Double_value>,
    "sessionquota":<Double_value>,
    "groupsessionlimit":<Double_value>,
    "portpreserveparity":<String_value>,
    "portpreserverange":<String_value>,
    "syncheck":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lsntransportprofile":{
    "transportprofilename":<String_value>,
    "sessiontimeout":true,
    "finrsttimeout":true,
    "stuntimeout":true,
    "synidletimeout":true,
    "portquota":true,
    "sessionquota":true,
    "groupsessionlimit":true,
    "portpreserveparity":true,
    "portpreserverange":true,
    "syncheck":true
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lsntransportprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lsntransportprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsntransportprofile": [ {
      "transportprofilename":<String_value>,
      "transportprotocol":<String_value>,
      "sessiontimeout":<Double_value>,
      "finrsttimeout":<Double_value>,
      "stuntimeout":<Double_value>,
      "synidletimeout":<Double_value>,
      "portquota":<Double_value>,
      "sessionquota":<Double_value>,
      "groupsessionlimit":<Double_value>,
      "portpreserveparity":<String_value>,
      "portpreserverange":<String_value>,
      "syncheck":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile/transportprofilename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile/transportprofilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile/transportprofilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsntransportprofile": [ {
    "transportprofilename":<String_value>,
    "transportprotocol":<String_value>,
    "sessiontimeout":<Double_value>,
    "finrsttimeout":<Double_value>,
    "stuntimeout":<Double_value>,
    "synidletimeout":<Double_value>,
    "portquota":<Double_value>,
    "sessionquota":<Double_value>,
    "groupsessionlimit":<Double_value>,
    "portpreserveparity":<String_value>,
    "portpreserverange":<String_value>,
    "syncheck":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lsntransportprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsntransportprofile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Network

May 20, 2016

Networking commands.

Name	Description
Interface	Configuration for interface
appalgparam	Configuration for AppAlg Param
arp	Configuration for arp
arpparam	Configuration for Global arp parameters
bridgegroup	Configuration for bridge group
bridgegroup_binding	Binding object showing the resources that can be bound to bridgegroup
bridgegroup_nsip6_binding	Binding object showing the nsip6 that can be bound to bridgegroup
bridgegroup_nsip_binding	Binding object showing the nsip that can be bound to bridgegroup
bridgegroup_vlan_binding	Binding object showing the vlan that can be bound to bridgegroup
bridgetable	Configuration for bridge table entry
channel	Configuration for channel
channel_binding	Binding object showing the resources that can be bound to channel
channel_interface_binding	Binding object showing the interface that can be bound to channel
ci	Configuration for "CI"
fis	Configuration for "FIS"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

fis_binding	
Name	**Description**
fis_channel_binding	Binding object showing the channel that can be bound to fis
fis_interface_binding	Binding object showing the interface that can be bound to fis
forwardingsession	Configuration for session forward
inat	Configuration for inbound nat
inatparam	Configuration for INAT parameter
interfacepair	Configuration for "Interface Pair"
ip6tunnel	Configuration for ip6 Tunnel
ip6tunnelparam	Configuration for ip6 tunnel parameter
ipset	Configuration for network ipset
ipset_binding	Binding object showing the resources that can be bound to ipset
ipset_nsip6_binding	Binding object showing the nsip6 that can be bound to ipset
ipset_nsip_binding	Binding object showing the nsip that can be bound to ipset
iptunnel	Configuration for ip Tunnel
iptunnelparam	Configuration for ip tunnel parameter
ipv6	Configuration for ip v6
l2param	Configuration for Layer 2 related parameter
l3param	Configuration for Layer 3 related parameter
l4param	Configuration for Layer 4 related parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lacp	
Name	**Description**
linkset	Configuration for link set
linkset_binding	Binding object showing the resources that can be bound to linkset
linkset_channel_binding	Binding object showing the channel that can be bound to linkset
linkset_interface_binding	Binding object showing the interface that can be bound to linkset
nat64	Configuration for nat64 config
nd6	Configuration for nd6
nd6ravariables	Configuration for nd6 Router Advertisment configuration variables
nd6ravariables_binding	Binding object showing the resources that can be bound to nd6ravariables
nd6ravariables_onlinkipv6prefix_binding	Binding object showing the onlinkipv6prefix that can be bound to nd6ravariables
netbridge	Configuration for network bridge
netbridge_binding	Binding object showing the resources that can be bound to netbridge
netbridge_iptunnel_binding	Binding object showing the iptunnel that can be bound to netbridge
netbridge_nsip6_binding	Binding object showing the nsip6 that can be bound to netbridge
netbridge_nsip_binding	Binding object showing the nsip that can be bound to netbridge
netbridge_vlan_binding	Binding object showing the vlan that can be bound to netbridge
netprofile	Configuration for Network profile
netprofile_binding	Binding object showing the resources that can be bound to netprofile
netprofile_natrule_binding	Binding object showing the natrule that can be bound to netprofile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

netprofile_srcportset_binding	Binding object showing the srcportset that can be bound to netprofile
Name	**Description**
onlinkipv6prefix	Configuration for on-link IPv6 global prefixes for Router Advertisment
ptp	Configuration for Precision Time Protocol
rnat	Configuration for RNAT configured route
rnat6	Configuration for IPv6 RNAT configured route
rnat6_binding	Binding object showing the resources that can be bound to rnat6
rnat6_nsip6_binding	Binding object showing the nsip6 that can be bound to rnat6
rnatglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to rnatglobal
rnatglobal_binding	Binding object showing the resources that can be bound to rnatglobal
rnatparam	Configuration for RNAT parameter
route	Configuration for route
route6	Configuration for route 6
rsskeytype	Configuration for RSS key type
vlan	Configuration for "VLAN"
vlan_binding	Binding object showing the resources that can be bound to vlan
vlan_channel_binding	Binding object showing the channel that can be bound to vlan
vlan_interface_binding	Binding object showing the interface that can be bound to vlan
vlan_linkset_binding	Binding object showing the linkset that can be bound to vlan
vlan_nsip6_binding	Binding object showing the nsip6 that can be bound to vlan

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
vlan_nsip_binding	Binding object showing the nsip that can be bound to vlan
vpath	Configuration for Vpath
vpathparam	Configuration for VpathParam
vrid	Configuration for Virtual Router ID
vrid6	Configuration for Virtual Router ID for IPv6
vrid6_binding	Binding object showing the resources that can be bound to vrid6
vrid6_interface_binding	Binding object showing the interface that can be bound to vrid6
vrid6_nsip6_binding	Binding object showing the nsip6 that can be bound to vrid6
vrid6_nsip_binding	Binding object showing the nsip that can be bound to vrid6
vrid_binding	Binding object showing the resources that can be bound to vrid
vrid_interface_binding	Binding object showing the interface that can be bound to vrid
vrid_nsip6_binding	Binding object showing the nsip6 that can be bound to vrid
vrid_nsip_binding	Binding object showing the nsip that can be bound to vrid
vridparam	Configuration for VR ID parameter
vxlan	Configuration for "VXLAN"
vxlan_binding	Binding object showing the resources that can be bound to vxlan
vxlan_iptunnel_binding	Binding object showing the iptunnel that can be bound to vxlan
vxlan_nsip6_binding	Binding object showing the nsip6 that can be bound to vxlan
vxlan_nsip_binding	Binding object showing the nsip that can be bound to vxlan

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appalgparam

May 20, 2016

Configuration for AppAlg Param resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
pptpgreidletimeout	<Double>	Read-write	Interval in sec, after which data sessions of PPTP GRE is cleared. Default value: 9000 Minimum value = 1 Maximum value = 9000

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/appalgparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appalgparam":{
    "pptpgreidletimeout":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/appalgparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"appalgparam":{
    "pptpgreidletimeout":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/appalgparam

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appalgparam": [ {
      "pptpgreidletimeout":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

arp

May 20, 2016

Configuration for arp resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipaddress	<String>	Read-write	IP address of the network device that you want to add to the ARP table. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
mac	<String>	Read-write	MAC address of the network device.
ifnum	<String>	Read-write	Interface through which the network device is accessible. Specify the interface in (slot/port) notation. For example, 1/3.
vxlan	<Double>	Read-write	ID of the VXLAN on which the IP address of this ARP entry is reachable. Minimum value = 1 Maximum value = 16777215
vtep	<String>	Read-write	IP address of the VXLAN tunnel endpoint (VTEP) through which the IP address of this ARP entry is reachable. Minimum length = 1
vlan	<Double>	Read-write	The VLAN ID through which packets are to be sent after matching the ARP entry. This is a numeric value.
ownernode	<Double>	Read-write	The owner node for the Arp entry. Minimum value = 0 Maximum value = 31
all	<Boolean>	Read-write	Remove all ARP entries from the ARP table of the NetScaler appliance.
timeout	<Double>	Read-only	The time, in seconds, after which the entry times out.
state	<Double>	Read-only	The state of the ARP entry.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
type	<String>	Read-only	Indicates whether this ARP entry was added manually or dynamically. When you manually add an ARP entry, the value for this parameter is STATIC. Otherwise, it is DYNAMIC. For the NSIP and loopback IP addresses, the value is PERMANENT. Possible values = STATIC, PERMANENT, DYNAMIC
channel	<Double>	Read-only	The tunnel, channel, or physical interface through which the ARP entry is identified.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | SEND | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/arp

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"arp":{
    "ipaddress":<String_value>,
    "td":<Double_value>,
    "mac":<String_value>,
    "ifnum":<String_value>,
    "vxlan":<Double_value>,
    "vtep":<String_value>,
    "vlan":<Double_value>,
    "ownernode":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/arp/ipaddress_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

send

URL: http://<netscaler-ip-address>/nitro/v1/config/arp?**action=send**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"arp":{
    "ipaddress":<String_value>,
    "td":<Double_value>,
    "all":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/arp

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/arp?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/arp?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of arp resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/arp?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/arp?**pagesize=#no&pageno=#no**

Use this query-parameter to get the arp resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "arp": [ {
    "ipaddress":<String_value>,
    "td":<Double_value>,
    "ownernode":<Double_value>,
    "mac":<String_value>,
    "ifnum":<String_value>,
    "timeout":<Double_value>,
    "state":<Double_value>,
    "flags":<Double_value>,
    "type":<String_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "vtep":<String_value>,
    "channel":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/arp?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "arp": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

arpparam

May 20, 2016

Configuration for Global arp parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
timeout	<Double>	Read-write	Time-out value (aging time) for the dynamically learned ARP entries, in seconds. The new value applies only to ARP entries that are dynamically learned after the new value is set. Previously existing ARP entries expire after the previously configured aging time. Default value: 1200 Minimum value = 5 Maximum value = 1200
spoofvalidation	<String>	Read-write	enable/disable arp spoofing validation. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/arpparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"arpparam":{
    "timeout":<Double_value>,
    "spoofvalidation":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/arpparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"arpparam":{
    "timeout":true,
    "spoofvalidation":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/arpparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "arpparam": [ {
      "timeout":<Double_value>,
      "spoofvalidation":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

bridgegroup

May 20, 2016

Configuration for bridge group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	An integer that uniquely identifies the bridge group. Minimum value = 1 Maximum value = 1000
dynamicrouting	<String>	Read-write	Enable dynamic routing for this bridgegroup. Default value: DISABLED Possible values = ENABLED, DISABLED
ipv6dynamicrouting	<String>	Read-write	Enable all IPv6 dynamic routing protocols on all VLANs bound to this bridgegroup. Note: For the ENABLED setting to work, you must configure IPv6 dynamic routing protocols from the VTYSH command line. Default value: DISABLED Possible values = ENABLED, DISABLED
flags	<Boolean>	Read-only	Temporary flag used for internal purpose.
portbitmap	<Double>	Read-only	Member interfaces of this bridge group.
tagbitmap	<Double>	Read-only	Tagged members of this bridge group.
ifaces	<String>	Read-only	Names of all member interfaces of this bridge group.
tagifaces	<String>	Read-only	Names of all tagged member interfaces of this bridge group.
rnat	<Boolean>	Read-only	Temporary flag used for internal purpose.
partitionname	<String>	Read-only	Name of the Partition to which this vlan bound to. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"bridgegroup":{
    "id":<Double_value>,
    "dynamicrouting":<String_value>,
    "ipv6dynamicrouting":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"bridgegroup":{
    "id":<Double_value>,
    "dynamicrouting":<String_value>,
    "ipv6dynamicrouting":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup?**action=unset**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"bridgegroup":{
    "id":<Double_value>,
    "dynamicrouting":true,
    "ipv6dynamicrouting":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of bridgegroup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the bridgegroup resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "bridgegroup": [ {
    "id":<Double_value>,
    "flags":<Boolean_value>,
    "portbitmap":<Double_value>,
    "tagbitmap":<Double_value>,
    "ifaces":<String_value>,
    "tagifaces":<String_value>,
    "dynamicrouting":<String_value>,
    "ipv6dynamicrouting":<String_value>,
    "rnat":<Boolean_value>,
    "partitionname":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup/id_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "bridgegroup": [ {
    "id":<Double_value>,
    "flags":<Boolean_value>,
    "portbitmap":<Double_value>,
    "tagbitmap":<Double_value>,
    "ifaces":<String_value>,
    "tagifaces":<String_value>,
    "dynamicrouting":<String_value>,
    "ipv6dynamicrouting":<String_value>,
    "rnat":<Boolean_value>,
    "partitionname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "bridgegroup": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

bridgegroup_binding

May 20, 2016

Binding object which returns the resources bound to bridgegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	The name of the bridge group. Minimum value = 1 Maximum value = 1000
bridgegroup_nsip_binding	<bridgegroup_nsip_binding[]>	Read-only	nsip that can be bound to bridgegroup.
bridgegroup_vlan_binding	<bridgegroup_vlan_binding[]>	Read-only	vlan that can be bound to bridgegroup.
bridgegroup_nsip6_binding	<bridgegroup_nsip6_binding[]>	Read-only	nsip6 that can be bound to bridgegroup.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_binding/id_value<Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "bridgegroup_binding": [ {
    "id":<Double_value>,
    "bridgegroup_vlan_binding":<bridgegroup_vlan_binding[]_value>,
    "bridgegroup_nsip_binding":<bridgegroup_nsip_binding[]_value>,
    "bridgegroup_nsip6_binding":<bridgegroup_nsip6_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

bridgegroup_nsip6_binding

Jul 07, 2017

Binding object showing the nsip6 that can be bound to bridgegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	The integer that uniquely identifies the bridge group. Minimum value = 1 Maximum value = 1000
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
netmask	<String>	Read-write	A subnet mask associated with the network address. Minimum length = 1
ipaddress	<String>	Read-write	The IP address assigned to the bridge group.
rnat	<Boolean>	Read-only	Temporary flag used for internal purpose.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/bridgegroup_nsip6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"bridgegroup_nsip6_binding":{
    "id":<Double_value>,
    "ipaddress":<String_value>,
    "netmask":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of bridgegroup_nsip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the bridgegroup_nsip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "bridgegroup_nsip6_binding": [ {
    "id":<Double_value>,
    "td":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "netmask":<String_value>,
    "ipaddress":<String_value>,
    "rnat":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_nsip6_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"bridgegroup_nsip6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

bridgegroup_nsip_binding

Jul 07, 2017

Binding object showing the nsip that can be bound to bridgegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	The integer that uniquely identifies the bridge group. Minimum value = 1 Maximum value = 1000
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
netmask	<String>	Read-write	The network mask for the subnet defined for the bridge group.
ipaddress	<String>	Read-write	The IP address assigned to the bridge group.
rnat	<Boolean>	Read-only	Temporary flag used for internal purpose.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/bridgegroup_nsip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"bridgegroup_nsip_binding":{
    "id":<Double_value>,
    "ipaddress":<String_value>,
    "netmask":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of bridgegroup_nsip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the bridgegroup_nsip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "bridgegroup_nsip_binding": [ {
    "id":<Double_value>,
    "td":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "netmask":<String_value>,
    "ipaddress":<String_value>,
    "rnat":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_nsip_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"bridgegroup_nsip_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

bridgegroup_vlan_binding

Jul 07, 2017

Binding object showing the vlan that can be bound to bridgegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	The integer that uniquely identifies the bridge group. Minimum value = 1 Maximum value = 1000
vlan	<Double>	Read-write	Names of all member VLANs.
rnat	<Boolean>	Read-only	Temporary flag used for internal purpose.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/bridgegroup_vlan_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"bridgegroup_vlan_binding":{
    "id":<Double_value>,
    "vlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of bridgegroup_vlan_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the bridgegroup_vlan_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "bridgegroup_vlan_binding": [ {
    "id":<Double_value>,
    "vlan":<Double_value>,
    "rnat":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgegroup_vlan_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"bridgegroup_vlan_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

bridgetable

May 20, 2016

Configuration for bridge table entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
bridgeage	<Double>	Read-write	Time-out value for the bridge table entries, in seconds. The new value applies only to the entries that are dynamically learned after the new value is set. Previously existing bridge table entries expire after the previously configured time-out value. Default value: 300 Minimum value = 60 Maximum value = 300
vlan	<Double>	Read-write	VLAN whose entries are to be removed. Minimum value = 1 Maximum value = 4094
ifnum	<String>	Read-write	INTERFACE whose entries are to be removed.
vxlan	<Double>	Read-write	VXLAN whose entries are to be removed. Minimum value = 1 Maximum value = 16777215
mac	<String>	Read-only	The MAC address of the target.
vtep	<String>	Read-only	The IP address of the VTEP.
flags	<Double>	Read-only	Display flags,.
channel	<Double>	Read-only	The Tunnel through which bridge entry is learned.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

UPDATE | UNSET | CLEAR | GET (ALL) | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgetable

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"bridgetable":{
    "bridgeage":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgetable?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"bridgetable":{
    "bridgeage":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgetable?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"bridgetable":{
    "vlan":<Double_value>,
    "ifnum":<String_value>,
    "vxlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgetable

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/bridgetable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/bridgetable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of bridgetable resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/bridgetable?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/bridgetable?**pagesize=#no&pageno=#no**

Use this query-parameter to get the bridgetable resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "bridgetable": [ {
      "bridgeage":<Double_value>,
      "mac":<String_value>,
      "ifnum":<String_value>,
      "vlan":<Double_value>,
      "vxlan":<Double_value>,
      "vtep":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "flags":<Double_value>,
    "channel":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/bridgetable?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "bridgetable": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

channel

May 20, 2016

Configuration for channel resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<String>	Read-write	ID for the LA channel or cluster LA channel or LR channel to be created. Specify an LA channel in LA/x notation, where x can range from 1 to 8 or cluster LA channel in CLA/x notation or Link redundant channel in LR/x notation, where x can range from 1 to 4. Cannot be changed after the LA channel is created.
ifnum	<String[]>	Read-write	Interfaces to be bound to the LA channel of a NetScaler appliance or to the LA channel of a cluster configuration. For an LA channel of a NetScaler appliance, specify an interface in C/U notation (for example, 1/3). For an LA channel of a cluster configuration, specify an interface in N/C/U notation (for example, 2/1/3). where C can take one of the following values: * 0 - Indicates a management interface. * 1 - Indicates a 1 Gbps port. * 10 - Indicates a 10 Gbps port. U is a unique integer for representing an interface in a particular port group. N is the ID of the node to which an interface belongs in a cluster configuration. Use spaces to separate multiple entries.
state	<String>	Read-write	Enable or disable the LA channel. Default value: ENABLED Possible values = ENABLED, DISABLED
mode	<String>	Read-write	The initital mode for the LA channel. Possible values = MANUAL, AUTO
conndistr	<String>	Read-write	The 'connection' distribution mode for the LA channel. Possible values = DISABLED, ENABLED
macdistr	<String>	Read-write	The 'MAC' distribution mode for the LA channel. Possible values = SOURCE, DESTINATION, BOTH
lamac	<String>	Read-write	Specifies a MAC address for the LA channels configured in NetScaler virtual appliances (VPX). This MAC address is persistent after each reboot. If you don't specify this parameter, a MAC address is generated randomly for each LA channel. These MAC addresses change after each reboot.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
speed	<String>	Read-write	greater than or equal to the value set for this parameter, the state of the interface is UP. Otherwise, the state is INACTIVE. Bound Interfaces whose state is INACTIVE do not process any traffic. Default value: AUTO Possible values = AUTO, 10, 100, 1000, 10000, 40000
flowctl	<String>	Read-write	Specifies the flow control type for this LA channel to manage the flow of frames. Flow control is a function as mentioned in clause 31 of the IEEE 802.3 standard. Flow control allows congested ports to pause traffic from the peer device. Flow control is achieved by sending PAUSE frames. Default value: OFF Possible values = OFF, RX, TX, RXTX
hamonitor	<String>	Read-write	In a High Availability (HA) configuration, monitor the LA channel for failure events. Failure of any LA channel that has HA MON enabled triggers HA failover. Default value: ON Possible values = ON, OFF
tagall	<String>	Read-write	Adds a four-byte 802.1q tag to every packet sent on this channel. The ON setting applies tags for all VLANs that are bound to this channel. OFF applies the tag for all VLANs other than the native VLAN. Default value: OFF Possible values = ON, OFF
trunk	<String>	Read-write	This is deprecated by tagall. Default value: OFF Possible values = ON, OFF
ifalias	<String>	Read-write	Alias name for the LA channel. Used only to enhance readability. To perform any operations, you have to specify the LA channel ID. Default value: " " Maximum length = 31
throughput	<Double>	Read-write	Low threshold value for the throughput of the LA channel, in Mbps. In an high availability (HA) configuration, failover is triggered when the LA channel has HA MON enabled and the throughput is below the specified threshold. Minimum value = 0 Maximum value = 160000
bandwidthhigh	<Double>	Read-write	High threshold value for the bandwidth usage of the LA channel, in Mbps. The NetScaler appliance generates an SNMP trap message when the bandwidth usage of the LA channel is greater than or equal to the specified high threshold value. Minimum value = 0 Maximum value = 160000
bandwidthnormal	<Double>	Read-write	Normal threshold value for the bandwidth usage of the LA channel, in Mbps. When the bandwidth usage of the LA channel returns to less than or equal to the specified normal threshold after exceeding the high threshold, the NetScaler appliance generates an SNMP trap message to indicate that the bandwidth usage has returned to normal. Minimum value = 0 Maximum value = 160000

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
mtu	<Double>	Read-write	The maximum transmission unit (MTU) is the largest packet size, measured in bytes excluding 14 bytes ethernet header and 4 bytes crc, that can be transmitted and received by this interface. Default value of MTU is 1500 on all the interface of Netscaler appliance any value configured more than 1500 on the interface will make the interface as jumbo enabled. In case of cluster backplane interface MTU value will be changed to 1514 by default, user has to change the backplane interface value to maximum mtu configured on any of the interface in cluster system plus 14 bytes more for backplane interface if Jumbo is enabled on any of the interface in a cluster system. Changing the backplane will bring back the MTU of backplane interface to default value of 1500. If a channel is configured as backplane then the same holds true for channel as well as member interfaces. In case of channel if member interfaces is configured as different mtu then the highest MTU configured MTU is treated as the LA MTU if MTU is not specified on LA explicitly. Low MTU interfaces in channel will be taken out of LA distribution list. Default value: 1500 Minimum value = 1500 Maximum value = 9216
lrminthroughput	<Double>	Read-write	Specifies the minimum throughput threshold (in Mbps) to be met by the active subchannel. Setting this parameter automatically divides an LACP channel into logical subchannels, with one subchannel active and the others in standby mode. When the maximum supported throughput of the active channel falls below the lrMinThroughput value, link failover occurs and a standby subchannel becomes active. Minimum value = 0 Maximum value = 80000
linkredundancy	<String>	Read-write	Link Redundancy for Cluster LAG. Default value: OFF Possible values = ON, OFF
devicename	<String>	Read-only	LA channel name in form LA/x, where x is channel ID, which ranges from 1 to 8 or LR channel name in form LR/x, where x is channel ID, which ranges from 1 to 4.
unit	<Double>	Read-only	Unit number of the channel. This is an internal reference number that the NetScaler uses to identify the channel.
description	<String>	Read-only	The IEEE standard that the channel is based on.
flags	<Double>	Read-only	Flags of this channel.
actualmtu	<Double>	Read-only	MTU of the channel. This is the maximum frame size that the channel can process.
vlan	<Double>	Read-only	Native VLAN of the channel.
mac	<String>	Read-only	MAC address of the channel.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			This value is reset when the channel state changes to DOWN.
downtime	<Double>	Read-only	Duration for which the channel is DOWN. (Example: 3 hours 1 minute 1 second). This value is reset when the channel state changes to UP.
reqmedia	<String>	Read-only	Requested media setting for this channel. Since there is no media associated with LA, the displayed values carry no significance. Possible values = AUTO, UTP, FIBER
reqspeed	<String>	Read-only	Requested speed setting for this channel. Since no media are associated with LA, this speed is used to determine the threshold for the slave interfaces. If the speed of the member interface is less than the requested speed, that interface is considered inactive. Possible values = AUTO, 10, 100, 1000, 10000, 40000
reqduplex	<String>	Read-only	Requested duplex setting for this channel. Since no media are associated with LA, the displayed values carry no significance. Possible values = AUTO, HALF, FULL
reqflowcontrol	<String>	Read-only	Requested flow control setting for this channel. Since no media are associated with LA, the displayed values carry no significance. Possible values = OFF, RX, TX, RXTX
media	<String>	Read-only	Requested media setting for this interface. Possible values = AUTO, UTP, FIBER
actspeed	<String>	Read-only	Actual speed setting for this channel. Possible values = AUTO, 10, 100, 1000, 10000, 40000
duplex	<String>	Read-only	Actualduplex setting for this interface. Possible values = AUTO, HALF, FULL
actflowctl	<String>	Read-only	Actual flow control setting for this channel. Possible values = OFF, RX, TX, RXTX
lamode	<String>	Read-only	The mode(AUTO/MANNUAL) for the LA channel. Possible values = MANUAL, AUTO
autoneg	<Double>	Read-only	Requested auto negotiation setting for this channel. Since no media are associated with LA, this setting has no effect.
autonegresult	<Double>	Read-only	Actual auto negotiation setting for this channel.
tagged	<Double>	Read-only	VLAN tags setting on this channel.
taggedany	<Double>	Read-only	Channel setting to accept/drop all tagged packets.
taggedautolearn	<Double>	Read-only	Dynaminc vlan membership on this channel.
hangdetect	<Double>	Read-only	Hang detect for this channel.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
hangreset	<Double>	Read-only	Hang reset for this channel.
linkstate	<Double>	Read-only	The current state of the link associated with the interface. For logical interfaces (LA), the state of the link is dependent on the state of the slave interfaces. For the link to be UP at least one of the slave interfaces needs to be UP.
intfstate	<Double>	Read-only	Current state of the specified interface. The interface state set to UP only if the link state is UP and administrative state is ENABLED.
rxpackets	<Double>	Read-only	Number of bytes received by all the slave interfaces of the channel since the NetScaler appliance was started or the interface statistics were cleared.
rxbytes	<Double>	Read-only	Number of packets received by all member interfaces since the NetScaler appliance was started or the interface statistics were cleared.
rxerrors	<Double>	Read-only	Number of inbound packets dropped by the hardware of the slave interfaces since the NetScaler appliance was started or the interface statistics were cleared. Possible causes of dropped packets are CRC, length (undersize or oversize), and alignment errors.
rxdrops	<Double>	Read-only	Number of inbound packets dropped by the channel's slave interfaces. Commonly dropped packets are multicast frames, spanning tree BPDUs, packets destined to a MAC not owned by the NetScaler when L2 mode is disabled, or packets tagged for a VLAN that is not bound to the interface. In most healthy networks, this statistic increments at a steady rate regardless of traffic load. A sharp spike in dropped packets generally indicates an issue with connected L2 switches, such as a forwarding database overflow resulting in packets being broadcast on all ports.
txpackets	<Double>	Read-only	Number of packets transmitted by slave interfaces of a channel since the NetScaler appliance was started or the interface statistics were cleared.
txbytes	<Double>	Read-only	Number of bytes transmitted by slave interfaces of a channel since the NetScaler appliance was started or the interface statistics were cleared.
txerrors	<Double>	Read-only	Number of outbound packets dropped by the hardware of a channel's slave interfaces since the NetScaler appliance was started or the interface statistics were cleared. Possible causes of dropped packets are length (undersize or oversize) errors and lack of resources.
txdrops	<Double>	Read-only	Number of packets dropped in transmission by a channel's slave interfaces for one of the following reasons: (1) VLAN mismatch. (2) Oversized packets. (3) Interface congestion. (4) Loopback packets sent on non-loopback interface.
indisc	<Double>	Read-only	Number of error-free inbound packets discarded by a channel's slave interfaces because of a lack of resources (for example, insufficient receive buffers).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			interfaces because of a lack of resources. This statistic is not available on: (1) 10G ports of NetScaler MPX 12500/12500/15500-10G platforms. (2) 10G data ports on NetScaler MPX 17500/19500/21500 platforms.
fctls	<Double>	Read-only	Number of times flow control is performed on a channel's slave interfaces because of pause frames.
hangs	<Double>	Read-only	Number of hangs that occurred on the channel's slave interfaces.
stsstalls	<Double>	Read-only	Number of status stalls that occurred on the channel's slave interfaces.
txstalls	<Double>	Read-only	Number of Tx stalls happened that occurred on the channel's slave interfaces.
rxstalls	<Double>	Read-only	Number of Rx stalls that occurred on the channel's slave interfaces.
bdgmuted	<Double>	Read-only	Number of times a channel's slave interfaces stopped transmitting and receiving packets because of MAC moves between ports.
vmac	<String>	Read-only	Virtual MAC of this channel.
vmac6	<String>	Read-only	Virtual MAC for IPv6 on this interface.
reqthroughput	<Double>	Read-only	Minimum required throughput for an interface. Failover is triggered if the operating throughput of a Link Aggregation (LA) channel for which HAMON is ON falls below this value. Minimum value = 0 Maximum value = 160000
actthroughput	<Double>	Read-only	Actual throughput for the interface.
backplane	<String>	Read-only	The cluster backplane status of the LA. If the status is enabled, the LA is part of the cluster backplane. By default, the backplane status is disabled. Possible values = ENABLED, DISABLED
cleartime	<Double>	Read-only	Time since the interface stats are cleared last time.
lacpmode	<String>	Read-only	The LACP mode of the specified interface. The possible values are: 1. Active: A port in active mode generates LACP protocol messages on a regular basis, regardless of any need expressed by its partner to receive them. 2. Passive: A port in passive mode generally does not transmit LACP messages unless its partner is in the active mode; that is, it does not speak unless spoken to. 3. Disabled: Removes the interface from the LA channel. If this is only interface in the LA channel, the LA channel is also deleted. . Possible values = DISABLED, ACTIVE, PASSIVE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
lacptimeout	<String>	Read-only	Time to wait for the LACPDU. If a LACPDU is not received within this interval, the NetScaler markes the link partner port as DOWN. Possible values: Long and Short. Long lacptimeout is 90 sec and Short LACP timeout is 3 sec. Possible values = LONG, SHORT
lacpactorpriority	<Double>	Read-only	LACP Actor Priority. A LACP port priority is configured on each port using LACP. LACP uses the port priority with the port number to form the port identifier. The port priority determines which ports should be put in standby mode when there is a hardware limitation that prevents all compatible ports from aggregating.
lacpactorportno	<Double>	Read-only	LACP Actor port number. LACP uses the port priority with the port number to form the port identifier.
lacppartnerstate	<String>	Read-only	LACP Partner State. Whether the port is in Active or Passive negotiating state. Possible values = MANUAL, AUTO
lacppartnertimeout	<String>	Read-only	The timeout value for the information revieved in LACPDUs. It can have values as SHORT or LONG. The SHORT timeout is 3s and the LONG timeout is 90s. Possible values = LONG, SHORT
lacppartneraggregation	<String>	Read-only	The Aggregation flag indicates that the participant will allow the link to be used as part of an aggregate. Otherwise the link is to be used as an individual link, i.e. not aggregated with any other. Possible values = NS_EMPTY_STR, AGG
lacppartnerinsync	<String>	Read-only	The Synchronization flag indicates that the transmitting participant.s mux component is in sync with the system id and key information transmitted. Possible values = NS_EMPTY_STR, SYNC
lacppartnercollecting	<String>	Read-only	The Collecting flag indicates that the participant.s collector, i.e. the reception component of the mux, is definitely on. If set the flag communicates collecting. Possible values = NS_EMPTY_STR, COLLECTING
lacppartnerdistributing	<String>	Read-only	The Distributing flag indicates that the participant.s distributor is not definitely off. If reset the flag indicates not distributing. Possible values = NS_EMPTY_STR, DISTRIBUTING
lacppartnerdefaulted	<String>	Read-only	If the timer expires in the Expired state, the Receive Machine enters the Defaulted state. Possible values = NS_EMPTY_STR, DEFAULTED
lacppartnerexpired	<String>	Read-only	If the LACPDUs are received for timeout period, the Receive Machine enters the Expired state and the timer is restarted with the timeout value of SHORT timeout. Possible values = NS_EMPTY_STR, EXPIRED
lacppartnerpriority	<Double>	Read-only	LACP Partner Priority. A LACP port priority is configured on each port using LACP. LACP uses the port priority with the port number to form the port identifier. The port priority determines which ports should be put in standby mode when there is a hardware limitation that prevents all compatible ports from

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
lacppartnersystemmac	<String>	Read-only	LACP Partner System MAC.
lacppartnersystempriority	<Double>	Read-only	LACP Partner System Priority. The LACP partner's system priority. The values for the priority range from 0 to 65535. The lower the value, the higher the system priority. The switch with the lower system priority value determines which links between LACP partner are active and which are in the standby for each LACP Channel.
lacppartnerportno	<Double>	Read-only	LACP Partner Port number. LACP uses the port priority with the port number to form the port identifier.
lacppartnerkey	<Double>	Read-only	LACP Partner Key. The LACP key used by the partner port.
lacpactoraggregation	<String>	Read-only	The Aggregation flag indicates that the participant will allow the link to be used as part of an aggregate. Otherwise the link is to be used as an individual link, i.e. not aggregated with any other. Possible values = NS_EMPTY_STR, AGG
lacpactorinsync	<String>	Read-only	The Synchronization flag indicates that the transmitting participant.s mux component is in sync with the system id and key information transmitted. Possible values = NS_EMPTY_STR, SYNC
lacpactorcollecting	<String>	Read-only	The Collecting flag indicates that the participant.s collector, i.e. the reception component of the mux, is definitely on. If set the flag communicates collecting. Possible values = NS_EMPTY_STR, COLLECTING
lacpactordistributing	<String>	Read-only	The Distributing flag indicates that the participant.s distributor is not definitely off. If reset the flag indicates not distributing. Possible values = NS_EMPTY_STR, DISTRIBUTING
lacpportmuxstate	<String>	Read-only	LACP Port MUX state. The state of the MUX control machine. The Mux Control Machine attaches the physical port to an aggregate port, using the Selection Logic to choose an appropriate port, and turns the distributor and collector for the physical port on or off as required by protocol information. Possible values = DETACHED, WAITING, ATTACHED, COLLECTING, DISTRIBUTING
lacpportrxstat	<String>	Read-only	LACP Port RX state. The state of the Receive machine. The Receive Machine maintains partner information, recording protocol information from LACPDUs sent by remote partner(s). Received information is subject to a timeout, and if sufficient time elapses the receive machine will revert to using default partner information. Possible values = INIT, PORT_DISABLED, LACP_DISABLED, EXPIRED, DEFAULTED, CURRENT
lacpportselectstate	<String>	Read-only	LACP Port SELECT state. The state of the SELECT state machine, It could be SELECTED or UNSELECTED. Possible values = UNSELECTED, SELECTED, STANDBY
			Link Layer Discovery Protocol (LLDP) mode for an interface. The resultant

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			global and the interface levels. Possible values = NONE, TRANSMITTER, RECEIVER, TRANSCEIVER
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/channel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"channel":{
    "id":<String_value>,
    "ifnum":<String[]_value>,
    "state":<String_value>,
    "mode":<String_value>,
    "conndistr":<String_value>,
    "macdistr":<String_value>,
    "lamac":<String_value>,
    "speed":<String_value>,
    "flowctl":<String_value>,
    "hamonitor":<String_value>,
    "tagall":<String_value>,
    "trunk":<String_value>,
    "ifalias":<String_value>,
    "throughput":<Double_value>,
    "bandwidthhigh":<Double_value>,
    "bandwidthnormal":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/channel/id_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/channel

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"channel":{
    "id":<String_value>,
    "state":<String_value>,
    "mode":<String_value>,
    "conndistr":<String_value>,
    "macdistr":<String_value>,
    "lamac":<String_value>,
    "speed":<String_value>,
    "mtu":<Double_value>,
    "flowctl":<String_value>,
    "hamonitor":<String_value>,
    "tagall":<String_value>,
    "trunk":<String_value>,
    "ifalias":<String_value>,
    "throughput":<Double_value>,
    "lrminthroughput":<Double_value>,
    "linkredundancy":<String_value>,
    "bandwidthhigh":<Double_value>,
    "bandwidthnormal":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/channel?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"channel":{
    "id":<String_value>,
    "state":true,
    "mode":true,
    "conndistr":true,
    "macdistr":true,
    "speed":true,
    "mtu":true,
    "flowctl":true,
    "hamonitor":true,
    "tagall":true,
    "trunk":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "ifalias":true,
        "throughput":true,
        "lrminthroughput":true,
        "linkredundancy":true,
        "bandwidthhigh":true,
        "bandwidthnormal":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/channel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/channel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/channel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of channel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/channel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/channel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the channel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "channel": [ {
    "id":<String_value>,
    "devicename":<String_value>,
    "unit":<Double_value>,
    "description":<String_value>,
    "flags":<Double_value>,
    "mtu":<Double_value>,
    "actualmtu":<Double_value>,
    "vlan":<Double_value>,
    "mac":<String_value>,
    "lamac":<String_value>,
    "uptime":<Double_value>,
    "downtime":<Double_value>,
    "reqmedia":<String_value>,
    "reqspeed":<String_value>,
    "reqduplex":<String_value>,
    "reqflowcontrol":<String_value>,
    "media":<String_value>,
    "actspeed":<String_value>,
    "duplex":<String_value>,
    "actflowctl":<String_value>,
    "conndistr":<String_value>,
    "macdistr":<String_value>,
    "mode":<String_value>,
    "lamode":<String_value>,
    "hamonitor":<String_value>,
    "state":<String_value>,
    "autoneg":<Double_value>,
    "autonegresult":<Double_value>,
    "tagged":<Double_value>,
    "tagall":<String_value>,
    "trunk":<String_value>,
    "taggedany":<Double_value>,
    "taggedautolearn":<Double_value>,
    "hangdetect":<Double_value>,
    "hangreset":<Double_value>,
    "linkstate":<Double_value>,
    "intfstate":<Double_value>,
    "rxpackets":<Double_value>,
    "rxbytes":<Double_value>,
    "rxerrors":<Double_value>,
    "rxdrops":<Double_value>,
    "txpackets":<Double_value>,
    "txbytes":<Double_value>,
    "txerrors":<Double_value>,
    "txdrops":<Double_value>,
    "indisc":<Double_value>,
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "outdisc":<Double_value>,
    "fctls":<Double_value>,
    "hangs":<Double_value>,
    "stsstalls":<Double_value>,
    "txstalls":<Double_value>,
    "rxstalls":<Double_value>,
    "bdgmuted":<Double_value>,
    "vmac":<String_value>,
    "vmac6":<String_value>,
    "ifalias":<String_value>,
    "reqthroughput":<Double_value>,
    "lrminthroughput":<Double_value>,
    "linkredundancy":<String_value>,
    "actthroughput":<Double_value>,
    "bandwidthhigh":<Double_value>,
    "bandwidthnormal":<Double_value>,
    "ifnum":<String[]_value>,
    "backplane":<String_value>,
    "cleartime":<Double_value>,
    "lacpmode":<String_value>,
    "lacptimeout":<String_value>,
    "lacpactorpriority":<Double_value>,
    "lacpactorportno":<Double_value>,
    "lacppartnerstate":<String_value>,
    "lacppartnertimeout":<String_value>,
    "lacppartneraggregation":<String_value>,
    "lacppartnerinsync":<String_value>,
    "lacppartnercollecting":<String_value>,
    "lacppartnerdistributing":<String_value>,
    "lacppartnerdefaulted":<String_value>,
    "lacppartnerexpired":<String_value>,
    "lacppartnerpriority":<Double_value>,
    "lacppartnersystemmac":<String_value>,
    "lacppartnersystempriority":<Double_value>,
    "lacppartnerportno":<Double_value>,
    "lacppartnerkey":<Double_value>,
    "lacpactoraggregation":<String_value>,
    "lacpactorinsync":<String_value>,
    "lacpactorcollecting":<String_value>,
    "lacpactordistributing":<String_value>,
    "lacpportmuxstate":<String_value>,
    "lacpportrxstat":<String_value>,
    "lacpportselectstate":<String_value>,
    "lldpmode":<String_value>
}]}
```


get

URL: http://<netscaler-ip-address>/nitro/v1/config/channel/id_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/channel/id_value<String>?**attrs=property-name1,property-name2**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/channel/id_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "channel": [ {
    "id":<String_value>,
    "devicename":<String_value>,
    "unit":<Double_value>,
    "description":<String_value>,
    "flags":<Double_value>,
    "mtu":<Double_value>,
    "actualmtu":<Double_value>,
    "vlan":<Double_value>,
    "mac":<String_value>,
    "lamac":<String_value>,
    "uptime":<Double_value>,
    "downtime":<Double_value>,
    "reqmedia":<String_value>,
    "reqspeed":<String_value>,
    "reqduplex":<String_value>,
    "reqflowcontrol":<String_value>,
    "media":<String_value>,
    "actspeed":<String_value>,
    "duplex":<String_value>,
    "actflowctl":<String_value>,
    "conndistr":<String_value>,
    "macdistr":<String_value>,
    "mode":<String_value>,
    "lamode":<String_value>,
    "hamonitor":<String_value>,
    "state":<String_value>,
    "autoneg":<Double_value>,
    "autonegresult":<Double_value>,
    "tagged":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

"tagall":<String_value>,
"trunk":<String_value>,
"taggedany":<Double_value>,
"taggedautolearn":<Double_value>,
"hangdetect":<Double_value>,
"hangreset":<Double_value>,
"linkstate":<Double_value>,
"intfstate":<Double_value>,
"rxpackets":<Double_value>,
"rxbytes":<Double_value>,
"rxerrors":<Double_value>,
"rxdrops":<Double_value>,
"txpackets":<Double_value>,
"txbytes":<Double_value>,
"txerrors":<Double_value>,
"txdrops":<Double_value>,
"indisc":<Double_value>,
"outdisc":<Double_value>,
"fctls":<Double_value>,
"hangs":<Double_value>,
"stsstalls":<Double_value>,
"txstalls":<Double_value>,
"rxstalls":<Double_value>,
"bdgmuted":<Double_value>,
"vmac":<String_value>,
"vmac6":<String_value>,
"ifalias":<String_value>,
"reqthroughput":<Double_value>,
"lrminthroughput":<Double_value>,
"linkredundancy":<String_value>,
"actthroughput":<Double_value>,
"bandwidthhigh":<Double_value>,
"bandwidthnormal":<Double_value>,
"ifnum":<String[]_value>,
"backplane":<String_value>,
"cleartime":<Double_value>,
"lacpmode":<String_value>,
"lacptimeout":<String_value>,
"lacpactorpriority":<Double_value>,
"lacpactorportno":<Double_value>,
"lacppartnerstate":<String_value>,
"lacppartnertimeout":<String_value>,
"lacppartneraggregation":<String_value>,
"lacppartnerinsync":<String_value>,
"lacppartnercollecting":<String_value>,
"lacppartnerdistributing":<String_value>,
"lacppartnerdefaulted":<String_value>,
"lacppartnerexpired":<String_value>,
"lacppartnerpriority":<Double_value>,
"lacppartnersystemmac":<String_value>,
"lacppartnersystempriority":<Double_value>,
"lacppartnerportno":<Double_value>,
"lacppartnerkey":<Double_value>,
"lacpactoraggregation":<String_value>,
"lacpactorinsync":<String_value>,
"lacpactorcollecting":<String_value>,
"lacpactordistributing":<String_value>,
"lacpportmuxstate":<String_value>,
"lacpportrxstat":<String_value>,
"lacpportselectstate":<String_value>,
"lldpmode":<String_value>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/channel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "channel": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

channel_binding

May 20, 2016

Binding object which returns the resources bound to channel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<String>	Read-write	ID of an LA channel or LA channel in cluster configuration whose details you want the NetScaler appliance to display. Specify an LA channel in LA/x notation, where x can range from 1 to 8 or a cluster LA channel in CLA/x notation or Link redundant channel in LR/x notation , where x can range from 1 to 4. Minimum length = 1
channel_interface_binding	<channel_interface_binding[]>	Read-only	interface that can be bound to channel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/channel_binding/id_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "channel_binding": [ {
    "id":<String_value>,
    "channel_interface_binding":<channel_interface_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

channel_interface_binding

May 20, 2016

Binding object showing the interface that can be bound to channel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<String>	Read-write	ID of the LA channel or the cluster LA channel to which you want to bind interfaces. Specify an LA channel in LA/x notation, where x can range from 1 to 8 or a cluster LA channel in CLA/x notation or Link redundant channel in LR/x notation , where x can range from 1 to 4.
ifnum	<String[]>	Read-write	Interfaces to be bound to the LA channel of a NetScaler appliance or to the LA channel of a cluster configuration. For an LA channel of a NetScaler appliance, specify an interface in C/U notation (for example, 1/3). For an LA channel of a cluster configuration, specify an interface in N/C/U notation (for example, 2/1/3). where C can take one of the following values: * 0 - Indicates a management interface. * 1 - Indicates a 1 Gbps port. * 10 - Indicates a 10 Gbps port. U is a unique integer for representing an interface in a particular port group. N is the ID of the node to which an interface belongs in a cluster configuration. Use spaces to separate multiple entries.
lamode	<String>	Read-only	The mode(AUTO/MANNUAL) for the LA channel. Possible values = MANUAL, AUTO
slaveduplex	<Double>	Read-only	Duplex of the member interfaces.
slaveflowctl	<Double>	Read-only	Flowcontrol of the member interfaces.
slavetime	<Double>	Read-only	UP time of the member interfaces.
slavestate	<Double>	Read-only	State of the member interfaces.
slavespeed	<Double>	Read-only	Speed of the member interfaces.
slavemedia	<Double>	Read-only	Media type of the member interfaces.
lractiveintf	<Boolean>	Read-only	LR set member interface state(active/inactive).
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/channel_interface_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"channel_interface_binding":{
    "id":<String_value>,
    "ifnum":<String[]_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/channel_interface_binding/id_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/channel_interface_binding/id_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/channel_interface_binding/id_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of channel_interface_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/channel_interface_binding/id_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the channel_interface_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "channel_interface_binding": [ {
    "id":<String_value>,
    "ifnum":<String[]_value>,
    "lamode":<String_value>,
    "slaveduplex":<Double_value>,
    "slaveflowctl":<Double_value>,
    "slavetime":<Double_value>,
    "slavestate":<Double_value>,
    "slavespeed":<Double_value>,
    "slavemedia":<Double_value>,
    "lractiveintf":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/channel_interface_binding/id_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"channel_interface_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

fis

May 20, 2016

Configuration for "FIS" resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the FIS to be created. Leading character must be a number or letter. Other characters allowed, after the first character, are @ _ - . (period) : (colon) # and space ( ). Note: In a cluster setup, the FIS name on each node must be unique. Minimum length = 1
ownernode	<Double>	Read-write	ID of the cluster node for which you are creating the FIS. Can be configured only through the cluster IP address. Minimum value = 0 Maximum value = 31
ifaces	<String>	Read-only	Interfaces to be bound to the FIS, in slot/port notation (for example, 1/3).
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/fis

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"fis":{
    "name":<String_value>,
    "ownernode":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/fis/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/fis

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/fis?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/fis?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of fis resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/fis?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/fis?**pagesize=#no&pageno=#no**

Use this query-parameter to get the fis resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "fis": [ {
      "name":<String_value>,
      "ifaces":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ownernode":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/fis/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/fis/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/fis/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "fis": [ {
    "name":<String_value>,
    "ifaces":<String_value>,
    "ownernode":<Double_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/fis?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "fis": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

fis_binding

May 20, 2016

Binding object which returns the resources bound to fis.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the FIS configured on the appliance. . Minimum length = 1
fis_channel_binding	<fis_channel_binding[]>	Read-only	channel that can be bound to fis.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/fis_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "fis_binding": [ {
    "name":<String_value>,
    "fis_channel_binding":<fis_channel_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

fis_channel_binding

May 20, 2016

Binding object showing the channel that can be bound to fis.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ownernode	<Double>	Read-write	ID of the cluster node for which you are creating the FIS. Can be configured only through the cluster IP address. Minimum value = 0 Maximum value = 31
name	<String>	Read-write	The name of the FIS to which you want to bind interfaces. Minimum length = 1
ifnum	<String>	Read-write	Interface to be bound to the FIS, specified in slot/port notation (for example, 1/3).
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/fis_channel_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"fis_channel_binding":{
    "name":<String_value>,
    "ifnum":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/fis_channel_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/fis_channel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/fis_channel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of fis_channel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/fis_channel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the fis_channel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "fis_channel_binding": [ {
    "ownernode":<Double_value>,
    "name":<String_value>,
    "ifnum":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/fis_channel_binding/name_value<String>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"fis_channel_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

inatparam

May 20, 2016

Configuration for INAT parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
nat46v6prefix	<String>	Read-write	The prefix used for translating packets received from private IPv6 servers into IPv4 packets. This prefix has a length of 96 bits (128-32 = 96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix. IPv6 packets addressed to this prefix have to be routed to the NetScaler appliance to ensure that the IPv6-IPv4 translation is done by the appliance.
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
nat46ignoretos	<String>	Read-write	Ignore TOS. Default value: NO Possible values = YES, NO
nat46zerochecksum	<String>	Read-write	Calculate checksum for UDP packets with zero checksum. Default value: ENABLED Possible values = ENABLED, DISABLED
nat46v6mtu	<Double>	Read-write	MTU setting for the IPv6 side. If the incoming IPv4 packet greater than this, either fragment or send icmp need fragmentation error. Default value: 1280 Minimum value = 1280 Maximum value = 9216
nat46fragheader	<String>	Read-write	When disabled, translator will not insert IPv6 fragmentation header for non fragmented IPv4 packets. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/inatparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"inatparam":{
    "nat46v6prefix":<String_value>,
    "td":<Double_value>,
    "nat46ignoretos":<String_value>,
    "nat46zerochecksum":<String_value>,
    "nat46v6mtu":<Double_value>,
    "nat46fragheader":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/inatparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"inatparam":{
    "nat46v6prefix":true,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/inatparam

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/inatparam?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/inatparam?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of inatparam resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/inatparam?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/inatparam?**pagesize=#no&pageno=#no**

Use this query-parameter to get the inatparam resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "inatparam": [ {
      "td":<Double_value>,
      "nat46v6prefix":<String_value>,
      "nat46ignoretos":<String_value>,
      "nat46zerochecksum":<String_value>,
      "nat46v6mtu":<Double_value>,
      "nat46fragheader":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/inatparam/td_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/inatparam/td_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/inatparam/td_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "inatparam": [ {
    "td":<Double_value>,
    "nat46v6prefix":<String_value>,
    "nat46ignoretos":<String_value>,
    "nat46zerochecksum":<String_value>,
    "nat46v6mtu":<Double_value>,
    "nat46fragheader":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/inatparam?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "inatparam": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

fis_interface_binding

May 20, 2016

Binding object showing the interface that can be bound to fis.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ownernode	<Double>	Read-write	ID of the cluster node for which you are creating the FIS. Can be configured only through the cluster IP address. Minimum value = 0 Maximum value = 31
name	<String>	Read-write	The name of the FIS to which you want to bind interfaces. Minimum length = 1
ifnum	<String>	Read-write	Interface to be bound to the FIS, specified in slot/port notation (for example, 1/3).

Operations (click to see Properties)

ADD: | DELETE:

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/fis_interface_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"fis_interface_binding":{
    "name":<String_value>,
    "ifnum":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/fis_interface_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

forwardingsession

May 20, 2016

Configuration for session forward resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the forwarding session rule. Can begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the rule is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rule" or 'my rule'). Minimum length = 1
network	<String>	Read-write	An IPv4 network address or IPv6 prefix of a network from which the forwarded traffic originates or to which it is destined. Minimum length = 1
netmask	<String>	Read-write	Subnet mask associated with the network. Minimum length = 1
acl6name	<String>	Read-write	Name of any configured ACL6 whose action is ALLOW. The rule of the ACL6 is used as a forwarding session rule. Minimum length = 1
aclname	<String>	Read-write	Name of any configured ACL whose action is ALLOW. The rule of the ACL is used as a forwarding session rule. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
connfailover	<String>	Read-write	Synchronize connection information with the secondary appliance in a high availability (HA) pair. That is, synchronize all connection-related information for the forwarding session. Default value: DISABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD | UPDATE | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/forwardingsession

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"forwardingsession":{
    "name":<String_value>,
    "network":<String_value>,
    "netmask":<String_value>,
    "acl6name":<String_value>,
    "aclname":<String_value>,
    "td":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "connfailover":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/forwardingsession

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"forwardingsession":{
    "name":<String_value>,
    "connfailover":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/forwardingsession/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/forwardingsession

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/forwardingsession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/forwardingsession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of forwardingsession resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/forwardingsession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/forwardingsession?**pagesize=#no&pageno=#no**

Use this query-parameter to get the forwardingsession resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "forwardingsession": [ {
    "name":<String_value>,
    "network":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "netmask":<String_value>,
    "td":<Double_value>,
    "aclname":<String_value>,
    "acl6name":<String_value>,
    "connfailover":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/forwardingsession/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/forwardingsession/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/forwardingsession/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "forwardingsession": [ {
    "name":<String_value>,
    "network":<String_value>,
    "netmask":<String_value>,
    "td":<Double_value>,
    "aclname":<String_value>,
    "acl6name":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "connfailover":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/forwardingsession?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "forwardingsession": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

inat

Jul 07, 2017

Configuration for inbound nat resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the Inbound NAT (INAT) entry. Leading character must be a number or letter. Other characters allowed, after the first character, are @ _ - . (period) : (colon) # and space ( ). Minimum length = 1
publicip	<String>	Read-write	Public IP address of packets received on the NetScaler appliance. Can be aNetScaler-owned VIP or VIP6 address. Minimum length = 1
privateip	<String>	Read-write	IP address of the server to which the packet is sent by the NetScaler. Can be an IPv4 or IPv6 address. Minimum length = 1
mode	<String>	Read-write	Stateless translation. Possible values = STATELESS
tcpproxy	<String>	Read-write	Enable TCP proxy, which enables the NetScaler appliance to optimize the RNAT TCP traffic by using Layer 4 features. Default value: DISABLED Possible values = ENABLED, DISABLED
ftp	<String>	Read-write	Enable the FTP protocol on the server for transferring files between the client and the server. Default value: DISABLED Possible values = ENABLED, DISABLED
tftp	<String>	Read-write	To enable/disable TFTP (Default DISABLED). Default value: DISABLED Possible values = ENABLED, DISABLED
usip	<String>	Read-write	Enable the NetScaler appliance to retain the source IP address of packets before sending the packets to the server. Possible values = ON, OFF
usnip	<String>	Read-write	Enable the NetScaler appliance to use a SNIP address as the source IP address of packets before sending the packets to the server. Possible values = ON, OFF
proxyip	<String>	Read-write	Unique IP address used as the source IP address in packets sent to the server. Must be a

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
useproxyport	<String>	Read-write	Enable the NetScaler appliance to proxy the source port of packets before sending the packets to the server. Default value: ENABLED Possible values = ENABLED, DISABLED
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
flags	<Double>	Read-only	Flags for different modes.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/inat

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"inat":{
    "name":<String_value>,
    "publicip":<String_value>,
    "privateip":<String_value>,
    "mode":<String_value>,
    "tcpproxy":<String_value>,
    "ftp":<String_value>,
    "tftp":<String_value>,
    "usip":<String_value>,
    "usnip":<String_value>,
    "proxyip":<String_value>,
    "useproxyport":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/inat/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/inat

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"inat":{
    "name":<String_value>,
    "privateip":<String_value>,
    "tcpproxy":<String_value>,
    "ftp":<String_value>,
    "tftp":<String_value>,
    "usip":<String_value>,
    "usnip":<String_value>,
    "proxyip":<String_value>,
    "useproxyport":<String_value>,
    "mode":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/inat?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"inat":{
    "name":<String_value>,
    "tcpproxy":true,
    "ftp":true,
    "tftp":true,
    "usip":true,
    "usnip":true,
    "proxyip":true,
    "useproxyport":true,
    "mode":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/inat

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/inat?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/inat?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of inat resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/inat?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/inat?**pagesize=#no&pageno=#no**

Use this query-parameter to get the inat resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "inat": [ {
      "name":<String_value>,
      "publicip":<String_value>,
      "privateip":<String_value>,
      "proxyip":<String_value>,
      "tcpproxy":<String_value>,
      "ftp":<String_value>,
      "tftp":<String_value>,
      "usip":<String_value>,
      "usnip":<String_value>,
      "useproxyport":<String_value>,
      "flags":<Double_value>,
      "mode":<String_value>,
      "td":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/inat/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/inat/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/inat/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "inat": [ {
    "name":<String_value>,
    "publicip":<String_value>,
    "privateip":<String_value>,
    "proxyip":<String_value>,
    "tcpproxy":<String_value>,
    "ftp":<String_value>,
    "tftp":<String_value>,
    "usip":<String_value>,
    "usnip":<String_value>,
    "useproxyport":<String_value>,
    "flags":<Double_value>,
    "mode":<String_value>,
    "td":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/inat?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "inat": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Interface

May 20, 2016

Configuration for interface resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<String>	Read-write	Interface number, in C/U format, where C can take one of the following values: * 0 - Indicates a management interface. * 1 - Indicates a 1 Gbps port. * 10 - Indicates a 10 Gbps port. * LA - Indicates a link aggregation port. * LO - Indicates a loop back port. U is a unique integer for representing an interface in a particular port group.
speed	<String>	Read-write	Ethernet speed of the interface, in Mbps. Notes: * If you set the speed as AUTO, the NetScaler appliance attempts to auto-negotiate or auto-sense the link speed of the interface when it is UP. You must enable auto negotiation on the interface. * If you set a speed other than AUTO, you must specify the same speed for the peer network device. Mismatched speed and duplex settings between the peer devices of a link lead to link errors, packet loss, and other errors. Some interfaces do not support certain speeds. If you specify an unsupported speed, an error message appears. Default value: AUTO Possible values = AUTO, 10, 100, 1000, 10000, 40000
duplex	<String>	Read-write	The duplex mode for the interface. Notes:* If you set the duplex mode to AUTO, the NetScaler appliance attempts to auto-negotiate the duplex mode of the interface when it is UP. You must enable auto negotiation on the interface. If you set a duplex mode other than AUTO, you must specify the same duplex mode for the peer network device. Mismatched speed and duplex settings between the peer devices of a link lead to link errors, packet loss, and other errors. Default value: AUTO Possible values = AUTO, HALF, FULL
flowctl	<String>	Read-write	802.3x flow control setting for the interface. The 802.3x specification does not define flow control for 10 Mbps and 100 Mbps speeds, but if a Gigabit Ethernet interface operates at those speeds, the flow control settings can be applied. The flow control setting that is finally applied to an interface depends on auto-negotiation. With the ON option, the peer negotiates the flow control, but the appliance then forces two-way flow control for the interface.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = OFF, RX, TX, RXTX
autoneg	<String>	Read-write	Auto-negotiation state of the interface. With the ENABLED setting, the NetScaler appliance auto-negotiates the speed and duplex settings with the peer network device on the link. The NetScaler appliance auto-negotiates the settings of only those parameters (speed or duplex mode) for which the value is set as AUTO. Default value: NSA_DVC_AUTONEG_ON Possible values = DISABLED, ENABLED
hamonitor	<String>	Read-write	In a High Availability (HA) configuration, monitor the interface for failure events. In an HA configuration, an interface that has HA MON enabled and is not bound to any Failover Interface Set (FIS), is a critical interface. Failure or disabling of any critical interface triggers HA failover. Default value: ON Possible values = ON, OFF
mtu	<Double>	Read-write	The maximum transmission unit (MTU) is the largest packet size, measured in bytes excluding 14 bytes ethernet header and 4 bytes crc, that can be transmitted and received by this interface. Default value of MTU is 1500 on all the interface of Netscaler appliance any value configured more than 1500 on the interface will make the interface as jumbo enabled. In case of cluster backplane interface MTU value will be changed to 1514 by default, user has to change the backplane interface value to maximum mtu configured on any of the interface in cluster system plus 14 bytes more for backplane interface if Jumbo is enabled on any of the interface in a cluster system. Changing the backplane will bring back the MTU of backplane interface to default value of 1500. If a channel is configured as backplane then the same holds true for channel as well as member interfaces. In case of channel if member interfaces is configured as different mtu then the highest MTU configured MTU is treated as the LA MTU if MTU is not specified on LA explicitly. Low MTU interfaces in channel will be taken out of LA distribution list. Default value: 1500 Minimum value = 1500 Maximum value = 9216
tagall	<String>	Read-write	Add a four-byte 802.1q tag to every packet sent on this interface. The ON setting applies the tag for this interface's native VLAN. OFF applies the tag for all VLANs other than the native VLAN. Default value: OFF Possible values = ON, OFF
trunk	<String>	Read-write	This argument is deprecated by tagall. Default value: OFF Possible values = ON, OFF
lacpmode	<String>	Read-write	Bind the interface to a LA channel created by the Link Aggregation control protocol (LACP). Available settings function as follows: * Active - The LA channel port of the NetScaler appliance generates LACPDU messages on a regular basis, regardless of any need expressed by its peer device to receive them. * Passive - The LA channel port of the NetScaler appliance does not transmit

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			the port does not speak unless spoken to. * Disabled - Unbinds the interface from the LA channel. If this is the only interface in the LA channel, the LA channel is removed. Default value: DISABLED Possible values = DISABLED, ACTIVE, PASSIVE
lacpkey	<Double>	Read-write	Integer identifying the LACP LA channel to which the interface is to be bound. For an LA channel of the NetScaler appliance, this digit specifies the variable x of an LA channel in LA/x notation, where x can range from 1 to 8. For example, if you specify 3 as the LACP key for an LA channel, the interface is bound to the LA channel LA/3. For an LA channel of a cluster configuration, this digit specifies the variable y of a cluster LA channel in CLA/(y-4) notation, where y can range from 5 to 8. For example, if you specify 6 as the LACP key for a cluster LA channel, the interface is bound to the cluster LA channel CLA/2. Minimum value = 1 Maximum value = 8
lagtype	<String>	Read-write	Type of entity (NetScaler appliance or cluster configuration) for which to create the channel. Default value: NODE Possible values = NODE, CLUSTER
lacppriority	<Double>	Read-write	LACP port priority, expressed as an integer. The lower the number, the higher the priority. The NetScaler appliance limits the number of interfaces in an LA channel to sixteen. Default value: 32768 Minimum value = 1 Maximum value = 65535
lacptimeout	<String>	Read-write	Interval at which the NetScaler appliance sends LACPDU messages to the peer device on the LA channel. Available settings function as follows: LONG - 30 seconds. SHORT - 1 second. Default value: NSA_LACP_TIMEOUT_LONG Possible values = LONG, SHORT
ifalias	<String>	Read-write	Alias name for the interface. Used only to enhance readability. To perform any operations, you have to specify the interface ID. Default value: " " Maximum length = 31
throughput	<Double>	Read-write	Low threshold value for the throughput of the interface, in Mbps. In an HA configuration, failover is triggered if the interface has HA MON enabled and the throughput is below the specified the threshold. Minimum value = 0 Maximum value = 160000
linkredundancy	<String>	Read-write	Link Redundancy for Cluster LAG. Default value: OFF Possible values = ON, OFF
			High threshold value for the bandwidth usage of the interface, in Mbps. The

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
bandwidthhigh	<Double>	Read-write	usage of the interface is greater than or equal to the specified high threshold value. Minimum value = 0 Maximum value = 160000
bandwidthnormal	<Double>	Read-write	Normal threshold value for the bandwidth usage of the interface, in Mbps. When the bandwidth usage of the interface becomes less than or equal to the specified normal threshold after exceeding the high threshold, the NetScaler appliance generates an SNMP trap message to indicate that the bandwidth usage has returned to normal. Minimum value = 0 Maximum value = 160000
lldpmode	<String>	Read-write	Link Layer Discovery Protocol (LLDP) mode for an interface. The resultant LLDP mode of an interface depends on the LLDP mode configured at the global and the interface levels. Possible values = NONE, TRANSMITTER, RECEIVER, TRANSCEIVER
lrsetpriority	<Double>	Read-write	LRSET port priority, expressed as an integer ranging from 1 to 1024. The highest priority is 1. The NetScaler limits the number of interfaces in an LRSET to 8. Within a LRSET the highest LR Priority Interface is considered as the first candidate for the Active interface, if the interface is UP. Default value: 1024 Minimum value = 1 Maximum value = 1024
devicename	<String>	Read-only	Name of the interface.
unit	<Double>	Read-only	Unit number for this interface, signifying the sequence number in which this interface is discovered on this Netscaler.
description	<String>	Read-only	Display the type of interface, the speeds at which this interface can operate, and, if applicable, the type of SFP,.
flags	<Double>	Read-only	Flags for this interface. Used for communicating the device states.
actualmtu	<Double>	Read-only	MTU for this interface (the largest frame that can transit this interface).
vlan	<Double>	Read-only	Native VLAN for this interface.
mac	<String>	Read-only	MAC address for this interface.
uptime	<Double>	Read-only	Duration for which the interface has been UP (Example: 3 hours 1 minute 1 second). This value is reset when the interface state changes to DOWN..
downtime	<Double>	Read-only	Duration for which the interface has been DOWN. This value is reset when the interface state changes to UP.(Example: 3 hours 1 minute 1 second).
reqmedia	<String>	Read-only	Requested media setting for this interface.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
reqspeed	<String>	Read-only	Requested speed setting for this interface. Possible values = AUTO, 10, 100, 1000, 10000, 40000
reqduplex	<String>	Read-only	Requested duplex setting for this interface. Possible values = AUTO, HALF, FULL
reqflowcontrol	<String>	Read-only	Requested flow control setting for this interface. Possible values = OFF, RX, TX, RXTX
actmedia	<String>	Read-only	Actual media setting for this interface. Possible values = AUTO, UTP, FIBER
actspeed	<String>	Read-only	Actual speed setting for this interface. Possible values = AUTO, 10, 100, 1000, 10000, 40000
actduplex	<String>	Read-only	Actual duplex setting for this interface. Possible values = AUTO, HALF, FULL
actflowctl	<String>	Read-only	Actual flow control setting for this interface. Possible values = OFF, RX, TX, RXTX
conndistr	<String>	Read-only	Connection distribution setting on this interface. Possible values = DISABLED, ENABLED
macdistr	<String>	Read-only	MAC distribution setting on this interface. Possible values = SOURCE, DESTINATION, BOTH
mode	<String>	Read-only	Interface Link Aggregation mode (Auto/Manual) setting. Possible values = MANUAL, AUTO
state	<String>	Read-only	Link state of the interface (UP/DOWN). Default value: ENABLED Possible values = ENABLED, DISABLED
autonegresult	<Double>	Read-only	Actual auto-negotiation setting for this interface.
tagged	<Double>	Read-only	VLAN tags setting on this channel.
taggedany	<Double>	Read-only	Interface setting to accept/drop all tagged packets.
taggedautolearn	<Double>	Read-only	Dynamic VLAN membership autolearning enabled or disabled on this interface.
hangdetect	<Double>	Read-only	Hang detection enabled or disabled for this interface.
hangreset	<Double>	Read-only	Hang reset enabled or disabled for this interface.
			The current state of the link associated with the interface. For logical interfaces (LA), the state of the link is dependent on the state of the slave

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			be UP.
intfstate	<Double>	Read-only	Current state of the specified interface. The interface state set to UP only if the link state is UP and administrative state is ENABLED.
rxpackets	<Double>	Read-only	Number of packets received by an interface since the NetScaler appliance was started or the interface statistics were cleared.
rxbytes	<Double>	Read-only	Number of bytes received by an interface since the NetScaler appliance was started or the interface statistics were cleared.
rxerrors	<Double>	Read-only	Number of inbound packets dropped by the hardware on a specified interface since the NetScaler appliance was started or the interface statistics were cleared. Packets can be dropped because of CRC, length (undersize or oversize), or alignment errors.
rxdrops	<Double>	Read-only	Number of inbound packets dropped by the specified interface. Commonly dropped packets are multicast frames, spanning tree BPDUs, packets destined to a MAC not owned by the NetScaler appliance when L2 mode is disabled, or packets tagged for a VLAN that is not bound to the interface. In most healthy networks, this statistic increments at a steady rate regardless of traffic load. A sharp spike in dropped packets generally indicates an issue with connected L2 switches, such as a forwarding database overflow resulting in packets being broadcast on all ports.
txpackets	<Double>	Read-only	Number of packets transmitted by an interface since the NetScaler appliance was started or the interface statistics were cleared.
txbytes	<Double>	Read-only	Number of bytes transmitted by an interface since the NetScaler appliance was started or the interface statistics were cleared.
txerrors	<Double>	Read-only	Number of outbound packets dropped by the hardware on a specified interface since the NetScaler appliance was started or the interface statistics were cleared. Packets can be dropped because of length (undersize or oversize) errors or a lack of resources. This statistic is available only for: (1) Loop back interface (LO) of all platforms. (2) All data ports on the NetScaler 12000 platform. (3) Management ports on the Netscaler MPX 15000 and 17000 platforms. .
txdrops	<Double>	Read-only	Number of packets dropped in transmission by the specified interface for one of the following reasons. (1) VLAN mismatch. (2) Oversized packets. (3) Interface congestion. (4) Loopback packets sent on non loop back interface. .
indisc	<Double>	Read-only	Number of error-free inbound packets discarded by the specified interface because of a lack of resources (for example, insufficient receive buffers).
			Number of error-free outbound packets discarded by the specified interface because of a lack of resources. This statistic is not available on:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			(2) 10G data ports on NetScaler MPX 17500/19500/21500 platforms.
fctls	<Double>	Read-only	Number of times flow control is performed on the specified interface because of received pause frames.
hangs	<Double>	Read-only	Number of times the specified interface detected hangs in the transmit and receive paths since the NetScaler appliance was started or the interface statistics were cleared.
stsstalls	<Double>	Read-only	Number of times the status updates for a specified interface were stalled since the NetScaler appliance was started or the interface statistics were cleared. A status stall is detected when the status of the interface is not updated by the NIC hardware within 0.8 seconds of the last update.
txstalls	<Double>	Read-only	Number of times the interface stalled, when transmitting packets, since the NetScaler appliance was started or the interface statistics were cleared. Transmit (Tx) stalls are detected when a packet posted for transmission is not transmitted in 4 seconds.
rxstalls	<Double>	Read-only	Number of times the interface stalled, when receiving packets, since the NetScaler appliance was started or the interface statistics were cleared. Receive (Rx) stalls are detected when the following conditions are met: (1)The link is up for more than 10 minutes. (2)Packets are transmitted, but no packets are received for 16 seconds. .
bdgmacmoved	<Double>	Read-only	Number of MAC moves between ports. A high rate of MAC moves typically indicates a bridge loop between two interfaces.
bdgmuted	<Double>	Read-only	Number of times the specified interface stopped transmitting and receiving packets because of MAC moves between ports.
vmac	<String>	Read-only	Virtual MAC of this interface.
vmac6	<String>	Read-only	Virtual MAC for IPv6 of this interface.
reqthroughput	<Double>	Read-only	Minimum required throughput for an interface. Failover is triggered if the operating throughput of a Link Aggregation (LA) channel for which HAMON is ON falls below this value. The possible values are: 1. 1000Mbps for 1G interfaces. 2. 10000Mbps for 10G interfaces. 3. 160000Mbps for Link Aggregation channels. Minimum value = 0 Maximum value = 160000
actthroughput	<Double>	Read-only	Actual throughput for the interface.
backplane	<String>	Read-only	The cluster backplane status of the interface. If the status is enabled, the interface is part of the cluster backplane. By default, the backplane status is

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = ENABLED, DISABLED
mtcm1	<String[]>	Read-only	Contains the LA Master, if the interface is part of LA channel.
cleartime	<Double>	Read-only	Time since the interface stats are cleared last time.
slavestate	<Double>	Read-only	State of the member interfaces.
slavemedia	<Double>	Read-only	Media type of the member interfaces.
slavespeed	<Double>	Read-only	Speed of the member interfaces.
slaveduplex	<Double>	Read-only	Duplex of the member interfaces.
slaveflowctl	<Double>	Read-only	Flowcontrol of the member interfaces.
slavetime	<Double>	Read-only	UP time of the member interfaces.
intftype	<String>	Read-only	Interface Type, this field will have the interface type either it is virtual, physical or loopback. . Possible values = BROADCOM 5700/5701, TIGON1/TIGON2, INTEL 82546, INTEL 8255X(PRO), Link Aggregate, Loopback, Intel 82541/47, Broadcom 5704, Chelsio 1G, Intel 8247X, Intel 82576 VF, Xen Virtual, Intel 10G, Intel 82599 VF, Hyper v, Cluster LAG, Intel 8247X SFP, XEN Interface, Chelsio 10G, KVM Virtio
lacpactormode	<String>	Read-only	* Active - The LA channel port of the NetScaler appliance generates LACPDU messages on a regular basis, regardless of any need expressed by its peer device to receive them. * Passive - The LA channel port of the NetScaler appliance does not transmit LACPDU messages unless the peer device port is in the active mode. That is, the port does not speak unless spoken to. * Disabled - Unbinds the interface from the LA channel. If this is the only interface in the LA channel, the LA channel is removed. Possible values = DISABLED, ACTIVE, PASSIVE
lacpactortimeout	<String>	Read-only	Interval at which the NetScaler appliance sends LACPDU messages to the peer device on the LA channel. Available settings function as follows: LONG - 30 seconds. SHORT - 1 second. Possible values = LONG, SHORT
lacpactorpriority	<Double>	Read-only	LACP Actor Priority. A LACP port priority is configured on each port using LACP. LACP uses the port priority with the port number to form the port identifier. The port priority determines which ports should be put in standby

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			from aggregating.
lacpactorportno	<Double>	Read-only	LACP Actor port number. LACP uses the port priority with the port number to form the port identifier.
lacppartnerstate	<String>	Read-only	LACP Partner State. Whether the port is in Active or Passive negotiating state. Possible values = MANUAL, AUTO
lacppartnertimeout	<String>	Read-only	The timeout value for the information revieved in LACPDUs. It can have values as SHORT or LONG. The SHORT timeout is 3s and the LONG timeout is 90s. Possible values = LONG, SHORT
lacppartneraggregation	<String>	Read-only	The Aggregation flag indicates that the participant will allow the link to be used as part of an aggregate. Otherwise the link is to be used as an individual link, i.e. not aggregated with any other. Possible values = NS_EMPTY_STR, AGG
lacppartnerinsync	<String>	Read-only	The Synchronization flag indicates that the transmitting participant.s mux component is in sync with the system id and key information transmitted. Possible values = NS_EMPTY_STR, SYNC
lacppartnercollecting	<String>	Read-only	The Collecting flag indicates that the participant.s collector, i.e. the reception component of the mux, is definitely on. If set the flag communicates collecting. Possible values = NS_EMPTY_STR, COLLECTING
lacppartnerdistributing	<String>	Read-only	The Distributing flag indicates that the participant.s distributor is not definitely off. If reset the flag indicates not distributing. Possible values = NS_EMPTY_STR, DISTRIBUTING
lacppartnerdefaulted	<String>	Read-only	If the timer expires in the Expired state, the Receive Machine enters the Defaulted state. Possible values = NS_EMPTY_STR, DEFAULTED
lacppartnerexpired	<String>	Read-only	If the LACPDUs are received for timeout period, the Receive Machine enters the Expired state and the timer is restarted with the timeout value of SHORT timeout. Possible values = NS_EMPTY_STR, EXPIRED
lacppartnerpriority	<Double>	Read-only	LACP Partner Priority. A LACP port priority is configured on each port using LACP. LACP uses the port priority with the port number to form the port identifier. The port priority determines which ports should be put in standby mode when there is a hardware limitation that prevents all compatible ports from aggregating.
lacppartnersystemmac	<String>	Read-only	LACP Partner System MAC.
lacppartnersystempriority	<Double>	Read-only	LACP Partner System Priority. The LACP partner's system priority. The values for the priority range from 0 to 65535. The lower the value, the higher the system priority. The switch with the lower system priority value determines

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			for each LACP Channel.
lacppartnerportno	<Double>	Read-only	LACP Partner Port number. LACP uses the port priority with the port number to form the port identifier.
lacppartnerkey	<Double>	Read-only	LACP Partner Key. The LACP key used by the partner port.
lacpactoraggregation	<String>	Read-only	The Aggregation flag indicates that the participant will allow the link to be used as part of an aggregate. Otherwise the link is to be used as an individual link, i.e. not aggregated with any other. Possible values = NS_EMPTY_STR, AGG
lacpactorinsync	<String>	Read-only	The Synchronization flag indicates that the transmitting participant.s mux component is in sync with the system id and key information transmitted. Possible values = NS_EMPTY_STR, SYNC
lacpactorcollecting	<String>	Read-only	The Collecting flag indicates that the participant.s collector, i.e. the reception component of the mux, is definitely on. If set the flag communicates collecting. Possible values = NS_EMPTY_STR, COLLECTING
lacpactordistributing	<String>	Read-only	The Distributing flag indicates that the participant.s distributor is not definitely off. If reset the flag indicates not distributing. Possible values = NS_EMPTY_STR, DISTRIBUTING
lacpportmuxstate	<String>	Read-only	LACP Port MUX state. The state of the MUX control machine. The Mux Control Machine attaches the physical port to an aggregate port, using the Selection Logic to choose an appropriate port, and turns the distributor and collector for the physical port on or off as required by protocol information. Possible values = DETACHED, WAITING, ATTACHED, COLLECTING, DISTRIBUTING
lacpportrxstat	<String>	Read-only	LACP Port RX state. The state of the Receive machine. The Receive Machine maintains partner information, recording protocol information from LACPDUs sent by remote partner(s). Received information is subject to a timeout, and if sufficient time elapses the receive machine will revert to using default partner information. Possible values = INIT, PORT_DISABLED, LACP_DISABLED, EXPIRED, DEFAULTED, CURRENT
lacpportselectstate	<String>	Read-only	LACP Port SELECT state. The state of the SELECT state machine, It could be SELECTED or UNSELECTED. Possible values = UNSELECTED, SELECTED, STANDBY
lractiveintf	<Boolean>	Read-only	LR set member interface state(active/inactive).
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

CLEAR | UPDATE | UNSET | ENABLE | DISABLE | RESET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/Interface?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"Interface":{
    "id":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/Interface

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"Interface":{
    "id":<String_value>,
    "speed":<String_value>,
    "duplex":<String_value>,
    "flowctl":<String_value>,
    "autoneg":<String_value>,
    "hamonitor":<String_value>,
    "mtu":<Double_value>,
    "tagall":<String_value>,
    "trunk":<String_value>,
    "lacpmode":<String_value>,
    "lacpkey":<Double_value>,
    "lagtype":<String_value>,
    "lacppriority":<Double_value>,
    "lacptimeout":<String_value>,
    "ifalias":<String_value>,
    "throughput":<Double_value>,
    "linkredundancy":<String_value>,
    "bandwidthhigh":<Double_value>,
    "bandwidthnormal":<Double_value>,
    "lldpmode":<String_value>,
    "lrsetpriority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/Interface?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"Interface":{
    "id":<String_value>,
    "speed":true,
    "duplex":true,
    "flowctl":true,
    "autoneg":true,
    "hamonitor":true,
    "mtu":true,
    "tagall":true,
    "trunk":true,
    "lacpmode":true,
    "lacpkey":true,
    "lacppriority":true,
    "lacptimeout":true,
    "ifalias":true,
    "throughput":true,
    "linkredundancy":true,
    "bandwidthhigh":true,
    "bandwidthnormal":true,
    "lldpmode":true,
    "lrsetpriority":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/Interface?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"Interface":{
    "id":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/Interface?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"Interface":{
    "id":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

reset

URL: http://<netscaler-ip-address>/nitro/v1/config/Interface?**action=reset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"Interface":{
    "id":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/Interface

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/Interface?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/Interface?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of Interface resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/Interface?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/Interface?**pagesize=#no&pageno=#no**

Use this query-parameter to get the Interface resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "Interface": [ {
    "id":<String_value>,
    "devicename":<String_value>,
    "unit":<Double_value>,
    "description":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

"flags":<Double_value>,
"mtu":<Double_value>,
"actualmtu":<Double_value>,
"vlan":<Double_value>,
"mac":<String_value>,
"uptime":<Double_value>,
"downtime":<Double_value>,
"reqmedia":<String_value>,
"reqspeed":<String_value>,
"reqduplex":<String_value>,
"reqflowcontrol":<String_value>,
"actmedia":<String_value>,
"actspeed":<String_value>,
"actduplex":<String_value>,
"actflowctl":<String_value>,
"conndistr":<String_value>,
"macdistr":<String_value>,
"mode":<String_value>,
"hamonitor":<String_value>,
"state":<String_value>,
"autoneg":<String_value>,
"autonegresult":<Double_value>,
"tagged":<Double_value>,
"tagall":<String_value>,
"trunk":<String_value>,
"taggedany":<Double_value>,
"taggedautolearn":<Double_value>,
"hangdetect":<Double_value>,
"hangreset":<Double_value>,
"linkstate":<Double_value>,
"intfstate":<Double_value>,
"rxpackets":<Double_value>,
"rxbytes":<Double_value>,
"rxerrors":<Double_value>,
"rxdrops":<Double_value>,
"txpackets":<Double_value>,
"txbytes":<Double_value>,
"txerrors":<Double_value>,
"txdrops":<Double_value>,
"indisc":<Double_value>,
"outdisc":<Double_value>,
"fctls":<Double_value>,
"hangs":<Double_value>,
"stsstalls":<Double_value>,
"txstalls":<Double_value>,
"rxstalls":<Double_value>,
"bdgmacmoved":<Double_value>,
"bdgmuted":<Double_value>,
"vmac":<String_value>,
"vmac6":<String_value>,
"lacpmode":<String_value>,
"lacpkey":<Double_value>,
"lacppriority":<Double_value>,
"lacptimeout":<String_value>,
"lagtype":<String_value>,
"ifalias":<String_value>,
"reqthroughput":<Double_value>,
"linkredundancy":<String_value>,
"actthroughput":<Double_value>,
"bandwidthhigh":<Double_value>,
"bandwidthnormal":<Double_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "backplane":<String_value>,
    "ifnum":<String[]_value>,
    "cleartime":<Double_value>,
    "slavestate":<Double_value>,
    "slavemedia":<Double_value>,
    "slavespeed":<Double_value>,
    "slaveduplex":<Double_value>,
    "slaveflowctl":<Double_value>,
    "slavetime":<Double_value>,
    "intftype":<String_value>,
    "lacpactormode":<String_value>,
    "lacpactortimeout":<String_value>,
    "lacpactorpriority":<Double_value>,
    "lacpactorportno":<Double_value>,
    "lacppartnerstate":<String_value>,
    "lacppartnertimeout":<String_value>,
    "lacppartneraggregation":<String_value>,
    "lacppartnerinsync":<String_value>,
    "lacppartnercollecting":<String_value>,
    "lacppartnerdistributing":<String_value>,
    "lacppartnerdefaulted":<String_value>,
    "lacppartnerexpired":<String_value>,
    "lacppartnerpriority":<Double_value>,
    "lacppartnersystemmac":<String_value>,
    "lacppartnersystempriority":<Double_value>,
    "lacppartnerportno":<Double_value>,
    "lacppartnerkey":<Double_value>,
    "lacpactoraggregation":<String_value>,
    "lacpactorinsync":<String_value>,
    "lacpactorcollecting":<String_value>,
    "lacpactordistributing":<String_value>,
    "lacpportmuxstate":<String_value>,
    "lacpportrxstat":<String_value>,
    "lacpportselectstate":<String_value>,
    "lldpmode":<String_value>,
    "lrsetpriority":<Double_value>,
    "lractiveintf":<Boolean_value>
}]}
```


get

URL: http://<netscaler-ip-address>/nitro/v1/config/Interface/id_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/Interface/id_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/Interface/id_value<String>?**view=summary**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "Interface": [ {
    "id":<String_value>,
    "devicename":<String_value>,
    "unit":<Double_value>,
    "description":<String_value>,
    "flags":<Double_value>,
    "mtu":<Double_value>,
    "actualmtu":<Double_value>,
    "vlan":<Double_value>,
    "mac":<String_value>,
    "uptime":<Double_value>,
    "downtime":<Double_value>,
    "reqmedia":<String_value>,
    "reqspeed":<String_value>,
    "reqduplex":<String_value>,
    "reqflowcontrol":<String_value>,
    "actmedia":<String_value>,
    "actspeed":<String_value>,
    "actduplex":<String_value>,
    "actflowctl":<String_value>,
    "conndistr":<String_value>,
    "macdistr":<String_value>,
    "mode":<String_value>,
    "hamonitor":<String_value>,
    "state":<String_value>,
    "autoneg":<String_value>,
    "autonegresult":<Double_value>,
    "tagged":<Double_value>,
    "tagall":<String_value>,
    "trunk":<String_value>,
    "taggedany":<Double_value>,
    "taggedautolearn":<Double_value>,
    "hangdetect":<Double_value>,
    "hangreset":<Double_value>,
    "linkstate":<Double_value>,
    "intfstate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

"rxpackets":<Double_value>,
"rxbytes":<Double_value>,
"rxerrors":<Double_value>,
"rxdrops":<Double_value>,
"txpackets":<Double_value>,
"txbytes":<Double_value>,
"txerrors":<Double_value>,
"txdrops":<Double_value>,
"indisc":<Double_value>,
"outdisc":<Double_value>,
"fctls":<Double_value>,
"hangs":<Double_value>,
"stsstalls":<Double_value>,
"txstalls":<Double_value>,
"rxstalls":<Double_value>,
"bdgmacmoved":<Double_value>,
"bdgmuted":<Double_value>,
"vmac":<String_value>,
"vmac6":<String_value>,
"lacpmode":<String_value>,
"lacpkey":<Double_value>,
"lacppriority":<Double_value>,
"lacptimeout":<String_value>,
"lagtype":<String_value>,
"ifalias":<String_value>,
"reqthroughput":<Double_value>,
"linkredundancy":<String_value>,
"actthroughput":<Double_value>,
"bandwidthhigh":<Double_value>,
"bandwidthnormal":<Double_value>,
"backplane":<String_value>,
"ifnum":<String[]_value>,
"cleartime":<Double_value>,
"slavestate":<Double_value>,
"slavemedia":<Double_value>,
"slavespeed":<Double_value>,
"slaveduplex":<Double_value>,
"slaveflowctl":<Double_value>,
"slavetime":<Double_value>,
"intftype":<String_value>,
"lacpactormode":<String_value>,
"lacpactortimeout":<String_value>,
"lacpactorpriority":<Double_value>,
"lacpactorportno":<Double_value>,
"lacppartnerstate":<String_value>,
"lacppartnertimeout":<String_value>,
"lacppartneraggregation":<String_value>,
"lacppartnerinsync":<String_value>,
"lacppartnercollecting":<String_value>,
"lacppartnerdistributing":<String_value>,
"lacppartnerdefaulted":<String_value>,
"lacppartnerexpired":<String_value>,
"lacppartnerpriority":<Double_value>,
"lacppartnersystemmac":<String_value>,
"lacppartnersystempriority":<Double_value>,
"lacppartnerportno":<Double_value>,
"lacppartnerkey":<Double_value>,
"lacpactoraggregation":<String_value>,
"lacpactorinsync":<String_value>,
"lacpactorcollecting":<String_value>,
"lacpactordistributing":<String_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "lacpportmuxstate":<String_value>,
    "lacpportrxstat":<String_value>,
    "lacpportselectstate":<String_value>,
    "lldpmode":<String_value>,
    "lrsetpriority":<Double_value>,
    "lractiveintf":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/Interface?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "Interface": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

interfacepair

May 20, 2016

Configuration for "Interface Pair" resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	The Interface pair id. Minimum value = 1 Maximum value = 255
ifnum	<String[]>	Read-write	The constituent interfaces in the interface pair. Minimum length = 1
ifaces	<String>	Read-only	Names of all member interfaces of this Interface Pair.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/interfacepair

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"interfacepair":{
    "id":<Double_value>,
    "ifnum":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/interfacepair/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/interfacepair

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/interfacepair?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/interfacepair?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of interfacepair resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/interfacepair?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/interfacepair?**pagesize=#no&pageno=#no**

Use this query-parameter to get the interfacepair resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "interfacepair": [ {
      "id":<Double_value>,
      "ifnum":<String[]_value>,
      "ifaces":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/interfacepair/id_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/interfacepair/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/interfacepair/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "interfacepair": [ {
      "id":<Double_value>,
      "ifnum":<String[]_value>,
      "ifaces":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/interfacepair?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "interfacepair": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ip6tunnel

May 20, 2016

Configuration for ip6 Tunnel resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the IPv6 Tunnel. Cannot be changed after the service group is created. Must begin with a number or letter, and can consist of letters, numbers, and the @ _ - . (period) : (colon) # and space ( ) characters. Minimum length = 1 Maximum length = 31
remote	<String>	Read-write	An IPv6 address of the remote NetScaler appliance used to set up the tunnel. Minimum length = 1
local	<String>	Read-write	An IPv6 address of the local NetScaler appliance used to set up the tunnel.
ownergroup	<String>	Read-write	The owner node group in a Cluster for the tunnel. Default value: DEFAULT_NG Minimum length = 1
remoteip	<String>	Read-only	The remote IP address or subnet of the tunnel.
type	<Double>	Read-only	The type of this tunnel.
encapip	<String>	Read-only	The effective local IP address of the tunnel. Used as the source of the encapsulated packets.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ip6tunnel":{
    "name":<String_value>,
    "remote":<String_value>,
    "local":<String_value>,
    "ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel?**args=name:<String_value>,remote:<String_value>**

Use this query-parameter to get ip6tunnel resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ip6tunnel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the ip6tunnel resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ip6tunnel": [ {
name:<String_value>,remote:<String_value>      "remoteip":<String_value>,
      "local":<String_value>,
      "type":<Double_value>,
      "encapip":<String_value>,
      "ownergroup":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ip6tunnel": [ {
name:<String_value>,remote:<String_value>      "remoteip":<String_value>,
      "local":<String_value>,
      "type":<Double_value>,
      "encapip":<String_value>,
      "ownergroup":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/ip6tunnel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ip6tunnel": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ip6tunnelparam

May 20, 2016

Configuration for ip6 tunnel parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
srcip	<String>	Read-write	Common source IPv6 address for all IPv6 tunnels. Must be a SNIP6 or VIP6 address. Minimum length = 1
dropfrag	<String>	Read-write	Drop any packet that requires fragmentation. Default value: NO Possible values = YES, NO
dropfragcputhreshold	<Double>	Read-write	Threshold value, as a percentage of CPU usage, at which to drop packets that require fragmentation. Applies only if dropFragparameter is set to NO. Minimum value = 1 Maximum value = 100
srciproundrobin	<String>	Read-write	Use a different source IPv6 address for each new session through a particular IPv6 tunnel, as determined by round robin selection of one of the SNIP6 addresses. This setting is ignored if a common global source IPv6 address has been specified for all the IPv6 tunnels. This setting does not apply to a tunnel for which a source IPv6 address has been specified. Default value: NO Possible values = YES, NO

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/ip6tunnelparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ip6tunnelparam":{
    "srcip":<String_value>,
    "dropfrag":<String_value>,
    "dropfragcputhreshold":<Double_value>,
    "srciproundrobin":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/ip6tunnelparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"ip6tunnelparam":{
    "srcip":true,
    "dropfrag":true,
    "dropfragcputhreshold":true,
    "srciproundrobin":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ip6tunnelparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ip6tunnelparam": [ {
    "srcip":<String_value>,
    "dropfrag":<String_value>,
    "dropfragcputhreshold":<Double_value>,
    "srciproundrobin":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ipset

May 20, 2016

Configuration for network ipset resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the IP set. Must begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the IP set is created. Choose a name that helps identify the IP set. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ipset":{
    "name":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ipset?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/ipset?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ipset resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/ipset?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/ipset?**pagesize=#no&pageno=#no**

Use this query-parameter to get the ipset resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ipset": [ {
      "name":<String_value>,
      "td":<Double_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ipset/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/ipset/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ipset": [ {
      "name":<String_value>,
      "td":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "ipset": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ipset_binding

Jul 07, 2017

Binding object which returns the resources bound to ipset.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the IP set whose details you want to display. Minimum length = 1
ipset_nsip_binding	<ipset_nsip_binding[]>	Read-only	nsip that can be bound to ipset.
ipset_nsip6_binding	<ipset_nsip6_binding[]>	Read-only	nsip6 that can be bound to ipset.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "ipset_binding": [ {
      "name":<String_value>,
      "ipset_nsip_binding":<ipset_nsip_binding[]_value>,
      "ipset_nsip6_binding":<ipset_nsip6_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ipset_nsip6_binding

Jul 07, 2017

Binding object showing the nsip6 that can be bound to ipset.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the IP set to which to bind IP addresses. Minimum length = 1
ipaddress	<String>	Read-write	One or more IP addresses bound to the IP set. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/ipset_nsip6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"ipset_nsip6_binding":{
    "name":<String_value>,
    "ipaddress":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset_nsip6_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset_nsip6_binding/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/ipset_nsip6_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of ipset_nsip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/ipset_nsip6_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the ipset_nsip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ipset_nsip6_binding": [ {
      "name":<String_value>,
      "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset_nsip6_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"ipset_nsip6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ipset_nsip_binding

Jul 07, 2017

Binding object showing the nsip that can be bound to ipset.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the IP set to which to bind IP addresses. Minimum length = 1
ipaddress	<String>	Read-write	One or more IP addresses bound to the IP set. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/ipset_nsip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"ipset_nsip_binding":{
    "name":<String_value>,
    "ipaddress":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset_nsip_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset_nsip_binding/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/ipset_nsip_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of ipset_nsip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/ipset_nsip_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the ipset_nsip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ipset_nsip_binding": [ {
      "name":<String_value>,
      "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/ipset_nsip_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"ipset_nsip_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

iptunnel

May 20, 2016

Configuration for ip Tunnel resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the IP tunnel. Leading character must be a number or letter. Other characters allowed, after the first character, are @ _ - . (period) : (colon) # and space ( ). Minimum length = 1
remote	<String>	Read-write	Public IPv4 address, of the remote device, used to set up the tunnel. For this parameter, you can alternatively specify a network address. Minimum length = 1
remotesubnetmask	<String>	Read-write	Subnet mask of the remote IP address of the tunnel.
local	<String>	Read-write	Type ofNetScaler owned public IPv4 address, configured on the local NetScaler appliance and used to set up the tunnel.
protocol	<String>	Read-write	Name of the protocol to be used on this tunnel. Default value: IPIP Possible values = IPIP, GRE, IPSEC, VXLAN
grepayload	<String>	Read-write	The payload GRE will carry. Default value: ETHERNETwithDOT1Q Possible values = ETHERNETwithDOT1Q, ETHERNET, IP
ipsecprofilename	<String>	Read-write	Name of IPSec profile to be associated. Default value: "ns_ipsec_default_profile"
vlan	<Double>	Read-write	The vlan for mulicast packets. Minimum value = 1 Maximum value = 4094
ownergroup	<String>	Read-write	The owner node group in a Cluster for the iptunnel. Default value: DEFAULT_NG Minimum length = 1
sysname	<String>	Read-only	The name of the ip tunnel.
type	<Double>	Read-only	The type of this tunnel.
encapip	<String>	Read-only	The effective local IP address of the tunnel. Used as the source of the encapsulated packets.
channel	<Double>	Read-only	The tunnel that is bound to a netbridge.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
tunneltype	<String[]>	Read-only	Possible values = Configured, Delete-In-Progress
ipsectunnelstatus	<String>	Read-only	Whether the ipsec on this tunnel is up or down. Possible values = DOWN, UP, PARTIAL-UP, UNKNOWN
pbrname	<String>	Read-only	Name for the PBR.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/iptunnel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"iptunnel":{
    "name":<String_value>,
    "remote":<String_value>,
    "remotesubnetmask":<String_value>,
    "local":<String_value>,
    "protocol":<String_value>,
    "grepayload":<String_value>,
    "ipsecprofilename":<String_value>,
    "vlan":<Double_value>,
    "ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/iptunnel/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/iptunnel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/iptunnel?**args=remote:<String_value>,remotesubnetmask:<String_value>,name:<String_value>**

Use this query-parameter to get iptunnel resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/iptunnel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/iptunnel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of iptunnel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/iptunnel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/iptunnel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the iptunnel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "iptunnel": [ {
remote:<String_value>,remotesubnetmask:<String_value>,name:<String_value>      "sysname":<String_value>,
      "local":<String_value>,
      "protocol":<String_value>,
      "grepayload":<String_value>,
      "type":<Double_value>,
      "encapip":<String_value>,
      "channel":<Double_value>,
      "ipsecprofilename":<String_value>,
      "vlan":<Double_value>,
      "tunneltype":<String[]_value>,
      "ipsectunnelstatus":<String_value>,
      "pbrname":<String_value>,
      "ownergroup":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/iptunnel/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/iptunnel/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/iptunnel/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "iptunnel": [ {
remote:<String_value>,remotesubnetmask:<String_value>,name:<String_value>      "sysname":<String_value>,
      "local":<String_value>,
      "protocol":<String_value>,
      "grepayload":<String_value>,
      "type":<Double_value>,
      "encapip":<String_value>,
      "channel":<Double_value>,
      "ipsecprofilename":<String_value>,
      "vlan":<Double_value>,
      "tunneltype":<String[]_value>,
      "ipsectunnelstatus":<String_value>,
      "pbrname":<String_value>,
      "ownergroup":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/iptunnel?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "iptunnel": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

iptunnelparam

May 20, 2016

Configuration for ip tunnel parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
srcip	<String>	Read-write	Common source-IP address for all tunnels. For a specific tunnel, this global setting is overridden if you have specified another source IP address. Must be a MIP or SNIP address. Minimum length = 1
dropfrag	<String>	Read-write	Drop any IP packet that requires fragmentation before it is sent through the tunnel. Default value: NO Possible values = YES, NO
dropfragcputhreshold	<Double>	Read-write	Threshold value, as a percentage of CPU usage, at which to drop packets that require fragmentation to use the IP tunnel. Applies only if dropFragparameter is set to NO. The default value, 0, specifies that this parameter is not set. Minimum value = 1 Maximum value = 100
srciproundrobin	<String>	Read-write	Use a different source IP address for each new session through a particular IP tunnel, as determined by round robin selection of one of the SNIP addresses. This setting is ignored if a common global source IP address has been specified for all the IP tunnels. This setting does not apply to a tunnel for which a source IP address has been specified. Default value: NO Possible values = YES, NO
enablestrictrx	<String>	Read-write	Strict PBR check for IPSec packets received through tunnel. Default value: NO Possible values = YES, NO
enablestricttx	<String>	Read-write	Strict PBR check for packets to be sent IPSec protected. Default value: NO Possible values = YES, NO

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/iptunnelparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"iptunnelparam":{
    "srcip":<String_value>,
    "dropfrag":<String_value>,
    "dropfragcputhreshold":<Double_value>,
    "srciproundrobin":<String_value>,
    "enablestrictrx":<String_value>,
    "enablestricttx":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/iptunnelparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"iptunnelparam":{
    "srcip":true,
    "dropfrag":true,
    "dropfragcputhreshold":true,
    "srciproundrobin":true,
    "enablestrictrx":true,
    "enablestricttx":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/iptunnelparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "iptunnelparam": [ {
      "srcip":<String_value>,
      "dropfrag":<String_value>,
      "dropfragcputhreshold":<Double_value>,
      "srciproundrobin":<String_value>,
      "enablestrictrx":<String_value>,
      "enablestricttx":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ipv6

Jul 07, 2017

Configuration for ip v6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ralearning	<String>	Read-write	Enable the NetScaler appliance to learn about various routes from Router Advertisement (RA) and Router Solicitation (RS) messages sent by the routers. Default value: DISABLED Possible values = ENABLED, DISABLED
routerredirection	<String>	Read-write	Enable the NetScaler appliance to do Router Redirection. Default value: DISABLED Possible values = ENABLED, DISABLED
ndbasereachtime	<Double>	Read-write	Base reachable time of the Neighbor Discovery (ND6) protocol. The time, in milliseconds, that the NetScaler appliance assumes an adjacent device is reachable after receiving a reachability confirmation. Default value: 30000 Minimum value = 1
ndretransmissiontime	<Double>	Read-write	Retransmission time of the Neighbor Discovery (ND6) protocol. The time, in milliseconds, between retransmitted Neighbor Solicitation (NS) messages, to an adjacent device. Default value: 1000 Minimum value = 1
natprefix	<String>	Read-write	Prefix used for translating packets from private IPv6 servers to IPv4 packets. This prefix has a length of 96 bits (128-32 = 96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix. IPv6 packets addressed to this prefix have to be routed to the NetScaler appliance to ensure that the IPv6-IPv4 translation is done by the appliance.
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
dodad	<String>	Read-write	Enable the NetScaler appliance to do Duplicate Address Detection (DAD) for all the NetScaler owned IPv6 addresses regardless of whether they are obtained through stateless auto configuration, DHCPv6, or manual configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = ENABLED, DISABLED
basereachtime	<Integer>	Read-only	ND6 base reachable time (ms).
reachtime	<Integer>	Read-only	ND6 computed reachable time (ms).
ndreachtime	<Double>	Read-only	ND6 computed reachable time (ms).
retransmissiontime	<Integer>	Read-only	ND6 retransmission time (ms).
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/ipv6

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ipv6":{
    "ralearning":<String_value>,
    "routerredirection":<String_value>,
    "ndbasereachtime":<Double_value>,
    "ndretransmissiontime":<Double_value>,
    "natprefix":<String_value>,
    "td":<Double_value>,
    "dodad":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/ipv6?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ipv6":{
    "ralearning":true,
    "routerredirection":true,
    "ndbasereachtime":true,
    "ndretransmissiontime":true,
    "natprefix":true,
    "td":<Double_value>,
    "dodad":true
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ipv6

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ipv6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/ipv6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ipv6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/ipv6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/ipv6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the ipv6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "ipv6": [ {
      "td":<Double_value>,
      "ralearning":<String_value>,
      "routerredirection":<String_value>,
      "basereachtime":<Integer_value>,
      "ndbasereachtime":<Double_value>,
      "reachtime":<Integer_value>,
      "ndreachtime":<Double_value>,
      "retransmissiontime":<Integer_value>,
      "ndretransmissiontime":<Double_value>,
      "natprefix":<String_value>,
      "dodad":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/ipv6/td_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ipv6/td_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/ipv6/td_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "ipv6": [ {
    "td":<Double_value>,
    "ralearning":<String_value>,
    "routerredirection":<String_value>,
    "basereachtime":<Integer_value>,
    "ndbasereachtime":<Double_value>,
    "reachtime":<Integer_value>,
    "ndreachtime":<Double_value>,
    "retransmissiontime":<Integer_value>,
    "ndretransmissiontime":<Double_value>,
    "natprefix":<String_value>,
    "dodad":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/ipv6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ipv6": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

l2param

Jul 07, 2017

Configuration for Layer 2 related parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
mbfpeermacupdate	<Double>	Read-write	When mbf_instant_learning is enabled, learn any changes in peer's MAC after this time interval, which is in 10ms ticks. Default value: 10
maxbridgecollision	<Double>	Read-write	Maximum bridge collision for loop detection . Default value: 20
bdggrpproxyarp	<String>	Read-write	Set/reset proxy ARP in bridge group deployment. Default value: ENABLED Possible values = ENABLED, DISABLED
bdgsetting	<String>	Read-write	Bridging settings for C2C behavior. Default value: DISABLED Possible values = ENABLED, DISABLED
garponvridintf	<String>	Read-write	Send GARP messagess on VRID-configured interfaces upon failover . Default value: ENABLED Possible values = ENABLED, DISABLED
macmodefwdmypkt	<String>	Read-write	MAC mode vserver forward packets destined to VIPs. Default value: DISABLED Possible values = ENABLED, DISABLED
usemymac	<String>	Read-write	Set/reset cfg_use_my_mac . Default value: DISABLED Possible values = ENABLED, DISABLED
proxyarp	<String>	Read-write	Set/reset cfg_proxy_arp_dr . Default value: ENABLED Possible values = ENABLED, DISABLED
garpreply	<String>	Read-write	Set/reset REPLY form of GARP . Default value: DISABLED Possible values = ENABLED, DISABLED
mbfinstlearning	<String>	Read-write	Enable instant learning of MAC changes in MBF mode. Default value: DISABLED Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
[illegible]	<String>	Read-write	Default value: DISABLED Possible values = ENABLED, DISABLED
skipproxyingbsdtraffic	<String>	Read-write	Enable the proxying of FreeBSD traffic. Default value: DISABLED Possible values = ENABLED, DISABLED
returntoethernetsender	<String>	Read-write	Return to ethernet sender. Default value: DISABLED Possible values = ENABLED, DISABLED
stopmacmoveupdate	<String>	Read-write	Stop propagation of server mac change to natpcbs. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/l2param

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"l2param":{
    "mbfpeermacupdate":<Double_value>,
    "maxbridgecollision":<Double_value>,
    "bdggrpproxyarp":<String_value>,
    "bdgsetting":<String_value>,
    "garponvridintf":<String_value>,
    "macmodefwdmypkt":<String_value>,
    "usemymac":<String_value>,
    "proxyarp":<String_value>,
    "garpreply":<String_value>,
    "mbfinstlearning":<String_value>,
    "rstintfonhafo":<String_value>,
    "skipproxyingbsdtraffic":<String_value>,
    "returntoethernetsender":<String_value>,
    "stopmacmoveupdate":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/l2param?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"l2param":{
    "mbfpeermacupdate":true,
    "maxbridgecollision":true,
    "bdggrpproxyarp":true,
    "bdgsetting":true,
    "garponvridintf":true,
    "macmodefwdmypkt":true,
    "usemymac":true,
    "proxyarp":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "garpreply":true,
    "mbfinstlearning":true,
    "rstintfonhafo":true,
    "skipproxyingbsdtraffic":true,
    "returntoethernetsender":true,
    "stopmacmoveupdate":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/l2param

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "l2param": [ {
    "maxbridgecollision":<Double_value>,
    "mbfpeermacupdate":<Double_value>,
    "bdggrpproxyarp":<String_value>,
    "bdgsetting":<String_value>,
    "garponvridintf":<String_value>,
    "macmodefwdmypkt":<String_value>,
    "usemymac":<String_value>,
    "proxyarp":<String_value>,
    "garpreply":<String_value>,
    "mbfinstlearning":<String_value>,
    "rstintfonhafo":<String_value>,
    "skipproxyingbsdtraffic":<String_value>,
    "returntoethernetsender":<String_value>,
    "stopmacmoveupdate":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

l3param

May 20, 2016

Configuration for Layer 3 related parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
srcnat	<String>	Read-write	Perform NAT if only the source is in the private network. Default value: ENABLED Possible values = ENABLED, DISABLED
icmpgenratethreshold	<Double>	Read-write	NS generated ICMP pkts per 10ms rate threshold. Default value: 100
overridernat	<String>	Read-write	USNIP/USIP settings override RNAT settings for configured service/virtual server traffic... . Default value: DISABLED Possible values = ENABLED, DISABLED
dropdfflag	<String>	Read-write	Enable dropping the IP DF flag. Default value: DISABLED Possible values = ENABLED, DISABLED
miproundrobin	<String>	Read-write	Enable round robin usage of mapped IPs. Default value: ENABLED Possible values = ENABLED, DISABLED
externalloopback	<String>	Read-write	Enable external loopback. Default value: DISABLED Possible values = ENABLED, DISABLED
tnlpmtuwoconn	<String>	Read-write	Enable external loopback. Default value: ENABLED Possible values = ENABLED, DISABLED
usipserverstraypkt	<String>	Read-write	Enable detection of stray server side pkts in USIP mode. Default value: DISABLED Possible values = ENABLED, DISABLED
forwardicmpfragments	<String>	Read-write	Enable forwarding of ICMP fragments. Default value: DISABLED Possible values = ENABLED, DISABLED
dropipfragments	<String>	Read-write	Enable dropping of IP fragments. Default value: DISABLED Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
acllogtime	<Double>	Read-write	Default value: 5000
icmperrgenerate	<String>	Read-write	Enable/Disable fragmentation required icmp error generation, before encapsulating a packet with vPath header. This knob is only functional for vPath Environment. Default value: ENABLED Possible values = ENABLED, DISABLED
overridelsn	<String>	Read-write	USNIP/USIP settings override LSN settings for configured service/virtual server traffic.. . Default value: DISABLED Possible values = ENABLED, DISABLED
implicitaclallow	<String>	Read-write	Do not apply ACLs for internal ports. Default value: ENABLED Possible values = ENABLED, DISABLED
dynamicrouting	<String>	Read-write	Enable/Disable Dynamic routing on partition. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/l3param

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"l3param":{
    "srcnat":<String_value>,
    "icmpgenratethreshold":<Double_value>,
    "overridernat":<String_value>,
    "dropdfflag":<String_value>,
    "miproundrobin":<String_value>,
    "externalloopback":<String_value>,
    "tnlpmtuwoconn":<String_value>,
    "usipserverstraypkt":<String_value>,
    "forwardicmpfragments":<String_value>,
    "dropipfragments":<String_value>,
    "acllogtime":<Double_value>,
    "icmperrgenerate":<String_value>,
    "overridelsn":<String_value>,
    "implicitaclallow":<String_value>,
    "dynamicrouting":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/l3param?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"l3param":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "srcnat":true,
    "icmpgenratethreshold":true,
    "overridernat":true,
    "dropdfflag":true,
    "miproundrobin":true,
    "externalloopback":true,
    "tnlpmtuwoconn":true,
    "usipserverstraypkt":true,
    "forwardicmpfragments":true,
    "dropipfragments":true,
    "acllogtime":true,
    "icmperrgenerate":true,
    "overridelsn":true,
    "implicitaclallow":true,
    "dynamicrouting":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/l3param

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "l3param": [ {
    "srcnat":<String_value>,
    "icmpgenratethreshold":<Double_value>,
    "overridernat":<String_value>,
    "dropdfflag":<String_value>,
    "miproundrobin":<String_value>,
    "externalloopback":<String_value>,
    "tnlpmtuwoconn":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "usipserverstraypkt":<String_value>,
    "forwardicmpfragments":<String_value>,
    "dropipfragments":<String_value>,
    "acllogtime":<Double_value>,
    "icmperrgenerate":<String_value>,
    "overridelsn":<String_value>,
    "implicitaclallow":<String_value>,
    "dynamicrouting":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

l4param

May 20, 2016

Configuration for Layer 4 related parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
l2connmethod	<String>	Read-write	Layer 2 connection method based on the combination of channel number, MAC address and VLAN. It is tuned with l2conn param of lb vserver. If l2conn of lb vserver is ON then method specified here will be used to identify a connection in addition to the 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>). Default value: MacVlanChannel Possible values = Channel, Vlan, VlanChannel, Mac, MacChannel, MacVlan, MacVlanChannel
l4switch	<String>	Read-write	In L4 switch topology, always clients and servers are on the same side. Enable l4switch to allow such connections. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/l4param

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"l4param":{
    "l2connmethod":<String_value>,
    "l4switch":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/l4param?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"l4param":{
    "l2connmethod":true,
    "l4switch":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/l4param

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "l4param": [ {
      "l2connmethod":<String_value>,
      "l4switch":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lacp

May 20, 2016

Configuration for Link aggregation control protocol resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
syspriority	<Double>	Read-write	Priority number that determines which peer device of an LACP LA channel can have control over the LA channel. This parameter is globally applied to all LACP channels on the NetScaler appliance. The lower the number, the higher the priority. Default value: 32768 Minimum value = 1 Maximum value = 65535
ownernode	<Double>	Read-write	The owner node in a cluster for which we want to set the lacp priority. Owner node can vary from 0 to 31. Ownernode value of 254 is used for Cluster. Default value: 255
devicename	<String>	Read-only	Name of the channel.
mac	<String>	Read-only	LACP system MAC.
flags	<Double>	Read-only	Flags of this channel.
lacpkey	<Double>	Read-only	LACP key of this channel.
clustersyspriority	<Double>	Read-only	LACP system (Cluster) priority.
clustermac	<String>	Read-only	LACP system (Cluster) mac.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/lacp

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"lacp":{
    "syspriority":<Double_value>,
    "ownernode":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/lacp

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lacp?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/lacp?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of lacp resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/lacp?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/lacp?**pagesize=#no&pageno=#no**

Use this query-parameter to get the lacp resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lacp": [ {
      "ownernode":<Double_value>,
      "devicename":<String_value>,
      "syspriority":<Double_value>,
      "mac":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "flags":<Double_value>,
    "lacpkey":<Double_value>,
    "clustersyspriority":<Double_value>,
    "clustermac":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/lacp/ownernode_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/lacp/ownernode_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/lacp/ownernode_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lacp": [ {
    "ownernode":<Double_value>,
    "devicename":<String_value>,
    "syspriority":<Double_value>,
    "mac":<String_value>,
    "flags":<Double_value>,
    "lacpkey":<Double_value>,
    "clustersyspriority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "clustermac":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/lacp?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lacp": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

linkset

May 20, 2016

Configuration for link set resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<String>	Read-write	Unique identifier for the linkset. Must be of the form LS/x, where x can be an integer from 1 to 32.
ifnum	<String>	Read-only	The interfaces to be bound to the linkset.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"linkset":{
    "id":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset/id_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/linkset?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/linkset?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of linkset resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/linkset?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/linkset?**pagesize=#no&pageno=#no**

Use this query-parameter to get the linkset resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "linkset": [ {
      "id":<String_value>,
      "ifnum":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset/id_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/linkset/id_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/linkset/id_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "linkset": [ {
      "id":<String_value>,
      "ifnum":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "linkset": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ci

May 20, 2016

Configuration for "CI" resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ifaces	<String>	Read-only	Interfaces that are critical for the appliance to operate in high availability mode.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ci

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ci?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/ci?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ci resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/ci?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/ci?**pagesize=#no&pageno=#no**

Use this query-parameter to get the ci resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ci": [ {
      "ifaces":<String_value>
}]}
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/ci?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ci": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

linkset_binding

Jul 07, 2017

Binding object which returns the resources bound to linkset.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<String>	Read-write	ID of the linkset for which to display information. If an ID is not provided, the display includes information about all linksets that are available in the cluster.
linkset_interface_binding	<linkset_interface_binding[]>	Read-only	interface that can be bound to linkset.
linkset_channel_binding	<linkset_channel_binding[]>	Read-only	channel that can be bound to linkset.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset_binding/id_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "linkset_binding": [ {
    "id":<String_value>,
    "linkset_interface_binding":<linkset_interface_binding[]_value>,
    "linkset_channel_binding":<linkset_channel_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

linkset_channel_binding

May 20, 2016

Binding object showing the channel that can be bound to linkset.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<String>	Read-write	ID of the linkset to which to bind the interfaces.
ifnum	<String>	Read-write	The interfaces to be bound to the linkset.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/linkset_channel_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"linkset_channel_binding":{
    "id":<String_value>,
    "ifnum":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset_channel_binding/id_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset_channel_binding/id_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/linkset_channel_binding/id_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of linkset_channel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/linkset_channel_binding/id_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the linkset_channel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "linkset_channel_binding": [ {
      "id":<String_value>,
      "ifnum":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset_channel_binding/id_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"linkset_channel_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

linkset_interface_binding

May 20, 2016

Binding object showing the interface that can be bound to linkset.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<String>	Read-write	ID of the linkset to which to bind the interfaces.
ifnum	<String>	Read-write	The interfaces to be bound to the linkset.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/linkset_interface_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"linkset_interface_binding":{
    "id":<String_value>,
    "ifnum":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset_interface_binding/id_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset_interface_binding/id_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/linkset_interface_binding/id_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of linkset_interface_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/linkset_interface_binding/id_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the linkset_interface_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "linkset_interface_binding": [ {
    "id":<String_value>,
    "ifnum":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/linkset_interface_binding/id_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"linkset_interface_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nat64

Jul 07, 2017

Configuration for nat64 config resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the NAT64 rule. Must begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the rule is created. Choose a name that helps identify the NAT64 rule. Minimum length = 1
acl6name	<String>	Read-write	Name of any configured ACL6 whose action is ALLOW. IPv6 Packets matching the condition of this ACL6 rule and destination IP address of these packets matching the NAT64 IPv6 prefix are considered for NAT64 translation. Minimum length = 1
netprofile	<String>	Read-write	Name of the configured netprofile. The NetScaler appliance selects one of the IP address in the netprofile as the source IP address of the translated IPv4 packet to be sent to the IPv4 server. Minimum length = 1 Maximum length = 127
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nat64

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nat64":{
    "name":<String_value>,
    "acl6name":<String_value>,
    "netprofile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nat64

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"nat64":{
    "name":<String_value>,
    "acl6name":<String_value>,
    "netprofile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nat64?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nat64":{
    "name":<String_value>,
    "netprofile":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nat64/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nat64

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nat64?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nat64?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nat64 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nat64?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nat64?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nat64 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "nat64": [ {
      "name":<String_value>,
      "acl6name":<String_value>,
      "netprofile":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nat64/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nat64/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nat64/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nat64": [ {
      "name":<String_value>,
      "acl6name":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "netprofile":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nat64?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nat64": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nd6

May 20, 2016

Configuration for nd6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
neighbor	<String>	Read-write	Link-local IPv6 address of the adjacent network device to add to the ND6 table.
mac	<String>	Read-write	MAC address of the adjacent network device.
ifnum	<String>	Read-write	Interface through which the adjacent network device is available, specified in slot/port notation (for example, 1/3). Use spaces to separate multiple entries.
vlan	<Integer>	Read-write	Integer value that uniquely identifies the VLAN on which the adjacent network device exists. Minimum value = 1 Maximum value = 4094
vxlan	<Double>	Read-write	ID of the VXLAN on which the IPv6 address of this ND6 entry is reachable. Minimum value = 1 Maximum value = 16777215
vtep	<String>	Read-write	IP address of the VXLAN tunnel endpoint (VTEP) through which the IPv6 address of this ND6 entry is reachable. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
state	<String>	Read-only	ND6 state. Default value: REACHABLE Possible values = INCOMPLETE, REACHABLE, STALE, DELAY, PROBE
timeout	<Double>	Read-only	Time elapsed.
flags	<Double>	Read-only	flag for static/permanent entry.
channel	<Double>	Read-only	The tunnel that is bound to a netbridge.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD | CLEAR | DELETE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nd6":{
    "neighbor":<String_value>,
    "mac":<String_value>,
    "ifnum":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "vlan":<Integer_value>,
    "vxlan":<Double_value>,
    "vtep":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nd6":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6/neighbor_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nd6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nd6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nd6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nd6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nd6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nd6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nd6": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "neighbor":<String_value>,
        "td":<Double_value>,
        "mac":<String_value>,
        "state":<String_value>,
        "timeout":<Double_value>,
        "ifnum":<String_value>,
        "vlan":<Integer_value>,
        "vxlan":<Double_value>,
        "vtep":<String_value>,
        "flags":<Double_value>,
        "channel":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nd6": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nd6ravariables

May 20, 2016

Configuration for nd6 Router Advertisment configuration variables resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vlan	<Double>	Read-write	The VLAN number. Minimum value = 0 Maximum value = 4094
ceaserouteradv	<String>	Read-write	Cease router advertisements on this vlan. Default value: NO Possible values = YES, NO
sendrouteradv	<String>	Read-write	whether the router sends periodic RAs and responds to Router Solicitations. Default value: NO Possible values = YES, NO
srclinklayeraddroption	<String>	Read-write	Include source link layer address option in RA messages. Default value: YES Possible values = YES, NO
onlyunicastrtadvresponse	<String>	Read-write	Send only Unicast Router Advertisements in respond to Router Solicitations. Default value: NO Possible values = YES, NO
managedaddrconfig	<String>	Read-write	Value to be placed in the Managed address configuration flag field. Default value: NO Possible values = YES, NO
otheraddrconfig	<String>	Read-write	Value to be placed in the Other configuration flag field. Default value: NO Possible values = YES, NO
currhoplimit	<Double>	Read-write	Current Hop limit. Default value: 64 Minimum value = 0 Maximum value = 255
maxrtadvinterval	<Double>	Read-write	Maximum time allowed between unsolicited multicast RAs, in seconds. Default value: 600 Minimum value = 4 Maximum value = 1800
			Minimum time interval between RA messages, in seconds.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 3 Maximum value = 1350
linkmtu	<Double>	Read-write	The Link MTU. Default value: 0 Minimum value = 0 Maximum value = 1500
reachabletime	<Double>	Read-write	Reachable time, in milliseconds. Default value: 0 Minimum value = 0 Maximum value = 3600000
retranstime	<Double>	Read-write	Retransmission time, in milliseconds. Default value: 0
defaultlifetime	<Integer>	Read-write	Default life time, in seconds. Default value: 1800 Minimum value = 0 Maximum value = 9000
lastrtadvtime	<Double>	Read-only	Last RA sent timestamp.
nextrtadvdelay	<Double>	Read-only	Next RA delay.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nd6ravariables":{
    "vlan":<Double_value>,
    "ceaserouteradv":<String_value>,
    "sendrouteradv":<String_value>,
    "srclinklayeraddroption":<String_value>,
    "onlyunicastrtadvresponse":<String_value>,
    "managedaddrconfig":<String_value>,
    "otheraddrconfig":<String_value>,
    "currhoplimit":<Double_value>,
    "maxrtadvinterval":<Double_value>,
    "minrtadvinterval":<Double_value>,
    "linkmtu":<Double_value>,
    "reachabletime":<Double_value>,
    "retranstime":<Double_value>,
    "defaultlifetime":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables?**action=unset**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nd6ravariables":{
    "vlan":<Double_value>,
    "ceaserouteradv":true,
    "sendrouteradv":true,
    "srclinklayeraddroption":true,
    "onlyunicastrtadvresponse":true,
    "managedaddrconfig":true,
    "otheraddrconfig":true,
    "currhoplimit":true,
    "maxrtadvinterval":true,
    "minrtadvinterval":true,
    "linkmtu":true,
    "reachabletime":true,
    "retranstime":true,
    "defaultlifetime":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nd6ravariables resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nd6ravariables resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nd6ravariables": [ {
    "vlan":<Double_value>,
    "ceaserouteradv":<String_value>,
    "sendrouteradv":<String_value>,
    "srclinklayeraddroption":<String_value>,
    "onlyunicastrtadvresponse":<String_value>,
    "managedaddrconfig":<String_value>,
    "otheraddrconfig":<String_value>,
    "currhoplimit":<Double_value>,
    "maxrtadvinterval":<Double_value>,
    "minrtadvinterval":<Double_value>,
    "linkmtu":<Double_value>,
    "reachabletime":<Double_value>,
    "retranstime":<Double_value>,
    "defaultlifetime":<Integer_value>,
    "lastrtadvtime":<Double_value>,
    "nextrtadvdelay":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables/vlan_value<Double>

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables/vlan_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables/vlan_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nd6ravariables": [ {
    "vlan":<Double_value>,
    "ceaserouteradv":<String_value>,
    "sendrouteradv":<String_value>,
    "srclinklayeraddroption":<String_value>,
    "onlyunicastrtadvresponse":<String_value>,
    "managedaddrconfig":<String_value>,
    "otheraddrconfig":<String_value>,
    "currhoplimit":<Double_value>,
    "maxrtadvinterval":<Double_value>,
    "minrtadvinterval":<Double_value>,
    "linkmtu":<Double_value>,
    "reachabletime":<Double_value>,
    "retranstime":<Double_value>,
    "defaultlifetime":<Integer_value>,
    "lastrtadvtime":<Double_value>,
    "nextrtadvdelay":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "nd6ravariables": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nd6ravariables_binding

Jul 07, 2017

Binding object which returns the resources bound to nd6ravariables.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vlan	<Double>	Read-write	The VLAN number. Minimum value = 0 Maximum value = 4094
nd6ravariables_onlinkipv6prefix_binding	<nd6ravariables_onlinkipv6prefix_binding[]>	Read-only	onlinkipv6prefix that can be bound to nd6ravariables.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables_binding/vlan_value<Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "nd6ravariables_binding": [ {
      "vlan":<Double_value>,
      "nd6ravariables_onlinkipv6prefix_binding":<nd6ravariables_onlinkipv6prefix_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nd6ravariables_onlinkipv6prefix_binding

May 20, 2016

Binding object showing the onlinkipv6prefix that can be bound to nd6ravariables.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipv6prefix	<String>	Read-write	Onlink prefixes for RA messages.
vlan	<Double>	Read-write	The VLAN number. Minimum value = 0 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"nd6ravariables_onlinkipv6prefix_binding":{
    "vlan":<Double_value>,
    "ipv6prefix":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nd6ravariables_onlinkipv6prefix_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nd6ravariables_onlinkipv6prefix_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nd6ravariables_onlinkipv6prefix_binding": [ {
    "ipv6prefix":<String_value>,
    "vlan":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nd6ravariables_onlinkipv6prefix_binding/vlan_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"nd6ravariables_onlinkipv6prefix_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

netbridge

May 20, 2016

Configuration for network bridge resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the network bridge.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"netbridge":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/netbridge?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/netbridge?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of netbridge resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/netbridge?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/netbridge?**pagesize=#no&pageno=#no**

Use this query-parameter to get the netbridge resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netbridge": [ {
    "name":<String_value>,

}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/netbridge/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/netbridge/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netbridge": [ {
      "name":<String_value>,

}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "netbridge": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

netbridge_binding

May 20, 2016

Binding object which returns the resources bound to netbridge.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the network bridge.
netbridge_iptunnel_binding	<netbridge_iptunnel_binding[]>	Read-only	iptunnel that can be bound to netbridge.
netbridge_nsip_binding	<netbridge_nsip_binding[]>	Read-only	nsip that can be bound to netbridge.
netbridge_vlan_binding	<netbridge_vlan_binding[]>	Read-only	vlan that can be bound to netbridge.
netbridge_nsip6_binding	<netbridge_nsip6_binding[]>	Read-only	nsip6 that can be bound to netbridge.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "netbridge_binding": [ {
    "name":<String_value>,
    "netbridge_nsip6_binding":<netbridge_nsip6_binding[]_value>,
    "netbridge_iptunnel_binding":<netbridge_iptunnel_binding[]_value>,
    "netbridge_vlan_binding":<netbridge_vlan_binding[]_value>,
    "netbridge_nsip_binding":<netbridge_nsip_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

netbridge_iptunnel_binding

May 20, 2016

Binding object showing the iptunnel that can be bound to netbridge.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the network bridge.
tunnel	<String>	Read-write	The name of the tunnel that is a part of this bridge.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/netbridge_iptunnel_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"netbridge_iptunnel_binding":{
    "name":<String_value>,
    "tunnel":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netbridge_iptunnel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the netbridge_iptunnel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netbridge_iptunnel_binding": [ {
    "name":<String_value>,
    "tunnel":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_iptunnel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"netbridge_iptunnel_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

netbridge_nsip6_binding

May 20, 2016

Binding object showing the nsip6 that can be bound to netbridge.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the network bridge.
netmask	<String>	Read-write	The network mask for the subnet.
ipaddress	<String>	Read-write	The subnet that is extended by this network bridge. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/netbridge_nsip6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"netbridge_nsip6_binding":{
    "name":<String_value>,
    "ipaddress":<String_value>,
    "netmask":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netbridge_nsip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the netbridge_nsip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netbridge_nsip6_binding": [ {
    "name":<String_value>,
    "netmask":<String_value>,
    "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_nsip6_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"netbridge_nsip6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

netbridge_nsip_binding

May 20, 2016

Binding object showing the nsip that can be bound to netbridge.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the network bridge.
netmask	<String>	Read-write	The network mask for the subnet.
ipaddress	<String>	Read-write	The subnet that is extended by this network bridge. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/netbridge_nsip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"netbridge_nsip_binding":{
    "name":<String_value>,
    "ipaddress":<String_value>,
    "netmask":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_nsip_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_nsip_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/netbridge_nsip_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netbridge_nsip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/netbridge_nsip_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the netbridge_nsip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netbridge_nsip_binding": [ {
    "name":<String_value>,
    "netmask":<String_value>,
    "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_nsip_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"netbridge_nsip_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

netbridge_vlan_binding

May 20, 2016

Binding object showing the vlan that can be bound to netbridge.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the network bridge.
vlan	<Double>	Read-write	The VLAN that is extended by this network bridge. Minimum value = 1 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/netbridge_vlan_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"netbridge_vlan_binding":{
    "name":<String_value>,
    "vlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_vlan_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_vlan_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/netbridge_vlan_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netbridge_vlan_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/netbridge_vlan_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the netbridge_vlan_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netbridge_vlan_binding": [ {
    "name":<String_value>,
    "vlan":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/netbridge_vlan_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"netbridge_vlan_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

netprofile_binding

Jul 07, 2017

Binding object which returns the resources bound to netprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the net profile whose details you want to display. Minimum length = 1
netprofile_srcportset_binding	<netprofile_srcportset_binding[]>	Read-only	srcportset that can be bound to netprofile.
netprofile_natrule_binding	<netprofile_natrule_binding[]>	Read-only	natrule that can be bound to netprofile.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "netprofile_binding": [ {
    "name":<String_value>,
    "netprofile_srcportset_binding":<netprofile_srcportset_binding[]_value>,
    "netprofile_natrule_binding":<netprofile_natrule_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

netprofile

May 20, 2016

Configuration for Network profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the net profile. Must begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the profile is created. Choose a name that helps identify the net profile. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcip	<String>	Read-write	IP address or the name of an IP set.
srcippersistency	<String>	Read-write	When the net profile is associated with a virtual server or its bound services, this option enables the NetScaler appliance to use the same address, specified in the net profile, to communicate to servers for all sessions initiated from a particular client to the virtual server. Default value: DISABLED Possible values = ENABLED, DISABLED
overridelsn	<String>	Read-write	USNIP/USIP settings override LSN settings for configured service/virtual server traffic.. . Default value: DISABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"netprofile":{
    "name":<String_value>,
    "td":<Double_value>,
    "srcip":<String_value>,
    "srcippersistency":<String_value>,
    "overridelsn":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"netprofile":{
    "name":<String_value>,
    "srcip":<String_value>,
    "srcippersistency":<String_value>,
    "overridelsn":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile?**action=unset**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"netprofile":{
    "name":<String_value>,
    "srcip":true,
    "srcippersistency":true,
    "overridelsn":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/netprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/netprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of netprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/netprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/netprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the netprofile resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netprofile": [ {
      "name":<String_value>,
      "srcip":<String_value>,
      "td":<Double_value>,
      "srcippersistency":<String_value>,
      "overridelsn":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/netprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/netprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netprofile": [ {
    "name":<String_value>,
    "srcip":<String_value>,
    "td":<Double_value>,
    "srcippersistency":<String_value>,
    "overridelsn":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netprofile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

netprofile_natrule_binding

May 20, 2016

Binding object showing the natrule that can be bound to netprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
rewriteip	<String>	Read-write	.
name	<String>	Read-write	Name of the netprofile to which to bind port ranges. Minimum length = 1
netmask	<String>	Read-write	.
natrule	<String>	Read-write	IPv4 network address on whose traffic you want the NetScaler appliance to do rewrite ip prefix.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/netprofile_natrule_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"netprofile_natrule_binding":{
    "name":<String_value>,
    "natrule":<String_value>,
    "netmask":<String_value>,
    "rewriteip":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile_natrule_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile_natrule_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/netprofile_natrule_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netprofile_natrule_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/netprofile_natrule_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the netprofile_natrule_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netprofile_natrule_binding": [ {
      "rewriteip":<String_value>,
      "name":<String_value>,
      "netmask":<String_value>,
      "natrule":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile_natrule_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"netprofile_natrule_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

netprofile_srcportset_binding

May 20, 2016

Binding object showing the srcportset that can be bound to netprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
srcportrange	<String>	Read-write	When the source port range is configured and associated with the netprofile bound to a service group, Netscaler will choose a port from the range configured for connection establishment at the backend servers. Minimum length = 1024 Maximum length = 65535
name	<String>	Read-write	Name of the netprofile to which to bind port ranges. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/netprofile_srcportset_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"netprofile_srcportset_binding":{
    "name":<String_value>,
    "srcportrange":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of netprofile_srcportset_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the netprofile_srcportset_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "netprofile_srcportset_binding": [ {
    "srcportrange":<String_value>,
    "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/netprofile_srcportset_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"netprofile_srcportset_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

onlinkipv6prefix

Jul 07, 2017

Configuration for on-link IPv6 global prefixes for Router Advertisment resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipv6prefix	<String>	Read-write	Onlink prefixes for RA messages.
onlinkprefix	<String>	Read-write	RA Prefix onlink flag. Default value: YES Possible values = YES, NO
autonomusprefix	<String>	Read-write	RA Prefix Autonomus flag. Default value: YES Possible values = YES, NO
depricateprefix	<String>	Read-write	Depricate the prefix. Default value: NO Possible values = YES, NO
decrementprefixlifetimes	<String>	Read-write	RA Prefix Autonomus flag. Default value: NO Possible values = YES, NO
prefixvalidelifetime	<Double>	Read-write	Valide life time of the prefix, in seconds. Default value: 2592000
prefixpreferredlifetime	<Double>	Read-write	Preferred life time of the prefix, in seconds. Default value: 604800
prefixcurrvalidelft	<Double>	Read-only	Prefix current valid life time.
prefixcurrpreferredlft	<Double>	Read-only	Prefix current prefered life time.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"onlinkipv6prefix":{
    "ipv6prefix":<String_value>,
    "onlinkprefix":<String_value>,
    "autonomusprefix":<String_value>,
    "depricateprefix":<String_value>,
    "decrementprefixlifetimes":<String_value>,
    "prefixvalidelifetime":<Double_value>,
    "prefixpreferredlifetime":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix/ipv6prefix_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"onlinkipv6prefix":{
    "ipv6prefix":<String_value>,
    "onlinkprefix":<String_value>,
    "autonomusprefix":<String_value>,
    "depricateprefix":<String_value>,
    "decrementprefixlifetimes":<String_value>,
    "prefixvalidelifetime":<Double_value>,
    "prefixpreferredlifetime":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"onlinkipv6prefix":{
    "ipv6prefix":<String_value>,
    "onlinkprefix":true,
    "autonomusprefix":true,
    "depricateprefix":true,
    "decrementprefixlifetimes":true,
    "prefixvalidelifetime":true,
    "prefixpreferredlifetime":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of onlinkipv6prefix resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix?**pagesize=#no&pageno=#no**

Use this query-parameter to get the onlinkipv6prefix resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "onlinkipv6prefix": [ {
      "ipv6prefix":<String_value>,
      "onlinkprefix":<String_value>,
      "autonomusprefix":<String_value>,
      "depricateprefix":<String_value>,
      "decrementprefixlifetimes":<String_value>,
      "prefixvalidelifetime":<Double_value>,
      "prefixpreferredlifetime":<Double_value>,
      "prefixcurrvalidelft":<Double_value>,
      "prefixcurrpreferredlft":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix/ipv6prefix_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix/ipv6prefix_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix/ipv6prefix_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "onlinkipv6prefix": [ {
    "ipv6prefix":<String_value>,
    "onlinkprefix":<String_value>,
    "autonomusprefix":<String_value>,
    "depricateprefix":<String_value>,
    "decrementprefixlifetimes":<String_value>,
    "prefixvalidelifetime":<Double_value>,
    "prefixpreferredlifetime":<Double_value>,
    "prefixcurrvalidelft":<Double_value>,
    "prefixcurrpreferredlft":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/onlinkipv6prefix?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "onlinkipv6prefix": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ptp

May 20, 2016

Configuration for Precision Time Protocol resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-write	Enables or disables Precision Time Protocol (PTP) on the appliance. If you disable PTP, make sure you enable Network Time Protocol (NTP) on the cluster. Default value: ENABLE Possible values = DISABLE, ENABLE

Operations (click to see Properties)

UPDATE | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/ptp

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ptp":{
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ptp

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ptp": [ {
    "state":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rnat

May 20, 2016

Configuration for RNAT configured route resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
network	<String>	Read-write	The network address defined for the RNAT entry. Minimum length = 1
netmask	<String>	Read-write	The subnet mask for the network address. Minimum length = 1
aclname	<String>	Read-write	An extended ACL defined for the RNAT entry. Minimum length = 1
redirectport	<Boolean>	Read-write	The port number to which the packets are redirected.
natip	<String>	Read-write	The NAT IP address defined for the RNAT entry. . Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
ownergroup	<String>	Read-write	The owner node group in a Cluster for this rnat rule. Default value: DEFAULT_NG Minimum length = 1
natip2	<String>	Read-write	The NAT IP(s) assigned to the RNAT. Minimum length = 1
srcippersistency	<String>	Read-write	Enables the NetScaler appliance to use the same NAT IP address for all RNAT sessions initiated from a particular server. Default value: DISABLED Possible values = ENABLED, DISABLED
useproxyport	<String>	Read-write	Enable source port proxying, which enables the NetScaler appliance to use the RNAT ips using proxied source port. Default value: ENABLED Possible values = ENABLED, DISABLED
			Synchronize connection information with the secondary appliance in a high

© 1999-2017 Citrix Systems, Inc. All rights reserved.

connfailover **Name**	<String> **Data Type**	Read-write **Permissions**	RNAT session. In order for this to work, tcpproxy should be DISABLED. To disable tcpproxy use "set rnatparam tcpproxy DISABLED". Default value: DISABLED Possible values = ENABLED, DISABLED **Description**
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

CLEAR | UPDATE | UNSET | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"rnat":{
    "network":<String_value>,
    "netmask":<String_value>,
    "aclname":<String_value>,
    "redirectport":<Boolean_value>,
    "natip":<String_value>,
    "td":<Double_value>,
    "ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rnat":{
    "network":<String_value>,
    "netmask":<String_value>,
    "natip":<String_value>,
    "td":<Double_value>,
    "aclname":<String_value>,
    "redirectport":<Boolean_value>,
    "natip2":<String_value>,
    "srcippersistency":<String_value>,
    "useproxyport":<String_value>,
    "ownergroup":<String_value>,
    "connfailover":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rnat":{
    "network":true,
    "netmask":true,
    "td":true,
    "aclname":true,
    "redirectport":true,
    "natip":true,
    "srcippersistency":true,
    "ownergroup":true,
    "useproxyport":true,
    "connfailover":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rnat?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/rnat?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rnat resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/rnat?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/rnat?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rnat resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rnat": [ {
      "network":<String_value>,
      "netmask":<String_value>,
      "td":<Double_value>,
      "natip":<String_value>,
      "aclname":<String_value>,
      "redirectport":<Boolean_value>,
      "srcippersistency":<String_value>,
      "useproxyport":<String_value>,
      "ownergroup":<String_value>,
      "connfailover":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "rnat": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rnat6

May 20, 2016

Configuration for IPv6 RNAT configured route resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the RNAT6 rule. Must begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the rule is created. Choose a name that helps identify the RNAT6 rule. Minimum length = 1
network	<String>	Read-write	IPv6 address of the network on whose traffic you want the NetScaler appliance to do RNAT processing. Minimum length = 1
acl6name	<String>	Read-write	Name of any configured ACL6 whose action is ALLOW. The rule of the ACL6 is used as an RNAT6 rule. Minimum length = 1
redirectport	<Integer>	Read-write	Port number to which the IPv6 packets are redirected. Applicable to TCP and UDP protocols. Minimum value = 1 Maximum value = 65535
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcippersistency	<String>	Read-write	Enable source ip persistency, which enables the NetScaler appliance to use the RNAT ips using source ip. Default value: DISABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | CLEAR | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rnat6":{
    "name":<String_value>,
    "network":<String_value>,
    "acl6name":<String_value>,
    "redirectport":<Integer_value>,
    "td":<Double_value>,
    "srcippersistency":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rnat6":{
    "name":<String_value>,
    "redirectport":<Integer_value>,
    "srcippersistency":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rnat6":{
    "name":<String_value>,
    "redirectport":true,
    "srcippersistency":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rnat6":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rnat6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/rnat6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rnat6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/rnat6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/rnat6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rnat6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rnat6": [ {
      "name":<String_value>,
      "network":<String_value>,
      "td":<Double_value>,
      "acl6name":<String_value>,
      "redirectport":<Integer_value>,
      "srcippersistency":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rnat6/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/rnat6/name_value<String>?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rnat6": [ {
    "name":<String_value>,
    "network":<String_value>,
    "td":<Double_value>,
    "acl6name":<String_value>,
    "redirectport":<Integer_value>,
    "srcippersistency":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{ "rnat6": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rnat6_binding

May 20, 2016

Binding object which returns the resources bound to rnat6.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the RNAT6 rule whose details you want to display. Minimum length = 1
rnat6_nsip6_binding	<rnat6_nsip6_binding[]>	Read-only	nsip6 that can be bound to rnat6.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "rnat6_binding": [ {
    "name":<String_value>,
    "rnat6_nsip6_binding":<rnat6_nsip6_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rnat6_nsip6_binding

Jul 07, 2017

Binding object showing the nsip6 that can be bound to rnat6.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the RNAT6 rule to which to bind NAT IPs. Minimum length = 1
natip6	<String>	Read-write	Nat IP Address. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/rnat6_nsip6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"rnat6_nsip6_binding":{
    "name":<String_value>,
    "natip6":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rnat6_nsip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rnat6_nsip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rnat6_nsip6_binding": [ {
    "name":<String_value>,
    "natip6":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rnat6_nsip6_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"rnat6_nsip6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rnatglobal_binding

May 20, 2016

Binding object which returns the resources bound to rnatglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
rnatglobal_auditsyslogpolicy_binding	<rnatglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to rnatglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/rnatglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "rnatglobal_binding": [ {
    "rnatglobal_auditsyslogpolicy_binding":<rnatglobal_auditsyslogpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rnatglobal_auditsyslogpolicy_binding

May 20, 2016

Binding object showing the auditsyslogpolicy that can be bound to rnatglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policy	<String>	Read-write	The policy Name.
all	<Boolean>	Read-write	Remove all RNAT global config.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"rnatglobal_auditsyslogpolicy_binding":{
    "policy":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rnatglobal_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rnatglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rnatglobal_auditsyslogpolicy_binding": [ {
    "priority":<Double_value>,
    "policy":<String_value>,
    "all":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rnatglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "rnatglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rnatparam

May 20, 2016

Configuration for RNAT parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
tcpproxy	<String>	Read-write	Enable TCP proxy, which enables the NetScaler appliance to optimize the RNAT TCP traffic by using Layer 4 features. Default value: ENABLED Possible values = ENABLED, DISABLED
srcippersistency	<String>	Read-write	Enable source ip persistency, which enables the NetScaler appliance to use the RNAT ips using source ip. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/rnatparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rnatparam":{
    "tcpproxy":<String_value>,
    "srcippersistency":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/rnatparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rnatparam":{
    "tcpproxy":true,
    "srcippersistency":true
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rnatparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rnatparam": [ {
      "tcpproxy":<String_value>,
      "srcippersistency":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

route

Jul 07, 2017

Configuration for route resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
network	<String>	Read-write	IPv4 network address for which to add a route entry in the routing table of the NetScaler appliance.
netmask	<String>	Read-write	The subnet mask associated with the network address.
gateway	<String>	Read-write	IP address of the gateway for this route. Can be either the IP address of the gateway, or can be null to specify a null interface route. Minimum length = 1
cost	<Double>	Read-write	Positive integer used by the routing algorithms to determine preference for using this route. The lower the cost, the higher the preference. Minimum value = 0 Maximum value = 65535
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
distance	<Double>	Read-write	Administrative distance of this route, which determines the preference of this route over other routes, with same destination, from different routing protocols. A lower value is preferred. Default value: 1 Minimum value = 0 Maximum value = 255
cost1	<Double>	Read-write	The cost of a route is used to compare routes of the same type. The route having the lowest cost is the most preferred route. Possible values: 0 through 65535. Default: 0. Minimum value = 0 Maximum value = 65535
weight	<Double>	Read-write	Positive integer used by the routing algorithms to determine preference for this route over others of equal cost. The lower the weight, the higher the preference. Default value: 1 Minimum value = 1 Maximum value = 65535
advertise	<String>	Read-write	Advertise this route. Possible values = DISABLED, ENABLED
protocol	<String[]>	Read-write	Routing protocol used for advertising this route. Default value: ADV_ROUTE_FLAGS Possible values = OSPF, ISIS, RIP, BGP
msr	<String>	Read-write	Monitor this route using a monitor of type ARP or PING. Default value: DISABLED Possible values = ENABLED, DISABLED
monitor	<String>	Read-write	Name of the monitor, of type ARP or PING, configured on the NetScaler appliance to monitor this route. Minimum length = 1
ownergroup	<String>	Read-write	The owner node group in a Cluster for this route. If owner node group is not specified then the route is treated as Striped route. Default value: DEFAULT_NG Minimum length = 1
routetype	<String>	Read-write	Protocol used by routes that you want to remove from the routing table of the NetScaler appliance. Possible values = CONNECTED, STATIC, DYNAMIC, OSPF, ISIS, RIP, BGP, VPATH
detail	<Boolean>	Read-write	Display a detailed view.
gatewayname	<String>	Read-only	The name of the gateway for this route. For a route other than a link load balancing (LLB) route, this value is null.
type	<Boolean>	Read-only	State of the RNAT.
dynamic	<Boolean>	Read-only	State of the route.
Static	<Boolean>	Read-only	.
permanent	<Boolean>	Read-only	.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
lbroute	<Boolean>	Read-only	.
adv	<Boolean>	Read-only	.
tunnel	<Boolean>	Read-only	Show whether it is a tunnel route or not.
data	<Boolean>	Read-only	Internal data of this route.
data0	<Boolean>	Read-only	Internal route type is stored, used for get.
flags	<Boolean>	Read-only	If this route is dynamic, the name of the routing protocol from which it was learned.
routeowners	<String[]>	Read-only	Use this option with -dynamic and in a cluster only to specify the set of nodes from which this dynamic route has been learnt. Possible values = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31
retain	<Integer>	Read-only	.
ospf	<Boolean>	Read-only	OSPF protocol.
isis	<Boolean>	Read-only	ISIS protocol.
rip	<Boolean>	Read-only	RIP protocol.
bgp	<Boolean>	Read-only	BGP protocol.
dhcp	<Boolean>	Read-only	.
advospf	<Boolean>	Read-only	Advertised through OSPF protocol.
advisis	<Boolean>	Read-only	Advertised through ISIS protocol.
advrip	<Boolean>	Read-only	Advertised through RIP protocol.
advbgp	<Boolean>	Read-only	Advertised through BGP protocol.
state	<Double>	Read-only	The state of the static route. Possible values: UP, DOWN.
totalprobes	<Double>	Read-only	The total number of probes sent.
totalfailedprobes	<Double>	Read-only	The total number of failed probes.
failedprobes	<Double>	Read-only	Number of the current failed monitoring probes.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter used with the message code.
monstatparam2	<Integer>	Read-only	Second parameter used with the message code.
monstatparam3	<Integer>	Read-only	Third parameter used with the message code.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | CLEAR | DELETE | UPDATE | UNSET | GET (ALL) | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/route

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"route":{
    "network":<String_value>,
    "netmask":<String_value>,
    "gateway":<String_value>,
    "cost":<Double_value>,
    "td":<Double_value>,
    "distance":<Double_value>,
    "cost1":<Double_value>,
    "weight":<Double_value>,
    "advertise":<String_value>,
    "protocol":<String[]_value>,
    "msr":<String_value>,
    "monitor":<String_value>,
    "ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/route?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"route":{
    "routetype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/route/network_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/route

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"route":{
    "network":<String_value>,
    "netmask":<String_value>,
    "gateway":<String_value>,
    "td":<Double_value>,
    "distance":<Double_value>,
    "cost1":<Double_value>,
    "weight":<Double_value>,
    "advertise":<String_value>,
    "protocol":<String[]_value>,
    "msr":<String_value>,
    "monitor":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/route?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"route":{
    "network":<String_value>,
    "netmask":<String_value>,
    "gateway":<String_value>,
    "td":<Double_value>,
    "advertise":true,
    "distance":true,
    "cost1":true,
    "weight":true,
    "protocol":true,
    "msr":true,
    "monitor":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/route

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

args

http://<netscaler-ip-address>/nitro/v1/config/route?**args=network:<String_value>,netmask:<String_value>,gateway:<String_value>,td:<Double_value>,routetype:<String_value>,detail:<Boolean_value**

Use this query-parameter to get route resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/route?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/route?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of route resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/route?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/route?**pagesize=#no&pageno=#no**

Use this query-parameter to get the route resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "route": [ {
network:<String_value>,netmask:<String_value>,gateway:<String_value>,td:<Double_value>,routetype:<String_value>,detail:<Boolean_value>      "gatewayname":<String_value>,
      "advertise":<String_value>,
      "type":<Boolean_value>,
      "dynamic":<Boolean_value>,
      "Static":<Boolean_value>,
      "permanent":<Boolean_value>,
      "direct":<Boolean_value>,
      "nat":<Boolean_value>,
      "lbroute":<Boolean_value>,
      "adv":<Boolean_value>,
      "tunnel":<Boolean_value>,
      "cost":<Double_value>,
      "distance":<Double_value>,
      "cost1":<Double_value>,
      "weight":<Double_value>,
      "protocol":<String[]_value>,
      "data":<Boolean_value>,
      "data0":<Boolean_value>,
      "flags":<Boolean_value>,
      "routeowners":<String[]_value>,
      "retain":<Integer_value>,
      "ospf":<Boolean_value>,
      "isis":<Boolean_value>,
      "rip":<Boolean_value>,
      "bgp":<Boolean_value>,
      "dhcp":<Boolean_value>,
      "advospf":<Boolean_value>,
      "advisis":<Boolean_value>,
      "advrip":<Boolean_value>,
      "advbgp":<Boolean_value>,
      "msr":<String_value>,
      "monitor":<String_value>,
      "state":<Double_value>,
      "totalprobes":<Double_value>,
      "totalfailedprobes":<Double_value>,
      "failedprobes":<Double_value>,
      "monstatcode":<Integer_value>,
      "monstatparam1":<Integer_value>,
      "monstatparam2":<Integer_value>,
      "monstatparam3":<Integer_value>,
      "ownergroup":<String_value>
  }]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/route?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "route": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

route6

May 20, 2016

Configuration for route 6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
network	<String>	Read-write	IPv6 network address for which to add a route entry to the routing table of the NetScaler appliance.
gateway	<String>	Read-write	The gateway for this route. The value for this parameter is either an IPv6 address or null. Default value: 0
vlan	<Double>	Read-write	Integer value that uniquely identifies a VLAN through which the NetScaler appliance forwards the packets for this route. Default value: 0 Minimum value = 0 Maximum value = 4094
vxlan	<Double>	Read-write	Integer value that uniquely identifies a VXLAN through which the NetScaler appliance forwards the packets for this route. Minimum value = 1 Maximum value = 16777215
weight	<Double>	Read-write	Positive integer used by the routing algorithms to determine preference for this route over others of equal cost. The lower the weight, the higher the preference. Default value: 1 Minimum value = 1 Maximum value = 65535
distance	<Double>	Read-write	Administrative distance of this route from the appliance. Default value: 1 Minimum value = 1 Maximum value = 254
cost	<Double>	Read-write	Positive integer used by the routing algorithms to determine preference for this route. The lower the cost, the higher the preference. Default value: 1 Minimum value = 0 Maximum value = 65535
advertise	<String>	Read-write	Advertise this route. Possible values = DISABLED, ENABLED
msr	<String>	Read-write	Monitor this route with a monitor of type ND6 or PING. Default value: DISABLED Possible values = ENABLED, DISABLED
monitor	<String>	Read-write	Name of the monitor, of type ND6 or PING, configured on the NetScaler appliance to monitor this route. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
ownergroup	<String>	Read-write	The owner node group in a Cluster for this route6. If owner node group is not specified then the route is treated as Striped route. Default value: DEFAULT_NG Minimum length = 1
routetype	<String>	Read-write	Type of IPv6 routes to remove from the routing table of the NetScaler appliance. Possible values = CONNECTED, STATIC, DYNAMIC, OSPF, ISIS, BGP, RIP, ND-RA-ROUTE, FIB6
detail	<Boolean>	Read-write	To get a detailed view.
gatewayname	<String>	Read-only	The name of the gateway for this route.
type	<Boolean>	Read-only	State of the RNAT.
dynamic	<Boolean>	Read-only	Whether this route is dynamically learned or not.
data	<Boolean>	Read-only	Internal data of this route.
flags	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
state	<Double>	Read-only	Whether this route is UP or DOWN.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
totalfailedprobes	<Double>	Read-only	The total number of failed probes.
failedprobes	<Double>	Read-only	Current number of failed monitoring probes.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
data1	<String>	Read-only	Internal data of this route. Possible values = ENABLED, DISABLED
routeowners	<String[]>	Read-only	Use this option with -dynamic and in a cluster only to specify the set of nodes from which this dynamic route has been learnt. Possible values = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31
retain	<Integer>	Read-only	.
Static	<Boolean>	Read-only	Static route.
permanent	<Boolean>	Read-only	Permanent Route.
connected	<Boolean>	Read-only	Connected Route.
ospfv3	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
isis	<Boolean>	Read-only	If this route is dynamic then which routing protocol was it learnt from.
active	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
bgp	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
rip	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
raroute	<Boolean>	Read-only	For a dynamic route, the routing protocol from which the route was learned.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | CLEAR | DELETE | UPDATE | UNSET | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/route6

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"route6":{
    "network":<String_value>,
    "gateway":<String_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "weight":<Double_value>,
    "distance":<Double_value>,
    "cost":<Double_value>,
    "advertise":<String_value>,
    "msr":<String_value>,
    "monitor":<String_value>,
    "td":<Double_value>,
    "ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/route6?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"route6":{
    "routetype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/route6/network_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/route6

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"route6":{
    "network":<String_value>,
    "gateway":<String_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "weight":<Double_value>,
    "distance":<Double_value>,
    "cost":<Double_value>,
    "advertise":<String_value>,
    "msr":<String_value>,
    "monitor":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/route6?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"route6":{
    "network":<String_value>,
    "gateway":<String_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "td":<Double_value>,
    "weight":true,
    "distance":true,
    "cost":true,
    "advertise":true,
    "msr":true,
    "monitor":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/route6

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/route6?**args=network:<String_value>,gateway:<String_value>,vlan:<Double_value>,vxlan:<Double_value>,td:<Double_value>,routetype:<String_value>,de**

Use this query-parameter to get route6 resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/route6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/route6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of route6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/route6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/route6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the route6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "route6": [ {
network:<String_value>,gateway:<String_value>,vlan:<Double_value>,vxlan:<Double_value>,td:<Double_value>,routetype:<String_value>,detail:<Boolean_value>      "gatewayname":<String_value>,
    "advertise":<String_value>,
    "type":<Boolean_value>,
    "dynamic":<Boolean_value>,
    "weight":<Double_value>,
    "distance":<Double_value>,
    "cost":<Double_value>,
    "data":<Boolean_value>,
    "flags":<Boolean_value>,
    "msr":<String_value>,
    "monitor":<String_value>,
    "state":<Double_value>,
    "totalprobes":<Double_value>,
    "totalfailedprobes":<Double_value>,
    "failedprobes":<Double_value>,
    "monstatcode":<Integer_value>,
    "monstatparam1":<Integer_value>,
    "monstatparam2":<Integer_value>,
    "monstatparam3":<Integer_value>,
    "data1":<String_value>,
    "routeowners":<String[]_value>,
    "retain":<Integer_value>,
    "Static":<Boolean_value>,
    "permanent":<Boolean_value>,
    "connected":<Boolean_value>,
    "ospfv3":<Boolean_value>,
    "isis":<Boolean_value>,
    "active":<Boolean_value>,
    "bgp":<Boolean_value>,
    "rip":<Boolean_value>,
    "raroute":<Boolean_value>,
    "ownergroup":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/route6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "route6": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rsskeytype

May 20, 2016

Configuration for RSS key type resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
rsstype	<String>	Read-write	Type of RSS key, possible values are SYMMETRIC and SYMMETRIC. Default value: SYMMETRIC Possible values = ASYMMETRIC, SYMMETRIC

Operations (click to see Properties)

UPDATE | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/rsskeytype

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rsskeytype":{
    "rsstype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rsskeytype

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rsskeytype": [ {
    "rsstype":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vlan

May 20, 2016

Configuration for "VLAN" resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	A positive integer that uniquely identifies a VLAN. Minimum value = 1 Maximum value = 4094
aliasname	<String>	Read-write	A name for the VLAN. Must begin with a letter, a number, or the underscore symbol, and can consist of from 1 to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the VLAN. However, you cannot perform any VLAN operation by specifying this name instead of the VLAN ID. Maximum length = 31
dynamicrouting	<String>	Read-write	Enable dynamic routing on this VLAN. Default value: DISABLED Possible values = ENABLED, DISABLED
ipv6dynamicrouting	<String>	Read-write	Enable all IPv6 dynamic routing protocols on this VLAN. Note: For the ENABLED setting to work, you must configure IPv6 dynamic routing protocols from the VTYSH command line. Default value: DISABLED Possible values = ENABLED, DISABLED
mtu	<Double>	Read-write	Specifies the maximum transmission unit (MTU), in bytes. The MTU is the largest packet size, excluding 14 bytes of ethernet header and 4 bytes of crc, that can be transmitted and received over this VLAN. Default value: 0 Minimum value = 500 Maximum value = 9216
linklocalipv6addr	<String>	Read-only	The link-local IP address assigned to the VLAN.
rnat	<Boolean>	Read-only	Temporary flag used for internal purpose.
portbitmap	<Double>	Read-only	Member interfaces of this vlan.
lsbitmap	<Double>	Read-only	Member linksets of this vlan.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
tagbitmap	<Double>	Read-only	Tagged members of this vlan.
lstagbitmap	<Double>	Read-only	Tagged linksets of this vlan.
ifaces	<String>	Read-only	Names of all member interfaces of this vlan.
tagifaces	<String>	Read-only	Names of all tagged member interfaces of this vlan.
ifnum	<String>	Read-only	The interface to be bound to the VLAN, specified in slot/port notation (for example, 1/3). Minimum length = 1
tagged	<Boolean>	Read-only	Make the interface an 802.1q tagged interface. Packets sent on this interface on this VLAN have an additional 4-byte 802.1q tag, which identifies the VLAN. To use 802.1q tagging, you must also configure the switch connected to the appliance's interfaces.
vlantd	<Double>	Read-only	Traffic domain associated with vlan. Minimum value = 0 Maximum value = 4094
sdxvlan	<String>	Read-only	SDX vlan. Default value: NO Possible values = YES, NO
partitionname	<String>	Read-only	Name of the Partition to which this vlan bound to. Minimum length = 1
vxlan	<Double>	Read-only	The VXLAN that extends this vlan.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vlan":{
    "id":<Double_value>,
    "aliasname":<String_value>,
    "dynamicrouting":<String_value>,
    "ipv6dynamicrouting":<String_value>,
    "mtu":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan/id_value<Double>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vlan":{
    "id":<Double_value>,
    "aliasname":<String_value>,
    "dynamicrouting":<String_value>,
    "ipv6dynamicrouting":<String_value>,
    "mtu":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"vlan":{
    "id":<Double_value>,
    "aliasname":true,
    "dynamicrouting":true,
    "ipv6dynamicrouting":true,
    "mtu":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vlan?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vlan?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vlan resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vlan?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vlan?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vlan resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vlan": [ {
    "id":<Double_value>,
    "aliasname":<String_value>,
    "linklocalipv6addr":<String_value>,
    "rnat":<Boolean_value>,
    "portbitmap":<Double_value>,
    "lsbitmap":<Double_value>,
    "tagbitmap":<Double_value>,
    "lstagbitmap":<Double_value>,
    "ifaces":<String_value>,
    "tagifaces":<String_value>,
    "dynamicrouting":<String_value>,
    "ipv6dynamicrouting":<String_value>,
    "ifnum":<String_value>,
    "tagged":<Boolean_value>,
    "vlantd":<Double_value>,
    "sdxvlan":<String_value>,
    "mtu":<Double_value>,
    "partitionname":<String_value>,
    "vxlan":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan/id_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vlan/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vlan/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vlan": [ {
    "id":<Double_value>,
    "aliasname":<String_value>,
    "linklocalipv6addr":<String_value>,
    "rnat":<Boolean_value>,
    "portbitmap":<Double_value>,
    "lsbitmap":<Double_value>,
    "tagbitmap":<Double_value>,
    "lstagbitmap":<Double_value>,
    "ifaces":<String_value>,
    "tagifaces":<String_value>,
    "dynamicrouting":<String_value>,
    "ipv6dynamicrouting":<String_value>,
    "ifnum":<String_value>,
    "tagged":<Boolean_value>,
    "vlantd":<Double_value>,
    "sdxvlan":<String_value>,
    "mtu":<Double_value>,
    "partitionname":<String_value>,
    "vxlan":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vlan": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vlan_binding

May 20, 2016

Binding object which returns the resources bound to vlan.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Integer that uniquely identifies the VLAN for which the details are to be displayed. Minimum value = 1 Maximum value = 4094
vlan_channel_binding	<vlan_channel_binding[]>	Read-only	channel that can be bound to vlan.
vlan_nsip_binding	<vlan_nsip_binding[]>	Read-only	nsip that can be bound to vlan.
vlan_interface_binding	<vlan_interface_binding[]>	Read-only	interface that can be bound to vlan.
vlan_linkset_binding	<vlan_linkset_binding[]>	Read-only	linkset that can be bound to vlan.
vlan_nsip6_binding	<vlan_nsip6_binding[]>	Read-only	nsip6 that can be bound to vlan.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_binding/id_value<Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "vlan_binding": [ {
    "id":<Double_value>,
    "vlan_interface_binding":<vlan_interface_binding[]_value>,
    "vlan_nsip6_binding":<vlan_nsip6_binding[]_value>,
    "vlan_nsip_binding":<vlan_nsip_binding[]_value>,
    "vlan_linkset_binding":<vlan_linkset_binding[]_value>,
    "vlan_channel_binding":<vlan_channel_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vlan_channel_binding

May 20, 2016

Binding object showing the channel that can be bound to vlan.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1
id	<Double>	Read-write	Specifies the virtual LAN ID. Minimum value = 1 Maximum value = 4094
ifnum	<String>	Read-write	The interface to be bound to the VLAN, specified in slot/port notation (for example, 1/3). Minimum length = 1
tagged	<Boolean>	Read-write	Make the interface an 802.1q tagged interface. Packets sent on this interface on this VLAN have an additional 4-byte 802.1q tag, which identifies the VLAN. To use 802.1q tagging, you must also configure the switch connected to the appliance's interfaces.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vlan_channel_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vlan_channel_binding":{
    "id":<Double_value>,
    "ifnum":<String_value>,
    "tagged":<Boolean_value>,
    "ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_channel_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_channel_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vlan_channel_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vlan_channel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vlan_channel_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vlan_channel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vlan_channel_binding": [ {
      "ownergroup":<String_value>,
      "id":<Double_value>,
      "ifnum":<String_value>,
      "tagged":<Boolean_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_channel_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vlan_channel_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vlan_interface_binding

May 20, 2016

Binding object showing the interface that can be bound to vlan.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1
id	<Double>	Read-write	Specifies the virtual LAN ID. Minimum value = 1 Maximum value = 4094
ifnum	<String>	Read-write	The interface to be bound to the VLAN, specified in slot/port notation (for example, 1/3). Minimum length = 1
tagged	<Boolean>	Read-write	Make the interface an 802.1q tagged interface. Packets sent on this interface on this VLAN have an additional 4-byte 802.1q tag, which identifies the VLAN. To use 802.1q tagging, you must also configure the switch connected to the appliance's interfaces.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vlan_interface_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vlan_interface_binding":{
    "id":<Double_value>,
    "ifnum":<String_value>,
    "tagged":<Boolean_value>,
    "ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_interface_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_interface_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vlan_interface_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vlan_interface_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vlan_interface_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vlan_interface_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vlan_interface_binding": [ {
      "ownergroup":<String_value>,
      "id":<Double_value>,
      "ifnum":<String_value>,
      "tagged":<Boolean_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_interface_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vlan_interface_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vlan_linkset_binding

Jul 07, 2017

Binding object showing the linkset that can be bound to vlan.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1
id	<Double>	Read-write	Specifies the virtual LAN ID. Minimum value = 1 Maximum value = 4094
ifnum	<String>	Read-write	The interface to be bound to the VLAN, specified in slot/port notation (for example, 1/3). Minimum length = 1
tagged	<Boolean>	Read-write	Make the interface an 802.1q tagged interface. Packets sent on this interface on this VLAN have an additional 4-byte 802.1q tag, which identifies the VLAN. To use 802.1q tagging, you must also configure the switch connected to the appliance's interfaces.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vlan_linkset_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vlan_linkset_binding":{
    "id":<Double_value>,
    "ifnum":<String_value>,
    "tagged":<Boolean_value>,
    "ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_linkset_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_linkset_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vlan_linkset_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vlan_linkset_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vlan_linkset_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vlan_linkset_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vlan_linkset_binding": [ {
      "ownergroup":<String_value>,
      "id":<Double_value>,
      "ifnum":<String_value>,
      "tagged":<Boolean_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_linkset_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vlan_linkset_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vlan_nsip6_binding

May 20, 2016

Binding object showing the nsip6 that can be bound to vlan.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1
id	<Double>	Read-write	Specifies the virtual LAN ID. Minimum value = 1 Maximum value = 4094
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
netmask	<String>	Read-write	Subnet mask for the network address defined for this VLAN. Minimum length = 1
ipaddress	<String>	Read-write	The IP address assigned to the VLAN.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vlan_nsip6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vlan_nsip6_binding":{
    "id":<Double_value>,
    "ipaddress":<String_value>,
    "netmask":<String_value>,
    "td":<Double_value>,
    "ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vlan_nsip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vlan_nsip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "vlan_nsip6_binding": [ {
    "ownergroup":<String_value>,
    "id":<Double_value>,
    "td":<Double_value>,
    "netmask":<String_value>,
    "ipaddress":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_nsip6_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vlan_nsip6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vlan_nsip_binding

May 20, 2016

Binding object showing the nsip that can be bound to vlan.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ownergroup	<String>	Read-write	The owner node group in a Cluster for this vlan. Default value: DEFAULT_NG Minimum length = 1
id	<Double>	Read-write	Specifies the virtual LAN ID. Minimum value = 1 Maximum value = 4094
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
netmask	<String>	Read-write	Subnet mask for the network address defined for this VLAN.
ipaddress	<String>	Read-write	The IP address assigned to the VLAN.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vlan_nsip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vlan_nsip_binding":{
    "id":<Double_value>,
    "ipaddress":<String_value>,
    "netmask":<String_value>,
    "td":<Double_value>,
    "ownergroup":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_nsip_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_nsip_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vlan_nsip_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vlan_nsip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vlan_nsip_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vlan_nsip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vlan_nsip_binding": [ {
    "ownergroup":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "id":<Double_value>,
    "td":<Double_value>,
    "netmask":<String_value>,
    "ipaddress":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vlan_nsip_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vlan_nsip_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpath

Jul 07, 2017

Configuration for Vpath resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the vPath. Must begin with a letter, number, or the underscore character (_), and can consist of letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore characters. Cannot be changed after the profile is created. Choose a name that helps identify the net profile. Minimum length = 1
destip	<String>	Read-write	This is the destination ip, where vPath encapsulated packets needs to be sent. Minimum length = 1
netmask	<String>	Read-write	Subnet mask associated with the destination network.
gateway	<String>	Read-write	Next hop gateway to reach the destination address.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpath

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpath":{
    "name":<String_value>,
    "destip":<String_value>,
    "netmask":<String_value>,
    "gateway":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpath/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpath

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpath?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpath?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpath resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpath?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpath?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpath resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "vpath": [ {
      "name":<String_value>,
      "destip":<String_value>,
      "netmask":<String_value>,
      "gateway":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpath/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpath/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpath/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpath": [ {
      "name":<String_value>,
      "destip":<String_value>,
      "netmask":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "gateway":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpath?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpath": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpathparam

May 20, 2016

Configuration for VpathParam resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
srcip	<String>	Read-write	source-IP address used for all vPath L3 encapsulations. Must be a MIP or SNIP address. Minimum length = 1
offload	<String>	Read-write	enable/disable vPath offload feature. Default value: DISABLED Possible values = ENABLED, DISABLED
encapsulation	<String>	Read-only	Global vPath encapsulation . Possible values = ENABLED, DISABLED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpathparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpathparam":{
    "srcip":<String_value>,
    "offload":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpathparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpathparam":{
    "srcip":true,
    "offload":true
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpathparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpathparam": [ {
      "srcip":<String_value>,
      "encapsulation":<String_value>,
      "offload":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vrid

May 20, 2016

Configuration for Virtual Router ID resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Integer that uniquely identifies the VMAC address. The generic VMAC address is in the form of 00:00:5e:00:01:<VRID>. For example, if you add a VRID with a value of 60 and bind it to an interface, the resulting VMAC address is 00:00:5e:00:01:3c, where 3c is the hexadecimal representation of 60. Minimum value = 1 Maximum value = 255
priority	<Double>	Read-write	Base priority (BP), in an active-active mode configuration, which ordinarily determines the master VIP address. Default value: 255 Minimum value = 0 Maximum value = 255
preemption	<String>	Read-write	In an active-active mode configuration, make a backup VIP address the master if its priority becomes higher than that of a master VIP address bound to this VMAC address. If you disable pre-emption while a backup VIP address is the master, the backup VIP address remains master until the original master VIP's priority becomes higher than that of the current master. Default value: ENABLED Possible values = ENABLED, DISABLED
sharing	<String>	Read-write	In an active-active mode configuration, enable the backup VIP address to process any traffic instead of dropping it. Default value: DISABLED Possible values = ENABLED, DISABLED
tracking	<String>	Read-write	The effective priority (EP) value, relative to the base priority (BP) value in an active-active mode configuration. When EP is set to a value other than None, it is EP, not BP, which determines the master VIP address. Available settings function as follows: * NONE - No tracking. EP = BP * ALL - If the status of all virtual servers is UP, EP = BP. Otherwise, EP = 0. * ONE - If the status of at least one virtual server is UP, EP = BP. Otherwise, EP = 0. * PROGRESSIVE - If the status of all virtual servers is UP, EP = BP. If the status of all virtual servers is DOWN, EP = 0. Otherwise EP = BP (1 - K/N), where N is the total number of virtual servers associated with the VIP address and K is the number of virtual servers for which the status is DOWN.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: NONE Possible values = NONE, ONE, ALL, PROGRESSIVE
ownernode	<Double>	Read-write	Assign a cluster node as the owner of this VMAC address. If no owner is configured, owner node is displayed as ALL and one node is dynamically elected as the owner. Minimum value = 0 Maximum value = 31
trackifnumpriority	<Double>	Read-write	Priority by which the Effective priority will be reduced if any of the tracked interfaces goes down. Default value: 0 Minimum value = 1 Maximum value = 255
preemptiondelaytimer	<Double>	Read-write	Preemption delay time in seconds. If any high priority node will come in network, it will wait for these many seconds before becoming master. Default value: 0 Minimum value = 1 Maximum value = 36000
all	<Boolean>	Read-write	Remove all the configured VMAC addresses from the NetScaler appliance.
ifaces	<String>	Read-only	Interfaces bound to this VRID.
type	<String>	Read-only	Indicates whether this VRID entry was added manually or dynamically. When you manually add a VRID entry, the value for this parameter is STATIC. Otherwise, it is DYNAMIC. Possible values = STATIC, DYNAMIC
effectivepriority	<Double>	Read-only	The effective priority of this VRID.
flags	<Double>	Read-only	Flags.
ipaddress	<String>	Read-only	The IP address bound to the VRID.
state	<Double>	Read-only	State of this VRID.
operationalownernode	<Double>	Read-only	Run time owner node of the vrid. Minimum value = 0 Maximum value = 31
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vrid":{
    "id":<Double_value>,
    "priority":<Double_value>,
    "preemption":<String_value>,
    "sharing":<String_value>,
    "tracking":<String_value>,
    "ownernode":<Double_value>,
    "trackifnumpriority":<Double_value>,
    "preemptiondelaytimer":<Double_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vrid":{
    "id":<Double_value>,
    "priority":<Double_value>,
    "preemption":<String_value>,
    "sharing":<String_value>,
    "tracking":<String_value>,
    "ownernode":<Double_value>,
    "trackifnumpriority":<Double_value>,
    "preemptiondelaytimer":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vrid":{
    "id":<Double_value>,
    "priority":true,
    "preemption":true,
    "sharing":true,
    "tracking":true,
    "ownernode":true,
    "trackifnumpriority":true,
    "preemptiondelaytimer":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vrid?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vrid?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vrid resources configured on NetScaler.Filtering can be done

© 1999-2017 Citrix Systems, Inc. All rights reserved.

on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vrid?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vrid?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vrid resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid": [ {
      "id":<Double_value>,
      "ifaces":<String_value>,
      "type":<String_value>,
      "priority":<Double_value>,
      "effectivepriority":<Double_value>,
      "preemption":<String_value>,
      "sharing":<String_value>,
      "tracking":<String_value>,
      "flags":<Double_value>,
      "ipaddress":<String_value>,
      "state":<Double_value>,
      "operationalownernode":<Double_value>,
      "ownernode":<Double_value>,
      "trackifnumpriority":<Double_value>,
      "preemptiondelaytimer":<Double_value>
}]}
```

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid/id_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vrid/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vrid/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid": [ {
      "id":<Double_value>,
      "ifaces":<String_value>,
      "type":<String_value>,
      "priority":<Double_value>,
      "effectivepriority":<Double_value>,
      "preemption":<String_value>,
      "sharing":<String_value>,
      "tracking":<String_value>,
      "flags":<Double_value>,
      "ipaddress":<String_value>,
      "state":<Double_value>,
      "operationalownernode":<Double_value>,
      "ownernode":<Double_value>,
      "trackifnumpriority":<Double_value>,
      "preemptiondelaytimer":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vrid6

May 20, 2016

Configuration for Virtual Router ID for IPv6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
all	<Boolean>	Read-write	Remove all configured VMAC6 addresses from the NetScaler appliance.
ifaces	<String>	Read-only	Interfaces bound to this VRID.
ifnum	<String>	Read-only	Interfaces bound to this VRID.
type	<String>	Read-only	Type (static or dynamic) of this VRID. Possible values = STATIC, DYNAMIC
priority	<Double>	Read-only	The priority of this VRID.
state	<Double>	Read-only	State of this VRID.
flags	<Double>	Read-only	Flags.
ipaddress	<String>	Read-only	The IP address bound to the VRID6.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vrid6":{
    "id":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6/id_value<Double>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vrid6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vrid6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vrid6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vrid6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vrid6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vrid6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid6": [ {
      "id":<Double_value>,
      "ifaces":<String_value>,
      "ifnum":<String_value>,
      "type":<String_value>,
      "priority":<Double_value>,
      "state":<Double_value>,
      "flags":<Double_value>,
      "ipaddress":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6/id_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vrid6/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vrid6/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid6": [ {
      "id":<Double_value>,
      "ifaces":<String_value>,
      "ifnum":<String_value>,
      "type":<String_value>,
      "priority":<Double_value>,
      "state":<Double_value>,
      "flags":<Double_value>,
      "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid6": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vrid6_binding

Jul 07, 2017

Binding object which returns the resources bound to vrid6.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
vrid6_interface_binding	<vrid6_interface_binding[]>	Read-only	interface that can be bound to vrid6.
vrid6_nsip_binding	<vrid6_nsip_binding[]>	Read-only	nsip that can be bound to vrid6.
vrid6_nsip6_binding	<vrid6_nsip6_binding[]>	Read-only	nsip6 that can be bound to vrid6.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6_binding/id_value<Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "vrid6_binding": [ {
    "id":<Double_value>,
    "vrid6_nsip6_binding":<vrid6_nsip6_binding[]_value>,
    "vrid6_nsip_binding":<vrid6_nsip_binding[]_value>,
    "vrid6_interface_binding":<vrid6_interface_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vrid6_interface_binding

Jul 07, 2017

Binding object showing the interface that can be bound to vrid6.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
ifnum	<String>	Read-write	Interfaces bound to this VRID.
flags	<Double>	Read-only	Flags.
vlan	<Double>	Read-only	The VLAN in which this VRID resides.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vrid6_interface_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vrid6_interface_binding":{
    "id":<Double_value>,
    "ifnum":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6_interface_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6_interface_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vrid6_interface_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid6_interface_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vrid6_interface_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vrid6_interface_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid6_interface_binding": [ {
      "id":<Double_value>,
      "ifnum":<String_value>,
      "flags":<Double_value>,
      "vlan":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6_interface_binding/id_value<Double>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vrid6_interface_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vrid6_nsip6_binding

May 20, 2016

Binding object showing the nsip6 that can be bound to vrid6.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
ifnum	<String>	Read-write	Interfaces to bind tothe VMAC6, specified in (slot/port) notation (for example, 1/2).Use spaces to separate multiple entries.
ipaddress	<String>	Read-write	The IP address bound to the VRID6.
flags	<Double>	Read-only	Flags.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6_nsip6_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vrid6_nsip6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid6_nsip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vrid6_nsip6_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vrid6_nsip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid6_nsip6_binding": [ {
    "id":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ifnum":<String_value>,
    "ipaddress":<String_value>,
    "flags":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6_nsip6_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vrid6_nsip6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vrid6_nsip_binding

May 20, 2016

Binding object showing the nsip that can be bound to vrid6.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Integer value that uniquely identifies a VMAC6 address. Minimum value = 1 Maximum value = 255
ifnum	<String>	Read-write	Interfaces to bind tothe VMAC6, specified in (slot/port) notation (for example, 1/2).Use spaces to separate multiple entries.
ipaddress	<String>	Read-write	The IP address bound to the VRID6.
flags	<Double>	Read-only	Flags.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6_nsip_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vrid6_nsip_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid6_nsip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vrid6_nsip_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vrid6_nsip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid6_nsip_binding": [ {
    "id":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ifnum":<String_value>,
    "ipaddress":<String_value>,
    "flags":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid6_nsip_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vrid6_nsip_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vridparam

May 20, 2016

Configuration for VR ID parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sendtomaster	<String>	Read-write	Forward packets to the master node, in an active-active mode configuration, if the virtual server is in the backup state and sharing is disabled. Default value: DISABLED Possible values = ENABLED, DISABLED
hellointerval	<Double>	Read-write	Interval, in milliseconds, between vrrp advertisement messages sent to the peer node in active-active mode. Default value: 1000 Minimum value = 200 Maximum value = 1000
deadinterval	<Double>	Read-write	Number of seconds after which a peer node in active-active mode is marked down if vrrp advertisements are not received from the peer node. Default value: 3 Minimum value = 1 Maximum value = 3

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vridparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vridparam":{
    "sendtomaster":<String_value>,
    "hellointerval":<Double_value>,
    "deadinterval":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vridparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vridparam":{
    "sendtomaster":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "hellointerval":true,
    "deadinterval":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vridparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vridparam": [ {
    "sendtomaster":<String_value>,
    "hellointerval":<Double_value>,
    "deadinterval":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vrid_binding

Jul 07, 2017

Binding object which returns the resources bound to vrid.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Integer value that uniquely identifies the VMAC address. Minimum value = 1 Maximum value = 255
vrid_nsip6_binding	<vrid_nsip6_binding[]>	Read-only	nsip6 that can be bound to vrid.
vrid_interface_binding	<vrid_interface_binding[]>	Read-only	interface that can be bound to vrid.
vrid_nsip_binding	<vrid_nsip_binding[]>	Read-only	nsip that can be bound to vrid.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid_binding/id_value<Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "vrid_binding": [ {
    "id":<Double_value>,
    "vrid_nsip6_binding":<vrid_nsip6_binding[]_value>,
    "vrid_nsip_binding":<vrid_nsip_binding[]_value>,
    "vrid_interface_binding":<vrid_interface_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vrid_interface_binding

Jul 07, 2017

Binding object showing the interface that can be bound to vrid.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ifnum	<String>	Read-write	Interfaces to bind to the VMAC, specified in (slot/port) notation (for example, 1/2).Use spaces to separate multiple entries.
trackifnum	<String>	Read-write	Interfaces which need to be tracked for this vrID.
ifaces	<String>	Read-write	Interfaces bound to this VRID.
id	<Double>	Read-write	Integer that uniquely identifies the VMAC address. The generic VMAC address is in the form of 00:00:5e:00:01:<VRID>. For example, if you add a VRID with a value of 60 and bind it to an interface, the resulting VMAC address is 00:00:5e:00:01:3c, where 3c is the hexadecimal representation of 60. Minimum value = 1 Maximum value = 255
flags	<Double>	Read-only	Flags.
vlan	<Double>	Read-only	The VLAN in which this VRID resides.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vrid_interface_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vrid_interface_binding":{
    "id":<Double_value>,
    "ifnum":<String_value>,
    "trackifnum":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid_interface_binding/id_value<Double>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid_interface_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vrid_interface_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid_interface_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vrid_interface_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vrid_interface_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "vrid_interface_binding": [ {
    "ifnum":<String_value>,
    "trackifnum":<String_value>,
    "ifaces":<String_value>,
    "id":<Double_value>,
    "flags":<Double_value>,
    "vlan":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid_interface_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vrid_interface_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vrid_nsip6_binding

May 20, 2016

Binding object showing the nsip6 that can be bound to vrid.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Integer that uniquely identifies the VMAC address. The generic VMAC address is in the form of 00:00:5e:00:01:<VRID>. For example, if you add a VRID with a value of 60 and bind it to an interface, the resulting VMAC address is 00:00:5e:00:01:3c, where 3c is the hexadecimal representation of 60. Minimum value = 1 Maximum value = 255
ipaddress	<String>	Read-write	The IP address bound to the VRID.
flags	<Double>	Read-only	Flags.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid_nsip6_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vrid_nsip6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid_nsip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vrid_nsip6_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vrid_nsip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid_nsip6_binding": [ {
      "id":<Double_value>,
      "ipaddress":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "flags":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid_nsip6_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vrid_nsip6_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vrid_nsip_binding

May 20, 2016

Binding object showing the nsip that can be bound to vrid.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	Integer that uniquely identifies the VMAC address. The generic VMAC address is in the form of 00:00:5e:00:01:<VRID>. For example, if you add a VRID with a value of 60 and bind it to an interface, the resulting VMAC address is 00:00:5e:00:01:3c, where 3c is the hexadecimal representation of 60. Minimum value = 1 Maximum value = 255
ipaddress	<String>	Read-write	The IP address bound to the VRID.
flags	<Double>	Read-only	Flags.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid_nsip_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vrid_nsip_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vrid_nsip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vrid_nsip_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vrid_nsip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vrid_nsip_binding": [ {
      "id":<Double_value>,
      "ipaddress":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "flags":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vrid_nsip_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vrid_nsip_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vxlan

May 20, 2016

Configuration for "VXLAN" resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN. Minimum value = 1 Maximum value = 16777215
vlan	<Double>	Read-write	ID of VLANs whose traffic is allowed over this VXLAN. If you do not specify any VLAN IDs, the NetScaler allows traffic of all VLANs that are not part of any other VXLANs. Minimum value = 1 Maximum value = 4094
port	<Integer>	Read-write	Specifies UDP destination port for VXLAN packets. Default value: 4789 Minimum value = 1 Maximum value = 65534
dynamicrouting	<String>	Read-write	Enable dynamic routing on this VXLAN. Default value: DISABLED Possible values = ENABLED, DISABLED
ipv6dynamicrouting	<String>	Read-write	Enable all IPv6 dynamic routing protocols on this VXLAN. Note: For the ENABLED setting to work, you must configure IPv6 dynamic routing protocols from the VTYSH command line. Default value: DISABLED Possible values = ENABLED, DISABLED
td	<Double>	Read-only	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vxlan":{
    "id":<Double_value>,
    "vlan":<Double_value>,
    "port":<Integer_value>,
    "dynamicrouting":<String_value>,
    "ipv6dynamicrouting":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vxlan":{
    "id":<Double_value>,
    "vlan":<Double_value>,
    "port":<Integer_value>,
    "dynamicrouting":<String_value>,
    "ipv6dynamicrouting":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan?**action=unset**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vxlan":{
    "id":<Double_value>,
    "vlan":true,
    "port":true,
    "dynamicrouting":true,
    "ipv6dynamicrouting":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vxlan?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vxlan?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vxlan resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vxlan?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vxlan?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vxlan resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vxlan": [ {
      "id":<Double_value>,
      "vlan":<Double_value>,
      "port":<Integer_value>,
      "dynamicrouting":<String_value>,
      "ipv6dynamicrouting":<String_value>,
      "td":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan/id_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vxlan/id_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vxlan/id_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vxlan": [ {
      "id":<Double_value>,
      "vlan":<Double_value>,
      "port":<Integer_value>,
      "dynamicrouting":<String_value>,
      "ipv6dynamicrouting":<String_value>,
      "td":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vxlan": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vxlan_binding

May 20, 2016

Binding object which returns the resources bound to vxlan.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN. Minimum value = 1 Maximum value = 16777215
vxlan_iptunnel_binding	<vxlan_iptunnel_binding[]>	Read-only	iptunnel that can be bound to vxlan.
vxlan_nsip_binding	<vxlan_nsip_binding[]>	Read-only	nsip that can be bound to vxlan.
vxlan_nsip6_binding	<vxlan_nsip6_binding[]>	Read-only	nsip6 that can be bound to vxlan.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan_binding/id_value<Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "vxlan_binding": [ {
      "id":<Double_value>,
      "vxlan_nsip6_binding":<vxlan_nsip6_binding[]_value>,
      "vxlan_nsip_binding":<vxlan_nsip_binding[]_value>,
      "vxlan_iptunnel_binding":<vxlan_iptunnel_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vxlan_iptunnel_binding

May 20, 2016

Binding object showing the iptunnel that can be bound to vxlan.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN. Minimum value = 1 Maximum value = 16777215
tunnel	<String>	Read-write	Specifies the name of the configured tunnel to be associated with this VXLAN.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vxlan_iptunnel_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vxlan_iptunnel_binding":{
    "id":<Double_value>,
    "tunnel":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan_iptunnel_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan_iptunnel_binding/id_value<Double>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vxlan_iptunnel_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vxlan_iptunnel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vxlan_iptunnel_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vxlan_iptunnel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vxlan_iptunnel_binding": [ {
      "id":<Double_value>,
      "tunnel":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan_iptunnel_binding/id_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vxlan_iptunnel_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vxlan_nsip6_binding

May 20, 2016

Binding object showing the nsip6 that can be bound to vxlan.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN. Minimum value = 1 Maximum value = 16777215
netmask	<String>	Read-write	Subnet mask for the network address defined for this VXLAN. Minimum length = 1
ipaddress	<String>	Read-write	The IP address assigned to the VXLAN.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vxlan_nsip6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vxlan_nsip6_binding":{
    "id":<Double_value>,
    "ipaddress":<String_value>,
    "netmask":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vxlan_nsip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vxlan_nsip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vxlan_nsip6_binding": [ {
      "id":<Double_value>,
      "netmask":<String_value>,
      "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan_nsip6_binding/id_value<Double>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vxlan_nsip6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vxlan_nsip_binding

May 20, 2016

Binding object showing the nsip that can be bound to vxlan.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	A positive integer, which is also called VXLAN Network Identifier (VNI), that uniquely identifies a VXLAN. Minimum value = 1 Maximum value = 16777215
netmask	<String>	Read-write	Subnet mask for the network address defined for this VXLAN.
ipaddress	<String>	Read-write	The IP address assigned to the VXLAN.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vxlan_nsip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vxlan_nsip_binding":{
    "id":<Double_value>,
    "ipaddress":<String_value>,
    "netmask":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vxlan_nsip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vxlan_nsip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vxlan_nsip_binding": [ {
      "id":<Double_value>,
      "netmask":<String_value>,
      "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vxlan_nsip_binding/id_value<Double>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vxlan_nsip_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

NS

May 20, 2016

System/Global level configuration.

Name	Description
nsacl	Configuration for ACL entry
nsacl6	Configuration for ACL6 entry
nsacls	Configuration for ACL entry
nsacls6	Configuration for ACL6 entry
nsappflowcollector	Configuration for appflowCollector
nsappflowparam	Configuration for appflowParam
nsaptlicense	Configuration for aptlicense
nsassignment	Configuration for assignment
nsconfig	Configuration for system config
nsconnectiontable	Configuration for TCP/IP connection table
nsconsoleloginprompt	
nsdhcpip	Configuration for dhcp IP
nsdhcpparams	Configuration for DHCP parameters
nsdiameter	Configuration for Diameter Parameters
nsencryptionparams	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsevents	
Name	**Description**
nsextension	Configuration for Extension
nsextension_binding	Binding object showing the resources that can be bound to nsextension
nsextension_extensionfunction_binding	Binding object showing the extensionfunction that can be bound to nsextension
nsfeature	Configuration for feature
nshardware	Configuration for hardware
nshostname	Configuration for host name
nshttpparam	Configuration for HTTP parameter
nshttpprofile	Configuration for HTTP profile
nsip	Configuration for ip
nsip6	Configuration for ip6
nslicense	Configuration for license
nslicenseproxyserver	Configuration for licenseproxyserver
nslimitidentifier	Configuration for limit Indetifier
nslimitidentifier_binding	Binding object showing the resources that can be bound to nslimitidentifier
nslimitidentifier_nslimitsessions_binding	Binding object showing the nslimitsessions that can be bound to nslimitidentifier
nslimitselector	Configuration for limit selector
nslimitsessions	Configuration for limit sessions
nsmode	Configuration for ns mode

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsparam	
Name	**Description**
nspartition	Configuration for admin partition
nspartition_binding	Binding object showing the resources that can be bound to nspartition
nspartition_bridgegroup_binding	Binding object showing the bridgegroup that can be bound to nspartition
nspartition_vlan_binding	Binding object showing the vlan that can be bound to nspartition
nspbr	Configuration for Policy Based Routing(PBR) entry
nspbr6	Configuration for PBR6 entry
nspbrs	Configuration for Policy based routing
nsratecontrol	Configuration for rate control
nsrollbackcmd	Configuration for Generate rollback commands
nsrpcnode	Configuration for rpc node
nsrunningconfig	Configuration for running configuration
nssavedconfig	Configuration for last saved configuration
nsservicefunction	Configuration for service Function
nsservicepath	Configuration for service Chain
nsservicepath_binding	Binding object showing the resources that can be bound to nsservicepath
nsservicepath_nsservicefunction_binding	Binding object showing the nsservicefunction that can be bound to nsservicepath
nssimpleacl	Configuration for simple ACL
nssimpleacl6	Configuration for simple ACL6

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
nsspparams	Configuration for Surge Protection parameter
nsstats	Configuration for stats
nssurgeq	Configuration for surge queue
nstcpbufparam	Configuration for tcp buffer parameter
nstcpparam	Configuration for tcp parameters
nstcpprofile	Configuration for TCP profile
nstimeout	Configuration for timeout
nstimer	Configuration for Timer
nstimer_autoscalepolicy_binding	Binding object showing the autoscalepolicy that can be bound to nstimer
nstimer_binding	Binding object showing the resources that can be bound to nstimer
nstrafficdomain	Configuration for Traffic Domain
nstrafficdomain_binding	Binding object showing the resources that can be bound to nstrafficdomain
nstrafficdomain_bridgegroup_binding	Binding object showing the bridgegroup that can be bound to nstrafficdomain
nstrafficdomain_vlan_binding	Binding object showing the vlan that can be bound to nstrafficdomain
nstrafficdomain_vxlan_binding	Binding object showing the vxlan that can be bound to nstrafficdomain
nsvariable	Configuration for variable
nsversion	Configuration for version
nsweblogparam	Configuration for Web log parameters
nsxmlnamespace	Configuration for XML namespace

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
reboot	
shutdown	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsacl

May 20, 2016

Configuration for ACL entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
aclname	<String>	Read-write	Name for the extended ACL rule. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the extended ACL rule is created. Minimum length = 1
aclaction	<String>	Read-write	Action to perform on incoming IPv4 packets that match the extended ACL rule. Available settings function as follows: * ALLOW - The NetScaler appliance processes the packet. * BRIDGE - The NetScaler appliance bridges the packet to the destination without processing it. * DENY - The NetScaler appliance drops the packet. Possible values = BRIDGE, DENY, ALLOW
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcip	<Boolean>	Read-write	IP address or range of IP addresses to match against the source IP address of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [10.102.29.30-10.102.29.189].
srcipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
srcipval	<String>	Read-write	IP address or range of IP addresses to match against the source IP address of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [10.102.29.30-10.102.29.189].
srcport	<Boolean>	Read-write	Port number or range of port numbers to match against the source port number of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90].
srcportop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
			Port number or range of port numbers to match against the source port number of

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			hyphen and enclose within brackets. For example: [40-90]. Maximum length = 65535
destip	<Boolean>	Read-write	IP address or range of IP addresses to match against the destination IP address of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [10.102.29.30-10.102.29.189].
destipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
destipval	<String>	Read-write	IP address or range of IP addresses to match against the destination IP address of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [10.102.29.30-10.102.29.189].
destport	<Boolean>	Read-write	Port number or range of port numbers to match against the destination port number of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90]. Note: The destination port can be specified only for TCP and UDP protocols.
destportop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
destportval	<String>	Read-write	Port number or range of port numbers to match against the destination port number of an incoming IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90]. Note: The destination port can be specified only for TCP and UDP protocols. Maximum length = 65535
ttl	<Double>	Read-write	Number of seconds, in multiples of four, after which the extended ACL rule expires. If you do not want the extended ACL rule to expire, do not specify a TTL value. Minimum value = 1 Maximum value = 2147483647
srcmac	<String>	Read-write	MAC address to match against the source MAC address of an incoming IPv4 packet.
srcmacmask	<String>	Read-write	Used to define range of Source MAC address. It takes string of 0 and 1, 0s are for exact match and 1s for wildcard. For matching first 3 bytes of MAC address, srcMacMask value "000000111111". . Default value: "000000000000"
protocol	<String>	Read-write	Protocol to match against the protocol of an incoming IPv4 packet. Possible values = ICMP, IGMP, TCP, EGP, IGP, ARGUS, UDP, RDP, RSVP, EIGRP, L2TP, ISIS
protocolnumber	<Double>	Read-write	Protocol to match against the protocol of an incoming IPv4 packet. Minimum value = 1 Maximum value = 255
vlan	<Double>	Read-write	ID of the VLAN. The NetScaler appliance applies the ACL rule only to the incoming packets of the specified VLAN. If you do not specify a VLAN ID, the appliance applies the ACL rule to the incoming packets on all VLANs. Minimum value = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
vxlan	<Double>	Read-write	VXLAN. The NetScaler appliance applies the ACL rule only to the incoming packets of the specified VXLAN. If you do not specify a VXLAN ID, the appliance applies the ACL rule to the incoming packets on all VXLANs. Minimum value = 1 Maximum value = 16777215
Interface	<String>	Read-write	ID of an interface. The NetScaler appliance applies the ACL rule only to the incoming packets from the specified interface. If you do not specify any value, the appliance applies the ACL rule to the incoming packets of all interfaces.
established	<Boolean>	Read-write	Allow only incoming TCP packets that have the ACK or RST bit set, if the action set for the ACL rule is ALLOW and these packets match the other conditions in the ACL rule.
icmptype	<Double>	Read-write	ICMP Message type to match against the message type of an incoming ICMP packet. For example, to block DESTINATION UNREACHABLE messages, you must specify 3 as the ICMP type. Note: This parameter can be specified only for the ICMP protocol. Minimum value = 0 Maximum value = 65536
icmpcode	<Double>	Read-write	Code of a particular ICMP message type to match against the ICMP code of an incoming ICMP packet. For example, to block DESTINATION HOST UNREACHABLE messages, specify 3 as the ICMP type and 1 as the ICMP code. If you set this parameter, you must set the ICMP Type parameter. Minimum value = 0 Maximum value = 65536
priority	<Double>	Read-write	Priority for the extended ACL rule that determines the order in which it is evaluated relative to the other extended ACL rules. If you do not specify priorities while creating extended ACL rules, the ACL rules are evaluated in the order in which they are created. Minimum value = 1 Maximum value = 100000
state	<String>	Read-write	Enable or disable the extended ACL rule. After you apply the extended ACL rules, the NetScaler appliance compares incoming packets against the enabled extended ACL rules. Default value: ENABLED Possible values = ENABLED, DISABLED
logstate	<String>	Read-write	Enable or disable logging of events related to the extended ACL rule. The log messages are stored in the configured syslog or auditlog server. Default value: DISABLED Possible values = ENABLED, DISABLED
ratelimit	<Double>	Read-write	Maximum number of log messages to be generated per second. If you set this parameter, you must enable the Log State parameter. Default value: 100 Minimum value = 1 Maximum value = 10000

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
newname	<String>	Read-write	underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
hits	<Double>	Read-only	The hits of this ACL.
kernelstate	<String>	Read-only	The commit status of the ACL. Default value: NOTAPPLIED Possible values = APPLIED, NOTAPPLIED, RE-APPLY, SFAPPLIED, SFNOTAPPLIED, SFAPPLIED61, SFNOTAPPLIED61
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacl":{
    "aclname":<String_value>,
    "aclaction":<String_value>,
    "td":<Double_value>,
    "srcip":<Boolean_value>,
    "srcipop":<String_value>,
    "srcipval":<String_value>,
    "srcport":<Boolean_value>,
    "srcportop":<String_value>,
    "srcportval":<String_value>,
    "destip":<Boolean_value>,
    "destipop":<String_value>,
    "destipval":<String_value>,
    "destport":<Boolean_value>,
    "destportop":<String_value>,
    "destportval":<String_value>,
    "ttl":<Double_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "Interface":<String_value>,
    "established":<Boolean_value>,
    "icmptype":<Double_value>,
    "icmpcode":<Double_value>,
    "priority":<Double_value>,
    "state":<String_value>,
    "logstate":<String_value>,
    "ratelimit":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl/aclname_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacl":{
    "aclname":<String_value>,
    "aclaction":<String_value>,
    "srcip":<Boolean_value>,
    "srcipop":<String_value>,
    "srcipval":<String_value>,
    "srcport":<Boolean_value>,
    "srcportop":<String_value>,
    "srcportval":<String_value>,
    "destip":<Boolean_value>,
    "destipop":<String_value>,
    "destipval":<String_value>,
    "destport":<Boolean_value>,
    "destportop":<String_value>,
    "destportval":<String_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "icmptype":<Double_value>,
    "icmpcode":<Double_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "Interface":<String_value>,
    "priority":<Double_value>,
    "logstate":<String_value>,
    "ratelimit":<Double_value>,
    "established":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacl":{
    "aclname":<String_value>,
    "srcip":true,
    "srcport":true,
    "destip":true,
    "destport":true,
    "srcmac":true,
    "srcmacmask":true,
    "protocol":true,
    "icmptype":true,
    "icmpcode":true,
    "vlan":true,
    "vxlan":true,
    "Interface":true,
    "logstate":true,
    "ratelimit":true,
    "established":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"nsacl":{
    "aclname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacl":{
    "aclname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacl":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "aclname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsacl?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsacl?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsacl resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsacl?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsacl?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsacl resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsacl": [ {
      "aclname":<String_value>,
      "td":<Double_value>,
      "aclaction":<String_value>,
      "srcmac":<String_value>,
      "srcmacmask":<String_value>,
      "protocol":<String_value>,
      "protocolnumber":<Double_value>,
      "srcportval":<String_value>,
      "destportval":<String_value>,
      "srcipval":<String_value>,
      "destipval":<String_value>,
      "vlan":<Double_value>,
      "vxlan":<Double_value>,
      "state":<String_value>,
      "ttl":<Double_value>,
      "icmptype":<Double_value>,
      "icmpcode":<Double_value>,
      "Interface":<String_value>,
      "hits":<Double_value>,
      "established":<Boolean_value>,
      "priority":<Double_value>,
      "srcipop":<String_value>,
      "destipop":<String_value>,
      "srcportop":<String_value>,
      "destportop":<String_value>,
      "kernelstate":<String_value>,
      "logstate":<String_value>,
      "ratelimit":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl/aclname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsacl/aclname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nsacl/aclname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsacl": [ {
    "aclname":<String_value>,
    "td":<Double_value>,
    "aclaction":<String_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "srcportval":<String_value>,
    "destportval":<String_value>,
    "srcipval":<String_value>,
    "destipval":<String_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "state":<String_value>,
    "ttl":<Double_value>,
    "icmptype":<Double_value>,
    "icmpcode":<Double_value>,
    "Interface":<String_value>,
    "hits":<Double_value>,
    "established":<Boolean_value>,
    "priority":<Double_value>,
    "srcipop":<String_value>,
    "destipop":<String_value>,
    "srcportop":<String_value>,
    "destportop":<String_value>,
    "kernelstate":<String_value>,
    "logstate":<String_value>,
    "ratelimit":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsacl": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsacl6

Jul 07, 2017

Configuration for ACL6 entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
acl6name	<String>	Read-write	Name for the ACL6 rule. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the ACL6 rule is created. Minimum length = 1
acl6action	<String>	Read-write	Action to perform on the incoming IPv6 packets that match the ACL6 rule. Available settings function as follows: * ALLOW - The NetScaler appliance processes the packet. * BRIDGE - The NetScaler appliance bridges the packet to the destination without processing it. * DENY - The NetScaler appliance drops the packet. Possible values = BRIDGE, DENY, ALLOW
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcipv6	<Boolean>	Read-write	IP address or range of IP addresses to match against the source IP address of an incoming IPv6 packet. In the command line interface, separate the range with a hyphen and enclose within brackets.
srcipop	<String>	Read-write	Logical operator. Possible values = =, !=, EQ, NEQ
srcipv6val	<String>	Read-write	Source IPv6 address (range).
srcport	<Boolean>	Read-write	Port number or range of port numbers to match against the source port number of an incoming IPv6 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90]. Note: The destination port can be specified only for TCP and UDP protocols.
srcportop	<String>	Read-write	Logical operator. Possible values = =, !=, EQ, NEQ

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum length = 65535
destipv6	<Boolean>	Read-write	IP address or range of IP addresses to match against the destination IP address of an incoming IPv6 packet. In the command line interface, separate the range with a hyphen and enclose within brackets.
destipop	<String>	Read-write	Logical operator. Possible values = =, !=, EQ, NEQ
destipv6val	<String>	Read-write	Destination IPv6 address (range).
destport	<Boolean>	Read-write	Port number or range of port numbers to match against the destination port number of an incoming IPv6 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90]. Note: The destination port can be specified only for TCP and UDP protocols.
destportop	<String>	Read-write	Logical operator. Possible values = =, !=, EQ, NEQ
destportval	<String>	Read-write	Destination port (range). Maximum length = 65535
ttl	<Double>	Read-write	Time to expire this ACL6 (in seconds). Minimum value = 1 Maximum value = 2147483647
srcmac	<String>	Read-write	MAC address to match against the source MAC address of an incoming IPv6 packet.
srcmacmask	<String>	Read-write	Used to define range of Source MAC address. It takes string of 0 and 1, 0s are for exact match and 1s for wildcard. For matching first 3 bytes of MAC address, srcMacMask value "000000111111". . Default value: "000000000000"
protocol	<String>	Read-write	Protocol, identified by protocol name, to match against the protocol of an incoming IPv6 packet. Possible values = ICMPV6, TCP, UDP
protocolnumber	<Double>	Read-write	Protocol, identified by protocol number, to match against the protocol of an incoming IPv6 packet. Minimum value = 1 Maximum value = 255
vlan	<Double>	Read-write	ID of the VLAN. The NetScaler appliance applies the ACL6 rule only to the incoming packets on the specified VLAN. If you do not specify a VLAN ID, the appliance applies the ACL6 rule to the incoming packets on all VLANs. Minimum value = 1 Maximum value = 4094
			ID of the VXLAN. The NetScaler appliance applies the ACL6 rule only to the incoming packets on the specified VXLAN. If you do not specify a VXLAN ID, the appliance applies the ACL6 rule to the incoming packets on all VXLANs.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum value = 16777215
Interface	<String>	Read-write	ID of an interface. The NetScaler appliance applies the ACL6 rule only to the incoming packets from the specified interface. If you do not specify any value, the appliance applies the ACL6 rule to the incoming packets from all interfaces.
established	<Boolean>	Read-write	Allow only incoming TCP packets that have the ACK or RST bit set if the action set for the ACL6 rule is ALLOW and these packets match the other conditions in the ACL6 rule.
icmptype	<Double>	Read-write	ICMP Message type to match against the message type of an incoming IPv6 ICMP packet. For example, to block DESTINATION UNREACHABLE messages, you must specify 3 as the ICMP type. Note: This parameter can be specified only for the ICMP protocol. Minimum value = 0 Maximum value = 65536
icmpcode	<Double>	Read-write	Code of a particular ICMP message type to match against the ICMP code of an incoming IPv6 ICMP packet. For example, to block DESTINATION HOST UNREACHABLE messages, specify 3 as the ICMP type and 1 as the ICMP code. If you set this parameter, you must set the ICMP Type parameter. Minimum value = 0 Maximum value = 65536
priority	<Double>	Read-write	Priority for the ACL6 rule, which determines the order in which it is evaluated relative to the other ACL6 rules. If you do not specify priorities while creating ACL6 rules, the ACL6 rules are evaluated in the order in which they are created. Minimum value = 1 Maximum value = 80000
state	<String>	Read-write	State of the ACL6. Default value: ENABLED Possible values = ENABLED, DISABLED
aclaction	<String>	Read-write	Action associated with the ACL6. Possible values = BRIDGE, DENY, ALLOW
newname	<String>	Read-write	New name for the ACL6 rule. Must begin with an ASCII alphabetic or underscore \(_\) character, and must contain only ASCII alphanumeric, underscore, hash \(\#\), period \(.\), space, colon \(:\), at \(@\), equals \(=\), and hyphen \(-\) characters. Minimum length = 1
kernelstate	<String>	Read-only	Commit status of the ACL6. Default value: NOTAPPLIED Possible values = APPLIED, NOTAPPLIED, RE-APPLY, SFAPPLIED, SFNOTAPPLIED
hits	<Double>	Read-only	Number of hits of this ACL6.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl6

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacl6":{
    "acl6name":<String_value>,
    "acl6action":<String_value>,
    "td":<Double_value>,
    "srcipv6":<Boolean_value>,
    "srcipop":<String_value>,
    "srcipv6val":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "srcport":<Boolean_value>,
    "srcportop":<String_value>,
    "srcportval":<String_value>,
    "destipv6":<Boolean_value>,
    "destipop":<String_value>,
    "destipv6val":<String_value>,
    "destport":<Boolean_value>,
    "destportop":<String_value>,
    "destportval":<String_value>,
    "ttl":<Double_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "Interface":<String_value>,
    "established":<Boolean_value>,
    "icmptype":<Double_value>,
    "icmpcode":<Double_value>,
    "priority":<Double_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl6/acl6name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl6

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacl6":{
    "acl6name":<String_value>,
    "aclaction":<String_value>,
    "srcipv6":<Boolean_value>,
    "srcipop":<String_value>,
    "srcipv6val":<String_value>,
    "srcport":<Boolean_value>,
    "srcportop":<String_value>,
    "srcportval":<String_value>,
    "destipv6":<Boolean_value>,
    "destipop":<String_value>,
    "destipv6val":<String_value>,
    "destport":<Boolean_value>,
    "destportop":<String_value>,
    "destportval":<String_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "icmptype":<Double_value>,
    "icmpcode":<Double_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "Interface":<String_value>,
    "priority":<Double_value>,
    "established":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl6?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"nsacl6":{
    "acl6name":<String_value>,
    "srcipv6":true,
    "srcport":true,
    "destipv6":true,
    "destport":true,
    "srcmac":true,
    "srcmacmask":true,
    "protocol":true,
    "icmptype":true,
    "icmpcode":true,
    "vlan":true,
    "vxlan":true,
    "Interface":true,
    "established":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl6?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacl6":{
    "acl6name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl6?**action=disable**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacl6":{
    "acl6name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl6?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacl6":{
    "acl6name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl6

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nsacl6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsacl6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsacl6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsacl6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsacl6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsacl6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsacl6": [ {
    "acl6name":<String_value>,
    "acl6action":<String_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "srcportval":<String_value>,
    "destportval":<String_value>,
    "srcipv6val":<String_value>,
    "destipv6val":<String_value>,
    "vlan":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "vxlan":<Double_value>,
    "state":<String_value>,
    "kernelstate":<String_value>,
    "ttl":<Double_value>,
    "icmptype":<Double_value>,
    "icmpcode":<Double_value>,
    "Interface":<String_value>,
    "hits":<Double_value>,
    "established":<Boolean_value>,
    "priority":<Double_value>,
    "srcipop":<String_value>,
    "destipop":<String_value>,
    "srcportop":<String_value>,
    "destportop":<String_value>,
    "td":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl6/acl6name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsacl6/acl6name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nsacl6/acl6name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "nsacl6": [ {
    "acl6name":<String_value>,
    "acl6action":<String_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "srcportval":<String_value>,
    "destportval":<String_value>,
    "srcipv6val":<String_value>,
    "destipv6val":<String_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "state":<String_value>,
    "kernelstate":<String_value>,
    "ttl":<Double_value>,
    "icmptype":<Double_value>,
    "icmpcode":<Double_value>,
    "Interface":<String_value>,
    "hits":<Double_value>,
    "established":<Boolean_value>,
    "priority":<Double_value>,
    "srcipop":<String_value>,
    "destipop":<String_value>,
    "srcportop":<String_value>,
    "destportop":<String_value>,
    "td":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacl6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "nsacl6": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsacls

Jul 07, 2017

Configuration for ACL entry resource.

Operations (click to see Properties)

RENUMBER | CLEAR | APPLY

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

renumber

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacls?**action=renumber**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"nsacls":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacls?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacls":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

apply

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacls?**action=apply**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacls":{
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsacls6

May 20, 2016

Configuration for ACL6 entry resource.

Operations (click to see Properties)

CLEAR | APPLY | RENUMBER

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacls6?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"nsacls6":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

apply

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacls6?**action=apply**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacls6":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

renumber

URL: http://<netscaler-ip-address>/nitro/v1/config/nsacls6?**action=renumber**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsacls6":{
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsappflowcollector

May 20, 2016

Configuration for appflowCollector resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the AppFlow collector. Minimum length = 1 Maximum length = 127
ipaddress	<String>	Read-write	The IPv4 address of the AppFlow collector.
port	<Integer>	Read-write	The UDP port on which the AppFlow collector is listening. Default value: 4739
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsappflowcollector":{
    "name":<String_value>,
    "ipaddress":<String_value>,
    "port":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsappflowcollector resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsappflowcollector resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsappflowcollector": [ {
      "name":<String_value>,
      "ipaddress":<String_value>,
      "port":<Integer_value>
}]}
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsappflowcollector": [ {
      "name":<String_value>,
      "ipaddress":<String_value>,
      "port":<Integer_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsappflowcollector?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "nsappflowcollector": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsappflowparam

Jul 07, 2017

Configuration for appflowParam resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
templaterefresh	<Double>	Read-write	IPFIX template refresh interval (in seconds). Default value: 600 Minimum value = 60 Maximum value = 3600
udppmtu	<Double>	Read-write	MTU to be used for IPFIX UDP packets. Default value: 1472 Minimum value = 128 Maximum value = 1472
httpurl	<String>	Read-write	Enable AppFlow HTTP URL logging. Default value: DISABLED Possible values = ENABLED, DISABLED
httpcookie	<String>	Read-write	Enable AppFlow HTTP cookie logging. Default value: DISABLED Possible values = ENABLED, DISABLED
httpreferer	<String>	Read-write	Enable AppFlow HTTP referer logging. Default value: DISABLED Possible values = ENABLED, DISABLED
httpmethod	<String>	Read-write	Enable AppFlow HTTP method logging. Default value: DISABLED Possible values = ENABLED, DISABLED
httphost	<String>	Read-write	Enable AppFlow HTTP host logging. Default value: DISABLED Possible values = ENABLED, DISABLED
httpuseragent	<String>	Read-write	Enable AppFlow HTTP user-agent logging. Default value: DISABLED Possible values = ENABLED, DISABLED
clienttrafficonly	<String>	Read-write	Control whether AppFlow records should be generated only for client-side traffic. Default value: NO Possible values = YES, NO

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsappflowparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsappflowparam":{
    "templaterefresh":<Double_value>,
    "udppmtu":<Double_value>,
    "httpurl":<String_value>,
    "httpcookie":<String_value>,
    "httpreferer":<String_value>,
    "httpmethod":<String_value>,
    "httphost":<String_value>,
    "httpuseragent":<String_value>,
    "clienttrafficonly":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsappflowparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsappflowparam":{
    "templaterefresh":true,
    "udppmtu":true,
    "httpurl":true,
    "httpcookie":true,
    "httpreferer":true,
    "httpmethod":true,
    "httphost":true,
    "httpuseragent":true,
    "clienttrafficonly":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsappflowparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsappflowparam": [ {
      "templaterefresh":<Double_value>,
      "udppmtu":<Double_value>,
      "httpurl":<String_value>,
      "httpcookie":<String_value>,
      "httpreferer":<String_value>,
      "httpmethod":<String_value>,
      "httphost":<String_value>,
      "httpuseragent":<String_value>,
      "clienttrafficonly":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsaptlicense

Jul 07, 2017

Configuration for aptlicense resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
serialno	<String>	Read-write	Hardware Serial Number/License Activation Code(LAC).
useproxy	<String>	Read-write	Specifies whether to use the licenseproxyserver to reach the internet. Make sure to configure licenseproxyserver to use this option. Default value: NO Possible values = YES, NO
id	<String>	Read-write	License ID.
sessionid	<String>	Read-write	Session ID.
bindtype	<String>	Read-write	Bind type.
countavailable	<String>	Read-write	The user can allocate one or more licenses. Ensure the value is less than (for partial allocation) or equal to the total number of available licenses.
licensedir	<String>	Read-write	License Directory.
response	<String>	Read-only	Response data as text blob.
counttotal	<String>	Read-only	Count.
name	<String>	Read-only	License name.
relevance	<String>	Read-only	License relevance.
datepurchased	<String>	Read-only	License purchase date.
datesa	<String>	Read-only	License SA date.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dateexp	<String>	Read-only	License expiry date.
Name	**Data Type**	**Permissions**	**Description**
features	<String[]>	Read-only	Features.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

CHANGE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

change

URL: http://<netscaler-ip-address>/nitro/v1/config/nsaptlicense?**action=update**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"nsaptlicense":{
    "id":<String_value>,
    "sessionid":<String_value>,
    "bindtype":<String_value>,
    "countavailable":<String_value>,
    "licensedir":<String_value>,
    "useproxy":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsaptlicense

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/nsaptlicense?**args=serialno:<String_value>,useproxy:<String_value>**

Use this query-parameter to get nsaptlicense resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsaptlicense": [ {
serialno:<String_value>,useproxy:<String_value>      "response":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "id":<String_value>,
    "sessionid":<String_value>,
    "bindtype":<String_value>,
    "countavailable":<String_value>,
    "counttotal":<String_value>,
    "name":<String_value>,
    "relevance":<String_value>,
    "datepurchased":<String_value>,
    "datesa":<String_value>,
    "dateexp":<String_value>,
    "features":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsaptlicense?**args=serialno:<String_value>,useproxy:<String_value>&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsaptlicense": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsassignment

May 20, 2016

Configuration for assignment resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the assignment. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the assignment is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my assignment" or ?my assignment?).
variable	<String>	Read-write	Left hand side of the assigment, of the form $variable-name (for a singleton variabled) or $variable-name[key-expression], where key-expression is a default syntax expression that evaluates to a text string and provides the key to select a map entry.
set	<String>	Read-write	Right hand side of the assignment. The default syntax expression is evaluated and assigned to theleft hand variable.
Add	<String>	Read-write	Right hand side of the assignment. The default syntax expression is evaluated and added to the left hand variable.
sub	<String>	Read-write	Right hand side of the assignment. The default syntax expression is evaluated and subtracted from the left hand variable.
append	<String>	Read-write	Right hand side of the assignment. The default syntax expression is evaluated and appended to the left hand variable.
clear	<Boolean>	Read-write	Clear the variable value. Deallocates a text value, and for a map, the text key.
comment	<String>	Read-write	Comment. Can be used to preserve information about this rewrite action.
newname	<String>	Read-write	New name for the assignment. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the rewrite policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my assignment" or ?my assignment?).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
hits	<Double>	Read-only	The number of times the action has been taken.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
referencecount	<Double>	Read-only	The number of references to the action.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsassignment

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsassignment":{
    "name":<String_value>,
    "variable":<String_value>,
    "set":<String_value>,
    "Add":<String_value>,
    "sub":<String_value>,
    "append":<String_value>,
    "clear":<Boolean_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsassignment

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsassignment":{
    "name":<String_value>,
    "variable":<String_value>,
    "set":<String_value>,
    "Add":<String_value>,
    "sub":<String_value>,
    "append":<String_value>,
    "clear":<Boolean_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsassignment?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsassignment":{
    "name":<String_value>,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsassignment/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/nsassignment?**action=rename**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsassignment":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsassignment

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsassignment?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsassignment?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsassignment resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsassignment?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsassignment?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsassignment resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsassignment": [ {
      "name":<String_value>,
      "variable":<String_value>,
      "set":<String_value>,
      "Add":<String_value>,
      "sub":<String_value>,
      "append":<String_value>,
      "clear":<Boolean_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "referencecount":<Double_value>,
      "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsassignment/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsassignment/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nsassignment/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsassignment": [ {
      "name":<String_value>,
      "variable":<String_value>,
      "set":<String_value>,
      "Add":<String_value>,
      "sub":<String_value>,
      "append":<String_value>,
      "clear":<Boolean_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "referencecount":<Double_value>,
      "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsassignment?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "nsassignment": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsconfig

Jul 07, 2017

Configuration for system config resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
force	<Boolean>	Read-write	Configurations will be cleared without prompting for confirmation.
level	<String>	Read-write	Types of configurations to be cleared. * basic: Clears all configurations except the following: - NSIP, default route (gateway), MIPs, and SNIPs - Network settings (DG, VLAN, RHI and DNS settings) - Cluster settings - HA node definitions - Feature and mode settings - nsroot password * extended: Clears the same configurations as the 'basic' option. In addition, it clears the nsroot password and feature and mode settings. * full: Clears all configurations except NSIP, default route, and interface settings. Note: When you clear the configurations through the cluster IP address, by specifying the level as 'full', the cluster is deleted and all cluster nodes become standalone appliances. The 'basic' and 'extended' levels are propagated to the cluster nodes. Possible values = basic, extended, full
ipaddress	<String>	Read-write	IP address of the NetScaler appliance. Commonly referred to as NSIP address. This parameter is mandatory to bring up the appliance. Minimum length = 1
netmask	<String>	Read-write	Netmask corresponding to the IP address. This parameter is mandatory to bring up the appliance.
nsvlan	<Double>	Read-write	VLAN (NSVLAN) for the subnet on which the IP address resides. Minimum value = 2 Maximum value = 4094
ifnum	<String[]>	Read-write	Interfaces of the appliances that must be bound to the NSVLAN. Minimum length = 1
tagged	<String>	Read-write	Specifies that the interfaces will be added as 802.1q tagged interfaces. Packets sent on these interface on this VLAN will have an additional 4-byte 802.1q tag which identifies the VLAN. To use 802.1q tagging, the switch connected to the appliance's interfaces

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: YES Possible values = YES, NO
httpport	<Integer[]>	Read-write	The HTTP ports on the Web server. This allows the system to perform connection off-load for any client request that has a destination port matching one of these configured ports. Minimum value = 1
maxconn	<Double>	Read-write	The maximum number of connections that will be made from the system to the web server(s) attached to it. The value entered here is applied globally to all attached servers. Minimum value = 0 Maximum value = 4294967294
maxreq	<Double>	Read-write	The maximum number of requests that the system can pass on a particular connection between the system and a server attached to it. Setting this value to 0 allows an unlimited number of requests to be passed. Minimum value = 0 Maximum value = 65535
cip	<String>	Read-write	The option to control (enable or disable) the insertion of the actual client IP address into the HTTP header request passed from the client to one, some, or all servers attached to the system. The passed address can then be accessed through a minor modification to the server. l If cipHeader is specified, it will be used as the client IP header. l If it is not specified, then the value that has been set by the set ns config CLI command will be used as the client IP header. Possible values = ENABLED, DISABLED
cipheader	<String>	Read-write	The text that will be used as the client IP header. Minimum length = 1
cookieversion	<String>	Read-write	The version of the cookie inserted by system. Possible values = 0, 1
securecookie	<String>	Read-write	enable/disable secure flag for persistence cookie. Default value: ENABLED Possible values = ENABLED, DISABLED
pmtumin	<Double>	Read-write	The minimum Path MTU. Default value: 576 Minimum value = 168 Maximum value = 1500
pmtutimeout	<Double>	Read-write	The timeout value in minutes. Default value: 10 Minimum value = 1 Maximum value = 1440
ftpportrange	<String>	Read-write	Port range configured for FTP services. Minimum length = 1024 Maximum length = 64000

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
crportrange	<String>	Read-write	Minimum length = 1 Maximum length = 65535
			Name of the timezone. Possible values = CoordinatedUniversalTime, GMT+01:00-CET-Europe/Andorra, GMT+04:00-GST-Asia/Dubai, GMT+04:30-AFT-Asia/Kabul, GMT-04:00-AST-America/Antigua, GMT-04:00-AST-America/Anguilla, GMT+01:00-CET-Europe/Tirane, GMT+04:00-AMT-Asia/Yerevan, GMT+01:00-WAT-Africa/Luanda, GMT+13:00-NZDT-Antarctica/McMurdo, GMT+13:00-NZDT-Antarctica/South_Pole, GMT-03:00-ROTT-Antarctica/Rothera, GMT-04:00-CLT-Antarctica/Palmer, GMT+05:00-MAWT-Antarctica/Mawson, GMT+07:00-DAVT-Antarctica/Davis, GMT+08:00-WST-Antarctica/Casey, GMT+06:00-VOST-Antarctica/Vostok, GMT+10:00-DDUT-Antarctica/DumontDUrville, GMT+03:00-SYOT-Antarctica/Syowa, GMT+11:00-MIST-Antarctica/Macquarie, GMT-03:00-ART-America/Argentina/Buenos_Aires, GMT-03:00-ART-America/Argentina/Cordoba, GMT-03:00-ART-America/Argentina/Salta, GMT-03:00-ART-America/Argentina/Jujuy, GMT-03:00-ART-America/Argentina/Tucuman, GMT-03:00-ART-America/Argentina/Catamarca, GMT-03:00-ART-America/Argentina/La_Rioja, GMT-03:00-ART-America/Argentina/San_Juan, GMT-03:00-ART-America/Argentina/Mendoza, GMT-03:00-WARST-America/Argentina/San_Luis, GMT-03:00-ART-America/Argentina/Rio_Gallegos, GMT-03:00-ART-America/Argentina/Ushuaia, GMT-11:00-SST-Pacific/Pago_Pago, GMT+01:00-CET-Europe/Vienna, GMT+11:00-LHST-Australia/Lord_Howe, GMT+11:00-EST-Australia/Hobart, GMT+11:00-EST-Australia/Currie, GMT+11:00-EST-Australia/Melbourne, GMT+11:00-EST-Australia/Sydney, GMT+10:30-CST-Australia/Broken_Hill, GMT+10:00-EST-Australia/Brisbane, GMT+10:00-EST-Australia/Lindeman, GMT+10:30-CST-Australia/Adelaide, GMT+09:30-CST-Australia/Darwin, GMT+08:00-WST-Australia/Perth, GMT+08:45-CWST-Australia/Eucla, GMT-04:00-AST-America/Aruba, GMT+02:00-EET-Europe/Mariehamn, GMT+04:00-AZT-Asia/Baku, GMT+01:00-CET-Europe/Sarajevo, GMT-04:00-AST-America/Barbados, GMT+06:00-BDT-Asia/Dhaka, GMT+01:00-CET-Europe/Brussels, GMT+00:00-GMT-Africa/Ouagadougou, GMT+02:00-EET-Europe/Sofia, GMT+03:00-AST-Asia/Bahrain, GMT+02:00-CAT-Africa/Bujumbura, GMT+01:00-WAT-Africa/Porto-Novo, GMT-04:00-AST-America/St_Barthelemy, GMT-03:00-ADT-Atlantic/Bermuda, GMT+08:00-BNT-Asia/Brunei, GMT-04:00-BOT-America/La_Paz, GMT-02:00-FNT-America/Noronha, GMT-03:00-BRT-America/Belem, GMT-03:00-BRT-America/Fortaleza, GMT-03:00-BRT-America/Recife, GMT-03:00-BRT-America/Araguaina, GMT-03:00-BRT-America/Maceio, GMT-03:00-BRT-America/Bahia, GMT-03:00-BRT-America/Sao_Paulo, GMT-04:00-AMT-America/Campo_Grande, GMT-04:00-AMT-America/Cuiaba, GMT-03:00-BRT-America/Santarem, GMT-04:00-AMT-America/Porto_Velho, GMT-04:00-AMT-America/Boa_Vista, GMT-04:00-AMT-America/Manaus, GMT-04:00-AMT-America/Eirunepe, GMT-04:00-AMT-America/Rio_Branco, GMT-04:00-EDT-America/Nassau, GMT+06:00-BTT-Asia/Thimphu, GMT+02:00-CAT-Africa/Gaborone, GMT+03:00-FET-Europe/Minsk, GMT-06:00-CST-America/Belize, GMT-02:30-NDT-America/St_Johns, GMT-03:00-ADT-America/Halifax, GMT-03:00-ADT-America/Glace_Bay, GMT-03:00-ADT-America/Moncton, GMT-03:00-ADT-America/Goose_Bay, GMT-04:00-AST-America/Blanc-Sablon, GMT-04:00-EDT-America/Montreal, GMT-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			04:00-EDT-America/Thunder_Bay, GMT-04:00-EDT-America/Iqaluit, GMT-04:00-EDT-America/Pangnirtung, GMT-05:00-CDT-America/Resolute, GMT-05:00-EST-America/Atikokan, GMT-05:00-CDT-America/Rankin_Inlet, GMT-05:00-CDT-America/Winnipeg, GMT-05:00-CDT-America/Rainy_River, GMT-06:00-CST-America/Regina, GMT-06:00-CST-America/Swift_Current, GMT-06:00-MDT-America/Edmonton, GMT-06:00-MDT-America/Cambridge_Bay, GMT-06:00-MDT-America/Yellowknife, GMT-06:00-MDT-America/Inuvik, GMT-07:00-MST-America/Dawson_Creek, GMT-07:00-PDT-America/Vancouver, GMT-07:00-PDT-America/Whitehorse, GMT-07:00-PDT-America/Dawson, GMT+06:30-CCT-Indian/Cocos, GMT+01:00-WAT-Africa/Kinshasa, GMT+02:00-CAT-Africa/Lubumbashi, GMT+01:00-WAT-Africa/Bangui, GMT+01:00-WAT-Africa/Brazzaville, GMT+01:00-CET-Europe/Zurich, GMT+00:00-GMT-Africa/Abidjan, GMT-10:00-CKT-Pacific/Rarotonga, GMT-04:00-CLT-America/Santiago, GMT-06:00-EAST-Pacific/Easter, GMT+01:00-WAT-Africa/Douala, GMT+08:00-CST-Asia/Shanghai, GMT+08:00-CST-Asia/Harbin, GMT+08:00-CST-Asia/Chongqing, GMT+08:00-CST-Asia/Urumqi, GMT+08:00-CST-Asia/Kashgar, GMT-05:00-COT-America/Bogota, GMT-06:00-CST-America/Costa_Rica, GMT-04:00-CDT-America/Havana, GMT-01:00-CVT-Atlantic/Cape_Verde, GMT+07:00-CXT-Indian/Christmas, GMT+02:00-EET-Asia/Nicosia, GMT+01:00-CET-Europe/Prague, GMT+01:00-CET-Europe/Berlin, GMT+03:00-EAT-Africa/Djibouti, GMT+01:00-CET-Europe/Copenhagen, GMT-04:00-AST-America/Dominica, GMT-04:00-AST-America/Santo_Domingo, GMT+01:00-CET-Africa/Algiers, GMT-05:00-ECT-America/Guayaquil, GMT-06:00-GALT-Pacific/Galapagos, GMT+02:00-EET-Europe/Tallinn, GMT+02:00-EET-Africa/Cairo, GMT+00:00-WET-Africa/El_Aaiun, GMT+03:00-EAT-Africa/Asmara, GMT+01:00-CET-Europe/Madrid, GMT+01:00-CET-Africa/Ceuta, GMT+00:00-WET-Atlantic/Canary, GMT+03:00-EAT-Africa/Addis_Ababa, GMT+02:00-EET-Europe/Helsinki, GMT+12:00-FJT-Pacific/Fiji, GMT-03:00-FKST-Atlantic/Stanley, GMT+10:00-CHUT-Pacific/Chuuk, GMT+11:00-PONT-Pacific/Pohnpei, GMT+11:00-KOST-Pacific/Kosrae, GMT+00:00-WET-Atlantic/Faroe, GMT+01:00-CET-Europe/Paris, GMT+01:00-WAT-Africa/Libreville, GMT+00:00-GMT-Europe/London, GMT-04:00-AST-America/Grenada, GMT+04:00-GET-Asia/Tbilisi, GMT-03:00-GFT-America/Cayenne, GMT+00:00-GMT-Europe/Guernsey, GMT+00:00-GMT-Africa/Accra, GMT+01:00-CET-Europe/Gibraltar, GMT-03:00-WGT-America/Godthab, GMT+00:00-GMT-America/Danmarkshavn, GMT-01:00-EGT-America/Scoresbysund, GMT-03:00-ADT-America/Thule, GMT+00:00-GMT-Africa/Banjul, GMT+00:00-GMT-Africa/Conakry, GMT-04:00-AST-America/Guadeloupe, GMT+01:00-WAT-Africa/Malabo, GMT+02:00-EET-Europe/Athens, GMT-02:00-GST-Atlantic/South_Georgia, GMT-06:00-CST-America/Guatemala, GMT+10:00-ChST-Pacific/Guam, GMT+00:00-GMT-Africa/Bissau, GMT-04:00-GYT-America/Guyana, GMT+08:00-HKT-Asia/Hong_Kong, GMT-06:00-CST-America/Tegucigalpa, GMT+01:00-CET-Europe/Zagreb, GMT-05:00-EST-America/Port-au-Prince, GMT+01:00-CET-Europe/Budapest, GMT+07:00-WIT-Asia/Jakarta, GMT+07:00-WIT-Asia/Pontianak, GMT+08:00-CIT-Asia/Makassar, GMT+09:00-EIT-Asia/Jayapura, GMT+00:00-GMT-Europe/Dublin, GMT+02:00-IST-Asia/Jerusalem, GMT+00:00-GMT-Europe/Isle_of_Man, GMT+05:30-IST-Asia/Kolkata, GMT+06:00-IOT-Indian/Chagos, GMT+03:00-AST-Asia/Baghdad, GMT+03:30-IRST-Asia/Tehran, GMT+00:00-GMT-Atlantic/Reykjavik, GMT+01:00-CET-Europe/Rome, GMT+00:00-GMT-Europe/Jersey, GMT-05:00-EST-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Asia/Tokyo, GMT+03:00-EAT-Africa/Nairobi, GMT+06:00-KGT-Asia/Bishkek, GMT+07:00-ICT-Asia/Phnom_Penh, GMT+12:00-GILT-Pacific/Tarawa, GMT+13:00-PHOT-Pacific/Enderbury, GMT+14:00-LINT-Pacific/Kiritimati, GMT+03:00-EAT-Indian/Comoro, GMT-04:00-AST-America/St_Kitts, GMT+09:00-KST-Asia/Pyongyang, GMT+09:00-KST-Asia/Seoul, GMT+03:00-AST-Asia/Kuwait, GMT-05:00-EST-America/Cayman, GMT+06:00-ALMT-Asia/Almaty, GMT+06:00-QYZT-Asia/Qyzylorda, GMT+05:00-AQTT-Asia/Aqtobe, GMT+05:00-AQTT-Asia/Aqtau, GMT+05:00-ORAT-Asia/Oral, GMT+07:00-ICT-Asia/Vientiane, GMT+02:00-EET-Asia/Beirut, GMT-04:00-AST-America/St_Lucia, GMT+01:00-CET-Europe/Vaduz, GMT+05:30-IST-Asia/Colombo, GMT+00:00-GMT-Africa/Monrovia, GMT+02:00-SAST-Africa/Maseru, GMT+02:00-EET-Europe/Vilnius, GMT+01:00-CET-Europe/Luxembourg, GMT+02:00-EET-Europe/Riga, GMT+02:00-EET-Africa/Tripoli, GMT+00:00-WET-Africa/Casablanca, GMT+01:00-CET-Europe/Monaco, GMT+02:00-EET-Europe/Chisinau, GMT+01:00-CET-Europe/Podgorica, GMT-04:00-AST-America/Marigot, GMT+03:00-EAT-Indian/Antananarivo, GMT+12:00-MHT-Pacific/Majuro, GMT+12:00-MHT-Pacific/Kwajalein, GMT+01:00-CET-Europe/Skopje, GMT+00:00-GMT-Africa/Bamako, GMT+06:30-MMT-Asia/Rangoon, GMT+08:00-ULAT-Asia/Ulaanbaatar, GMT+07:00-HOVT-Asia/Hovd, GMT+08:00-CHOT-Asia/Choibalsan, GMT+08:00-CST-Asia/Macau, GMT+10:00-ChST-Pacific/Saipan, GMT-04:00-AST-America/Martinique, GMT+00:00-GMT-Africa/Nouakchott, GMT-04:00-AST-America/Montserrat, GMT+01:00-CET-Europe/Malta, GMT+04:00-MUT-Indian/Mauritius, GMT+05:00-MVT-Indian/Maldives, GMT+02:00-CAT-Africa/Blantyre, GMT-06:00-CST-America/Mexico_City, GMT-06:00-CST-America/Cancun, GMT-06:00-CST-America/Merida, GMT-06:00-CST-America/Monterrey, GMT-05:00-CDT-America/Matamoros, GMT-07:00-MST-America/Mazatlan, GMT-07:00-MST-America/Chihuahua, GMT-06:00-MDT-America/Ojinaga, GMT-07:00-MST-America/Hermosillo, GMT-07:00-PDT-America/Tijuana, GMT-08:00-PST-America/Santa_Isabel, GMT-06:00-CST-America/Bahia_Banderas, GMT+08:00-MYT-Asia/Kuala_Lumpur, GMT+08:00-MYT-Asia/Kuching, GMT+02:00-CAT-Africa/Maputo, GMT+02:00-WAST-Africa/Windhoek, GMT+11:00-NCT-Pacific/Noumea, GMT+01:00-WAT-Africa/Niamey, GMT+11:30-NFT-Pacific/Norfolk, GMT+01:00-WAT-Africa/Lagos, GMT-06:00-CST-America/Managua, GMT+01:00-CET-Europe/Amsterdam, GMT+01:00-CET-Europe/Oslo, GMT+05:45-NPT-Asia/Kathmandu, GMT+12:00-NRT-Pacific/Nauru, GMT-11:00-NUT-Pacific/Niue, GMT+13:00-NZDT-Pacific/Auckland, GMT+13:45-CHADT-Pacific/Chatham, GMT+04:00-GST-Asia/Muscat, GMT-05:00-EST-America/Panama, GMT-05:00-PET-America/Lima, GMT-10:00-TAHT-Pacific/Tahiti, GMT-09:30-MART-Pacific/Marquesas, GMT-09:00-GAMT-Pacific/Gambier, GMT+10:00-PGT-Pacific/Port_Moresby, GMT+08:00-PHT-Asia/Manila, GMT+05:00-PKT-Asia/Karachi, GMT+01:00-CET-Europe/Warsaw, GMT-02:00-PMDT-America/Miquelon, GMT-08:00-PST-Pacific/Pitcairn, GMT-04:00-AST-America/Puerto_Rico, GMT+02:00-EET-Asia/Gaza, GMT+02:00-EET-Asia/Hebron, GMT+00:00-WET-Europe/Lisbon, GMT+00:00-WET-Atlantic/Madeira, GMT-01:00-AZOT-Atlantic/Azores, GMT+09:00-PWT-Pacific/Palau, GMT-03:00-PYST-America/Asuncion, GMT+03:00-AST-Asia/Qatar, GMT+04:00-RET-Indian/Reunion, GMT+02:00-EET-Europe/Bucharest, GMT+01:00-CET-Europe/Belgrade, GMT+03:00-FET-Europe/Kaliningrad, GMT+04:00-MSK-Europe/Moscow, GMT+04:00-VOLT-Europe/Volgograd, GMT+04:00-SAMT-Europe/Samara, GMT+06:00-YEKT-Asia/Yekaterinburg, GMT+07:00-OMST-Asia/Omsk, GMT+07:00-NOVT-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Asia/Krasnoyarsk, GMT+09:00-IRKT-Asia/Irkutsk, GMT+10:00-YAKT-Asia/Yakutsk, GMT+11:00-VLAT-Asia/Vladivostok, GMT+11:00-SAKT-Asia/Sakhalin, GMT+12:00-MAGT-Asia/Magadan, GMT+12:00-PETT-Asia/Kamchatka, GMT+12:00-ANAT-Asia/Anadyr, GMT+02:00-CAT-Africa/Kigali, GMT+03:00-AST-Asia/Riyadh, GMT+11:00-SBT-Pacific/Guadalcanal, GMT+04:00-SCT-Indian/Mahe, GMT+03:00-EAT-Africa/Khartoum, GMT+01:00-CET-Europe/Stockholm, GMT+08:00-SGT-Asia/Singapore, GMT+00:00-GMT-Atlantic/St_Helena, GMT+01:00-CET-Europe/Ljubljana, GMT+01:00-CET-Arctic/Longyearbyen, GMT+01:00-CET-Europe/Bratislava, GMT+00:00-GMT-Africa/Freetown, GMT+01:00-CET-Europe/San_Marino, GMT+00:00-GMT-Africa/Dakar, GMT+03:00-EAT-Africa/Mogadishu, GMT-03:00-SRT-America/Paramaribo, GMT+00:00-GMT-Africa/Sao_Tome, GMT-06:00-CST-America/El_Salvador, GMT+02:00-EET-Asia/Damascus, GMT+02:00-SAST-Africa/Mbabane, GMT-04:00-EDT-America/Grand_Turk, GMT+01:00-WAT-Africa/Ndjamena, GMT+05:00-TFT-Indian/Kerguelen, GMT+00:00-GMT-Africa/Lome, GMT+07:00-ICT-Asia/Bangkok, GMT+05:00-TJT-Asia/Dushanbe, GMT-10:00-TKT-Pacific/Fakaofo, GMT+09:00-TLT-Asia/Dili, GMT+05:00-TMT-Asia/Ashgabat, GMT+01:00-CET-Africa/Tunis, GMT+13:00-TOT-Pacific/Tongatapu, GMT+02:00-EET-Europe/Istanbul, GMT-04:00-AST-America/Port_of_Spain, GMT+12:00-TVT-Pacific/Funafuti, GMT+08:00-CST-Asia/Taipei, GMT+03:00-EAT-Africa/Dar_es_Salaam, GMT+02:00-EET-Europe/Kiev, GMT+02:00-EET-Europe/Uzhgorod, GMT+02:00-EET-Europe/Zaporozhye, GMT+02:00-EET-Europe/Simferopol, GMT+03:00-EAT-Africa/Kampala, GMT-10:00-HST-Pacific/Johnston, GMT-11:00-SST-Pacific/Midway, GMT+12:00-WAKT-Pacific/Wake, GMT-04:00-EDT-America/New_York, GMT-04:00-EDT-America/Detroit, GMT-04:00-EDT-America/Kentucky/Louisville, GMT-04:00-EDT-America/Kentucky/Monticello, GMT-04:00-EDT-America/Indiana/Indianapolis, GMT-04:00-EDT-America/Indiana/Vincennes, GMT-04:00-EDT-America/Indiana/Winamac, GMT-04:00-EDT-America/Indiana/Marengo, GMT-04:00-EDT-America/Indiana/Petersburg, GMT-04:00-EDT-America/Indiana/Vevay, GMT-05:00-CDT-America/Chicago, GMT-05:00-CDT-America/Indiana/Tell_City, GMT-05:00-CDT-America/Indiana/Knox, GMT-05:00-CDT-America/Menominee, GMT-05:00-CDT-America/North_Dakota/Center, GMT-05:00-CDT-America/North_Dakota/New_Salem, GMT-05:00-CDT-America/North_Dakota/Beulah, GMT-06:00-MDT-America/Denver, GMT-06:00-MDT-America/Boise, GMT-06:00-MDT-America/Shiprock, GMT-07:00-MST-America/Phoenix, GMT-07:00-PDT-America/Los_Angeles, GMT-08:00-AKDT-America/Anchorage, GMT-08:00-AKDT-America/Juneau, GMT-08:00-AKDT-America/Sitka, GMT-08:00-AKDT-America/Yakutat, GMT-08:00-AKDT-America/Nome, GMT-09:00-HADT-America/Adak, GMT-08:00-MeST-America/Metlakatla, GMT-10:00-HST-Pacific/Honolulu, GMT-03:00-UYT-America/Montevideo, GMT+05:00-UZT-Asia/Samarkand, GMT+05:00-UZT-Asia/Tashkent, GMT+01:00-CET-Europe/Vatican, GMT-04:00-AST-America/St_Vincent, GMT-04:30-VET-America/Caracas, GMT-04:00-AST-America/Tortola, GMT-04:00-AST-America/St_Thomas, GMT+07:00-ICT-Asia/Ho_Chi_Minh, GMT+11:00-VUT-Pacific/Efate, GMT+12:00-WFT-Pacific/Wallis, GMT+14:00-WSDT-Pacific/Apia, GMT+03:00-AST-Asia/Aden, GMT+03:00-EAT-Indian/Mayotte, GMT+02:00-SAST-Africa/Johannesburg, GMT+02:00-CAT-Africa/Lusaka, GMT+02:00-CAT-Africa/Harare
			The percentage of shared quota to be granted at a time for maxClient. Default value: 10

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum value = 100
exclusivequotamaxclient	<Double>	Read-write	The percentage of maxClient to be given to PEs. Default value: 80 Minimum value = 0 Maximum value = 100
grantquotaspillover	<Double>	Read-write	The percentage of shared quota to be granted at a time for spillover. Default value: 10 Minimum value = 0 Maximum value = 100
exclusivequotaspillover	<Double>	Read-write	The percentage of max limit to be given to PEs. Default value: 80 Minimum value = 0 Maximum value = 100
config1	<String>	Read-write	Location of the configurations.
config2	<String>	Read-write	Location of the configurations.
outtype	<String>	Read-write	Format to display the difference in configurations. Possible values = cli, xml
template	<Boolean>	Read-write	File that contains the commands to be compared.
ignoredevicespecific	<Boolean>	Read-write	Suppress device specific differences.
weakpassword	<Boolean>	Read-write	Option to list all weak passwords (not adhering to strong password requirements). Takes config file as input, if no input specified, running configuration is considered. Command => query ns config -weakpassword / query ns config -weakpassword /nsconfig/ns.conf.
config	<String>	Read-write	configuration File to be used to find weak passwords, if not specified, running config is taken as input.
message	<String>	Read-only	.
mappedip	<String>	Read-only	Mapped IP Address of the System. Minimum length = 1
range	<Double>	Read-only	The range of Mapped IP addresses to be configured. Minimum value = 1
systemtype	<String>	Read-only	The type of the System. Possible Values: Standalone, HA, Cluster. Possible values = Stand-alone, HA, Cluster
primaryip	<String>	Read-only	HA Master Node IP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
lastconfigchangedtime	<String>	Read-only	Time when the configuration was last modified.
lastconfigsavetime	<String>	Read-only	Time when the configuration was last saved through savensconfig.
currentsytemtime	<String>	Read-only	current system time in date format.
systemtime	<Double>	Read-only	current system time.
configchanged	<Boolean>	Read-only	returns True if configuration has changed since last saved config.
response	<String>	Read-only	.

Operations (click to see Properties)

CLEAR | UPDATE | UNSET | SAVE | DIFF | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconfig?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsconfig":{
    "force":<Boolean_value>,
    "level":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconfig

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsconfig":{
    "ipaddress":<String_value>,
    "netmask":<String_value>,
    "nsvlan":<Double_value>,
    "ifnum":<String[]_value>,
    "tagged":<String_value>,
    "httpport":<Integer[]_value>,
    "maxconn":<Double_value>,
    "maxreq":<Double_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "cookieversion":<String_value>,
    "securecookie":<String_value>,
    "pmtumin":<Double_value>,
    "pmtutimeout":<Double_value>,
    "ftpportrange":<String_value>,
    "crportrange":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "timezone":<String_value>,
    "grantquotamaxclient":<Double_value>,
    "exclusivequotamaxclient":<Double_value>,
    "grantquotaspillover":<Double_value>,
    "exclusivequotaspillover":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconfig?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsconfig":{
    "nsvlan":true,
    "ftpportrange":true,
    "crportrange":true,
    "timezone":true,
    "ipaddress":true,
    "netmask":true,
    "ifnum":true,
    "tagged":true,
    "httpport":true,
    "maxconn":true,
    "maxreq":true,
    "cip":true,
    "cipheader":true,
    "cookieversion":true,
    "securecookie":true,
    "pmtumin":true,
    "pmtutimeout":true,
    "grantquotamaxclient":true,
    "exclusivequotamaxclient":true,
    "grantquotaspillover":true,
    "exclusivequotaspillover":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

save

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconfig?**action=save**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsconfig":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

diff

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconfig?**action=diff**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsconfig":{
    "config1":<String_value>,
    "config2":<String_value>,
    "outtype":<String_value>,
    "template":<Boolean_value>,
    "ignoredevicespecific":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "nsconfig": [ {
      "response":<String_value>

}]}
```

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconfig

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsconfig": [ {
      "ipaddress":<String_value>,
      "netmask":<String_value>,
      "mappedip":<String_value>,
      "range":<Double_value>,
      "nsvlan":<Double_value>,
      "ifnum":<String[]_value>,
      "tagged":<String_value>,
      "httpport":<Integer[]_value>,
      "maxconn":<Double_value>,
      "maxreq":<Double_value>,
      "cip":<String_value>,
      "cipheader":<String_value>,
      "cookieversion":<String_value>,
      "securecookie":<String_value>,
      "failover":<String_value>,
      "systemtype":<String_value>,
      "primaryip":<String_value>,
      "pmtumin":<Double_value>,
      "pmtutimeout":<Double_value>,
      "ftpportrange":<String_value>,
      "crportrange":<String_value>,
      "flags":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "timezone":<String_value>,
    "lastconfigchangedtime":<String_value>,
    "lastconfigsavetime":<String_value>,
    "currentsytemtime":<String_value>,
    "systemtime":<Double_value>,
    "grantquotamaxclient":<Double_value>,
    "exclusivequotamaxclient":<Double_value>,
    "grantquotaspillover":<Double_value>,
    "exclusivequotaspillover":<Double_value>,
    "configchanged":<Boolean_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsconnectiontable

May 20, 2016

Configuration for TCP/IP connection table resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
			The maximum length of filter expression is 255 and it can be of following format: <expression> [<relop> <expression>] <relop> = (&& | ||) connectiontable supports two types of filter expressions: Classic Expressions: [Note: Classic Expressions are not supported in non-default partitions] <expression> = the expression string in the format: <qualifier> <operator> <qualifier-value> <qualifier> = SOURCEIP. <qualifier-value> = A valid IP address. <qualifier> = SOURCEPORT. <qualifier-value> = A valid port number. <qualifier> = DESTIP. <qualifier-value> = A valid IP address. <qualifier> = DESTPORT. <qualifier-value> = A valid port number. <qualifier> = IP. <qualifier-value> = A valid IP address. <qualifier> = PORT. <qualifier-value> = A valid port number. <qualifier> = IDLETIME. <qualifier-value> = A positive integer indicating the idle time. <qualifier> = SVCNAME. <qualifier-value> = The name of a service. <qualifier> = VSVRNAME. <qualifier-value> = The name of a vserver. <qualifier> = CONNID <qualifier-value> = A valid PCB dev number. <qualifier> = INTF <qualifier-value> = A valid interface id in the form of x/y (n/x/y in case of cluster interface). <qualifier> = VLAN <qualifier-value> = A valid VLAN ID. <qualifier> = STATE. <qualifier-value> = (CLOSE_WAIT | CLOSED | CLOSING | ESTABLISHED | FIN_WAIT_1 | FIN_WAIT_2 | LAST_ACK | LISTEN | SYN_RECEIVED | SYN_SENT | TIME_WAIT) <qualifier> = SVCTYPE. <qualifier-value> = (HTTP | FTP | TCP | UDP | SSL | SSL_BRIDGE | SSL_TCP | NNTP | RPCSVR | RPCSVRS | RPCCLNT | DNS | ADNS | SNMP | RTSP | DHCPRA | ANY | MONITOR | MONITOR_UDP | MONITOR_PING | SIP_UDP | MYSQL | MSSQL | UNKNOWN) <operator> = (== | eq | != | neq | > | gt | < | lt | >= | ge | <= | le | BETWEEN) Default Expressions: <expression> =: CONNECTION.<qualifier>.<qualifier-method>.(<qualifier-value>) <qualifier> = SRCIP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address example = CONNECTION.SRCIP.EQ(127.0.0.1) <qualifier> = DSTIP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address. example = CONNECTION.DSTIP.EQ(127.0.0.1) <qualifier> = IP <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv4 address. example = CONNECTION.IP.EQ(127.0.0.1)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
filterexpression	<String>	Read-write	<qualifier> = SRCIPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.SRCIPv6.EQ(2001:db8:0:0:1::1) <qualifier> = DSTIPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.DSTIPv6.EQ(2001:db8:0:0:1::1) <qualifier> = IPv6 <qualifier-method> = [EQ | NE] <qualifier-value> = A valid IPv6 address. example = CONNECTION.IPv6.EQ(2001:db8:0:0:1::1) <qualifier> = SRCPORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.SRCPORT.EQ(80) <qualifier> = DSTPORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.DSTPORT.EQ(80) <qualifier> = PORT <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid port number. example = CONNECTION.PORT.EQ(80) <qualifier> = SVCNAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = service name. example = CONNECTION.SVCNAME.EQ("name") <qualifier> = LB_VSERVER.NAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = LB vserver name. example = CONNECTION.LB_VSERVER.NAME.EQ("name") <qualifier> = CS_VSERVER.NAME <qualifier-method> = [EQ | NE | CONTAINS | STARTSWITH | ENDSWITH] <qualifier-value> = CS vserver name. example = CONNECTION.CS_VSERVER.NAME.EQ("name") <qualifier> = INTF <qualifier-method> = [EQ | NE] <qualifier-value> = A valid interface id in the form of x/y (n/x/y in case of cluster interface). examle = CONNECTION.INTF.EQ("0/1/1") <qualifier> = VLANID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid VLAN ID. example = CONNECTION.VLANID.EQ(0) <qualifier> = CONNID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid PCB dev number. example = CONNECTION.CONNID.EQ(0) <qualifier> = PPEID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid core ID. example = CONNECTION.PPEID.EQ(0) <qualifier> = IDLETIME <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A positive integer indicating the idletime. example = CONNECTION.IDLETIME.LT(100) <qualifier> = TCPSTATE <qualifier-method> = [EQ | NE]

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			ESTABLISHED | FIN_WAIT_1 | FIN_WAIT_2 | LAST_ACK | LISTEN | SYN_RECEIVED | SYN_SENT | TIME_WAIT | NOT_APPLICABLE) example = CONNECTION.TCPSTATE.EQ(LISTEN) <qualifier> = SERVICE_TYPE <qualifier-method> = [EQ | NE] <qualifier-value> = (SVC_HTTP | FTP | TCP | UDP | SSL | SSL_BRIDGE | SSL_TCP | NNTP | RPCSVR | RPCSVRS | RPCCLNT | SVC_DNS | ADNS | SNMP | RTSP | DHCPRA | ANY| MONITOR | MONITOR_UDP | MONITOR_PING | SIP_UDP | SVC_MYSQL | SVC_MSSQL | SERVICE_UNKNOWN) example = CONNECTION.SERVICE_TYPE.EQ(ANY) <qualifier> = TRAFFIC_DOMAIN_ID <qualifier-method> = [EQ | NE | GT | GE | LT | LE | BETWEEN] <qualifier-value> = A valid traffic domain ID. example = CONNECTION.TRAFFIC_DOMAIN_ID.EQ(0) common usecases: Filtering out loopback connections and view present connections through netsclaer show connectiontable "CONNECTION.IP.NEQ(127.0.0.1) && CONNECTION.TCPSTATE.EQ(ESTABLISHED)" -detail full show connections from a particular sourceip and targeted to port 80 show connectiontable "CONNECTION.SRCIP.EQ(10.102.1.91) && CONNECTION.DSTPORT.EQ(80)" show connection particular to a service and its linked client connections show connectiontable "CONNECTION.SVCNAME.EQ("S1")" -detail link show connections for a particular servicetype(e.g.http) show connectiontable "CONNECTION.SERVICE_TYPE.EQ(TCP)" viewing connections that have been idle for a long time show connectiontable "CONNECTION.IDLETIME.GT(100)" show connections for a particular interface and vlan show connectiontable "CONNECTION.INTF.EQ("1/1") && CONNECTION.VLANID.EQ(1)" .
link	<Boolean>	Read-write	Display link information if available.
name	<Boolean>	Read-write	Display name instead of IP for local entities.
detail	<String[]>	Read-write	Specify display options for the connection table. * LINK - Displays the linked PCB (Protocol Control Block). * NAME - Displays along with the service name. * CONNFAILOVER - Displays PCB with connection failover. * FULL - Displays all available details. Possible values = LINK, NAME, CONNFAILOVER, FULL, NNM
listen	<Boolean>	Read-write	Display listening services only.
sourceip	<String>	Read-only	Source IP of the connection.
sourceport	<Integer>	Read-only	Source port of the connection. Range 1 - 65535
destip	<String>	Read-only	Destination IP of the connection.
destport	<Integer>	Read-only	Destination port of the connection. Range 1 - 65535
svctype	<String>	Read-only	Protocol supported by the connection. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, HTTPSVR, HTTPCLIENT, NAT, HA, AAA, SINCTCP, VPNAFTP, MONITORS, SSLVPNUDP, SINCUDP, RIP, UDPCLNT, SASP, RPCCLNTS, ROUTE, AUDIT, STA HTTP, STA SSL, DNS RESOLVE, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, ICA, RADIUS, RADIUSListener, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, TFTP_DATA
idletime	<Integer>	Read-only	Time since last activity was detected on the connection.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = CLOSED, LISTEN, SYN_SENT, SYN_RECEIVED, ESTABLISHED, CLOSE_WAIT, FIN_WAIT_1, CLOSING, LAST_ACK, FIN_WAIT_2, TIME_WAIT, NA
linksourceip	<String>	Read-only	Source IP of the link connection.
linksourceport	<Integer>	Read-only	Source port of the link connection. Range 1 - 65535
linkdestip	<String>	Read-only	Destination IP of the link connection.
linkdestport	<Integer>	Read-only	Destination port of the link connection. Range 1 - 65535
linkservicetype	<String>	Read-only	Protocol supported by the link connection. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, HTTPSVR, HTTPCLIENT, NAT, HA, AAA, SINCTCP, VPNAFTP, MONITORS, SSLVPNUDP, SINCUDP, RIP, UDPCLNT, SASP, RPCCLNTS, ROUTE, AUDIT, STA HTTP, STA SSL, DNS RESOLVE, RDP, MYSQL, MSSQL, DIAMETER, SSL_DIAMETER, TFTP, ORACLE, ICA, RADIUS, RADIUSListener, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX, TFTP_DATA
linkidletime	<Integer>	Read-only	Time since last activity was detected on link connection.
linkstate	<String>	Read-only	TCP/IP current state of link connection. Possible values = CLOSED, LISTEN, SYN_SENT, SYN_RECEIVED, ESTABLISHED, CLOSE_WAIT, FIN_WAIT_1, CLOSING, LAST_ACK, FIN_WAIT_2, TIME_WAIT, NA
entityname	<String>	Read-only	NetScaler entity name for the connection.
linkentityname	<String>	Read-only	NetScaler entity name for link connection.
connid	<Double>	Read-only	Unique transaction number for the connection.
linkconnid	<Double>	Read-only	Unique transaction number for the peer connection.
connproperties	<String[]>	Read-only	flags used to store connection properties like client, server etc. Possible values = LINK, HASLINK, CLIENT, SERVER, NNM, MPTCP, SUBFLOW
optionflags	<String[]>	Read-only	flags used to store TCP options like Sack, WS. Possible values = sack, timstmp, ws
nswsvalue	<Double>	Read-only	netscaler window scaling value.
peerwsvalue	<Double>	Read-only	peer window scaling value.
mss	<Integer>	Read-only	Client side MSS for the connection - used in server SYN.
retxretrycnt	<Integer>	Read-only	Retransmission retry count for the connection.
rcvwnd	<Integer>	Read-only	Received Advertised Window for the connection.
advwnd	<Integer>	Read-only	Sent advertised window for the connection.
sndcwnd	<Integer>	Read-only	sent congestion window for the connection.
iss	<Integer>	Read-only	Initial send sequence number for the connection.
irs	<Integer>	Read-only	Initial receive sequence number for the connection.
rcvnxt	<Integer>	Read-only	next expecting seq number for the connection.
maxack	<Integer>	Read-only	current running max ack sent for the connection.
sndnxt	<Integer>	Read-only	next bytes seq number for the connection.
sndunack	<Integer>	Read-only	Most recently received ACK for the connection.
httpendseq	<Integer>	Read-only	HTTP parsing tracking seq number for the connection.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Possible values = INITIALIZED, CONTENT_LENGTH, CHUNKED, WAIT_FINAL_ACK, UNKNOWN, DONE, WAIT_FIN
Name	**Data Type**	**Permissions**	**Description**
trcount	<Integer>	Read-only	Max reuests allowed per connection.
priority	<String>	Read-only	priority of the connection. Possible values = SC Priority, Priority queue1, priority queue2, priority queue3, DoS Priority, Surge Priority
httpreqver	<String>	Read-only	current HTTP request version on the connection. Possible values = HTTP_1_0, HTTP_1_1, HTTP_0_9, RTSP_1_0, SIP_2_0, VPN_ICA_SOCKS, VPN_ICA_CGP
httprequest	<String>	Read-only	current HTTP request type on the connection. Possible values = OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT, RPCCONNECT
httprspcode	<Integer>	Read-only	current response type on the connection.
rttsmoothed	<Integer>	Read-only	smoothed RTT value of the connection.
rttvariance	<Integer>	Read-only	RTT variance for the connection.
outoforderpkts	<Integer>	Read-only	held packets on the connection.
linkoptionflag	<String[]>	Read-only	Link connection's TCP option flag for Sack and WS. Possible values = sack, timstmp, ws
linknswsvalue	<Double>	Read-only	Link connection-s netscaler window scaling value.
linkpeerwsvalue	<Double>	Read-only	Link connection-s peer netscaler window scaling value.
targetnodeidnnm	<Integer>	Read-only	NNM connection target node ID.
sourcenodeidnnm	<Integer>	Read-only	NNM connection source node ID.
channelidnnm	<Double>	Read-only	NNM connection channel ID.
msgversionnnm	<Double>	Read-only	nnm message version.
td	<Double>	Read-only	Traffic Domain Id.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconnectiontable

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/nsconnectiontable?**args=filterexpression:<String_value>,link:<Boolean_value>,name:<Boolean_value>,detail:<String[]_value>,listen:<Boolean_value>**

Use this query-parameter to get nsconnectiontable resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsconnectiontable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsconnectiontable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsconnectiontable resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsconnectiontable?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsconnectiontable?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsconnectiontable resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsconnectiontable": [ {
filterexpression:<String_value>,link:<Boolean_value>,name:<Boolean_value>,detail:<String[]_value>,listen:<Boolean_value>      "sourceip":<String_value>,
      "sourceport":<Integer_value>,
      "destip":<String_value>,
      "destport":<Integer_value>,
      "svctype":<String_value>,
      "idletime":<Integer_value>,
      "state":<String_value>,
      "linksourceip":<String_value>,
      "linksourceport":<Integer_value>,
      "linkdestip":<String_value>,
      "linkdestport":<Integer_value>,
      "linkservicetype":<String_value>,
      "linkidletime":<Integer_value>,
      "linkstate":<String_value>,
      "entityname":<String_value>,
      "linkentityname":<String_value>,
      "connectionnumber":<Integer_value>,
      "linkconnectionnumber":<Integer_value>,
      "connid":<Double_value>,
      "linkconnid":<Double_value>,
      "connproperties":<String[]_value>,
      "optionflags":<String[]_value>,
      "nswsvalue":<Double_value>,
      "peerwsvalue":<Double_value>,
      "mss":<Integer_value>,
      "retxretrycnt":<Integer_value>,
      "rcvwnd":<Integer_value>,
      "advwnd":<Integer_value>,
      "sndcwnd":<Integer_value>,
      "iss":<Integer_value>,
      "irs":<Integer_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "rcvnxt":<Integer_value>,
    "maxack":<Integer_value>,
    "sndnxt":<Integer_value>,
    "sndunack":<Integer_value>,
    "httpendseq":<Integer_value>,
    "httpstate":<String_value>,
    "trcount":<Integer_value>,
    "priority":<String_value>,
    "httpreqver":<String_value>,
    "httprequest":<String_value>,
    "httprspcode":<Integer_value>,
    "rttsmoothed":<Integer_value>,
    "rttvariance":<Integer_value>,
    "outoforderpkts":<Integer_value>,
    "linkoptionflag":<String[]_value>,
    "linknswsvalue":<Double_value>,
    "linkpeerwsvalue":<Double_value>,
    "targetnodeidnnm":<Integer_value>,
    "sourcenodeidnnm":<Integer_value>,
    "channelidnnm":<Double_value>,
    "msgversionnnm":<Double_value>,
    "td":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconnectiontable?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsconnectiontable": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsconsoleloginprompt

May 20, 2016

Configuration for 0 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
promptstring	<String>	Read-write	Console login prompt string.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconsoleloginprompt

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsconsoleloginprompt":{
    "promptstring":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconsoleloginprompt?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsconsoleloginprompt":{
    "promptstring":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsconsoleloginprompt

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsconsoleloginprompt": [ {
      "promptstring":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsdhcpip

May 20, 2016

Configuration for dhcp IP resource.

Operations (click to see Properties)

RELEASE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

release

URL: http://<netscaler-ip-address>/nitro/v1/config/nsdhcpip?**action=release**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"nsdhcpip":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsdhcpparams

May 20, 2016

Configuration for DHCP parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
dhcpclient	<String>	Read-write	Enables DHCP client to acquire IP address from the DHCP server in the next boot. When set to OFF, disables the DHCP client in the next boot. Default value: OFF Possible values = ON, OFF
saveroute	<String>	Read-write	DHCP acquired routes are saved on the NetScaler appliance. Default value: OFF Possible values = ON, OFF
ipaddress	<String>	Read-only	DHCP acquired IP.
netmask	<String>	Read-only	DHCP acquired Netmask.
hostrtgw	<String>	Read-only	DHCP acquired Gateway.
running	<Boolean>	Read-only	DHCP mode.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsdhcpparams

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsdhcpparams":{
    "dhcpclient":<String_value>,
    "saveroute":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsdhcpparams?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"nsdhcpparams":{
    "dhcpclient":true,
    "saveroute":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsdhcpparams

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsdhcpparams": [ {
    "dhcpclient":<String_value>,
    "ipaddress":<String_value>,
    "netmask":<String_value>,
    "hostrtgw":<String_value>,
    "running":<Boolean_value>,
    "saveroute":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsdiameter

May 20, 2016

Configuration for Diameter Parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
identity	<String>	Read-write	DiameterIdentity to be used by NS. DiameterIdentity is used to identify a Diameter node uniquely. Before setting up diameter configuration, Netscaler (as a Diameter node) MUST be assigned a unique DiameterIdentity. example => set ns diameter -identity netscaler.com Now whenever Netscaler system needs to use identity in diameter messages. It will use 'netscaler.com' as Origin-Host AVP as defined in RFC3588 . Minimum length = 1
realm	<String>	Read-write	Diameter Realm to be used by NS. example => set ns diameter -realm com Now whenever Netscaler system needs to use realm in diameter messages. It will use 'com' as Origin-Realm AVP as defined in RFC3588 . Minimum length = 1
servercloseepropagation	<String>	Read-write	when a Server connection goes down, whether to close the corresponding client connection if there were requests pending on the server. Default value: NO Possible values = YES, NO

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsdiameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsdiameter":{
    "identity":<String_value>,
    "realm":<String_value>,
    "serverclosepropagation":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsdiameter?**action=unset**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsdiameter":{
    "servercloseproagation":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsdiameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsdiameter": [ {
    "identity":<String_value>,
    "realm":<String_value>,
    "servercloseproagation":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsencryptionparams

May 20, 2016

Configuration for 0 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
method	<String>	Read-write	Cipher method (and key length) to be used to encrypt and decrypt content. The default value is AES256. Possible values = NONE, RC4, DES3, AES128, AES192, AES256
keyvalue	<String>	Read-write	The base64-encoded key generation number, method, and key value. Note: * Do not include this argument if you are changing the encryption method. * To generate a new key value for the current encryption method, specify an empty string \("\") as the value of this parameter. The parameter is passed implicitly, with its automatically generated value, to the NetScaler packet engines even when it is not included in the command. Passing the parameter to the packet engines enables the appliance to save the key value to the configuration file and to propagate the key value to the secondary appliance in a high availability setup.

Operations (click to see Properties)

UPDATE | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsencryptionparams

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsencryptionparams":{
    "method":<String_value>,
    "keyvalue":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsencryptionparams

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "nsencryptionparams": [ {
      "method":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsevents

Jul 07, 2017

Configuration for events resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
eventno	<Double>	Read-write	Event number starting from which events must be shown.
time	<Double>	Read-only	Event no.
eventcode	<Double>	Read-only	event Code.
devid	<Double>	Read-only	Device Name.
devname	<String>	Read-only	Device Name.
text	<String>	Read-only	Event no.
data0	<Double>	Read-only	additional event information.
data1	<Double>	Read-only	additional event information.
data2	<Double>	Read-only	additional event information.
data3	<Double>	Read-only	additional event information.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsevents

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/nsevents?**args=eventno:<Double_value>**

Use this query-parameter to get nsevents resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsevents?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsevents?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsevents resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nsevents?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsevents?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsevents resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsevents": [ {
eventno:<Double_value>      "time":<Double_value>,
      "eventcode":<Double_value>,
      "devid":<Double_value>,
      "devname":<String_value>,
      "text":<String_value>,
      "data0":<Double_value>,
      "data1":<Double_value>,
      "data2":<Double_value>,
      "data3":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsevents?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsevents": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsextension

May 20, 2016

Configuration for Extension resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
src	<String>	Read-write	Local path to and name of, or URL (protocol, host, path, and file name) for, the file in which to store the imported extension. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
name	<String>	Read-write	Name to assign to the extension object on the NetScaler appliance. Minimum length = 1 Maximum length = 31
comment	<String>	Read-write	Any comments to preserve information about the extension object. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrites the existing file.
trace	<String>	Read-write	Enables tracing to the NS log file of extension execution: off - turns off tracing (equivalent to unset ns extension <extension-name> -trace) calls - traces extension function calls with arguments and function returns with the first return value lines - traces the above plus line numbers for executed extension lines all - traces the above plus local variables changed by executed extension lines Note that the DEBUG log level must be enabled to see extension tracing. This can be done by set audit syslogParams -loglevel ALL or -loglevel DEBUG. Default value: off Possible values = off, calls, lines, all
tracefunctions	<String>	Read-write	Comma-separated list of extension functions to trace. By default, all extension functions are traced. Maximum length = 256
tracevariables	<String>	Read-write	Comma-separated list of variables (in traced extension functions) to trace. By default, all variables are traced. Maximum length = 256
detail	<String>	Read-write	Show detail for extension function. Possible values = brief, all

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = WSDL, CustomSettings, XMLSchema, XMLErrorPage, htmlpage, CustomResp, Extension
functionhits	<Double>	Read-only	Number of time function evaluates successfully.
functionundefhits	<Double>	Read-only	Number of times error occured in evaluating extension function.
functionhaltcount	<Double>	Read-only	Number of time function evaluation is halted.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

IMPORT | DELETE | ADD | CHANGE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension?**action=Import**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsextension":{
    "src":<String_value>,
    "name":<String_value>,
    "comment":<String_value>,
    "overwrite":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"nsextension":{
    "name":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

change

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension?**action=update**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsextension":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsextension":{
    "name":<String_value>,
    "trace":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "tracefunctions":<String_value>,
    "tracevariables":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsextension":{
    "name":<String_value>,
    "trace":true,
    "tracefunctions":true,
    "tracevariables":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/nsextension?**args=name:<String_value>,detail:<String_value>**

Use this query-parameter to get nsextension resources based on additional properties.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsextension?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsextension?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsextension resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsextension?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsextension?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsextension resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsextension": [ {
name:<String_value>,detail:<String_value>      "src":<String_value>,
      "type":<String_value>,
      "comment":<String_value>,
      "functionhits":<Double_value>,
      "functionundefhits":<Double_value>,
      "functionhaltcount":<Double_value>,
      "trace":<String_value>,
      "tracefunctions":<String_value>,
      "tracevariables":<String_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsextension/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nsextension/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsextension": [ {
name:<String_value>,detail:<String_value>      "src":<String_value>,
      "type":<String_value>,
      "comment":<String_value>,
      "functionhits":<Double_value>,
      "functionundefhits":<Double_value>,
      "functionhaltcount":<Double_value>,
      "trace":<String_value>,
      "tracefunctions":<String_value>,
      "tracevariables":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "nsextension": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsip

Jul 07, 2017

Configuration for ip resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipaddress	<String>	Read-write	IPv4 address to create on the NetScaler appliance. Cannot be changed after the IP address is created. Minimum length = 1
netmask	<String>	Read-write	Subnet mask associated with the IP address.
type	<String>	Read-write	Type of the IP address to create on the NetScaler appliance. Cannot be changed after the IP address is created. The following are the different types of NetScaler owned IP addresses: * A Subnet IP (SNIP) address is used by the NetScaler ADC to communicate with the servers. The NetScaler also uses the subnet IP address when generating its own packets, such as packets related to dynamic routing protocols, or to send monitor probes to check the health of the servers. * A Virtual IP (VIP) address is the IP address associated with a virtual server. It is the IP address to which clients connect. An appliance managing a wide range of traffic may have many VIPs configured. Some of the attributes of the VIP address are customized to meet the requirements of the virtual server. * A GSLB site IP (GSLBIP) address is associated with a GSLB site. It is not mandatory to specify a GSLBIP address when you initially configure the NetScaler appliance. A GSLBIP address is used only when you create a GSLB site. * A Cluster IP (CLIP) address is the management address of the cluster. All cluster configurations must be performed by accessing the cluster through this IP address. Default value: SNIP Possible values = SNIP, VIP, NSIP, GSLBsiteIP, CLIP
arp	<String>	Read-write	Respond to ARP requests for this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
icmp	<String>	Read-write	Respond to ICMP requests for this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
			Use this option to set (enable or disable) the virtual server attribute for this IP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = ENABLED, DISABLED
telnet	<String>	Read-write	Allow Telnet access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
ftp	<String>	Read-write	Allow File Transfer Protocol (FTP) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
gui	<String>	Read-write	Allow graphical user interface (GUI) access to this IP address. Default value: ENABLED Possible values = ENABLED, SECUREONLY, DISABLED
ssh	<String>	Read-write	Allow secure shell (SSH) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
snmp	<String>	Read-write	Allow Simple Network Management Protocol (SNMP) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
mgmtaccess	<String>	Read-write	Allow access to management applications on this IP address. Default value: DISABLED Possible values = ENABLED, DISABLED
restrictaccess	<String>	Read-write	Block access to nonmanagement applications on this IP. This option is applicable for MIPs, SNIPs, and NSIP, and is disabled by default. Nonmanagement applications can run on the underlying NetScaler Free BSD operating system. Default value: DISABLED Possible values = ENABLED, DISABLED
dynamicrouting	<String>	Read-write	Allow dynamic routing on this IP address. Specific to Subnet IP (SNIP) address. Default value: DISABLED Possible values = ENABLED, DISABLED
ospf	<String>	Read-write	Use this option to enable or disable OSPF on this IP address for the entity. Default value: DISABLED Possible values = ENABLED, DISABLED
bgp	<String>	Read-write	Use this option to enable or disable BGP on this IP address for the entity. Default value: DISABLED Possible values = ENABLED, DISABLED
rip	<String>	Read-write	Use this option to enable or disable RIP on this IP address for the entity. Default value: DISABLED Possible values = ENABLED, DISABLED
hostroute	<String>	Read-write	Advertise a route for the VIP address using the dynamic routing protocols running on the NetScaler appliance. Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: -1
metric	<Integer>	Read-write	Integer value to add to or subtract from the cost of the route advertised for the VIP address. Minimum value = -16777215
vserverrhilevel	<String>	Read-write	Advertise the route for the Virtual IP (VIP) address on the basis of the state of the virtual servers associated with that VIP. * NONE - Advertise the route for the VIP address, regardless of the state of the virtual servers associated with the address. * ONE VSERVER - Advertise the route for the VIP address if at least one of the associated virtual servers is in UP state. * ALL VSERVER - Advertise the route for the VIP address if all of the associated virtual servers are in UP state. * VSVR_CNTRLD - Advertise the route for the VIP address according to the RHIstate (RHI STATE) parameter setting on all the associated virtual servers of the VIP address along with their states. When Vserver RHI Level (RHI) parameter is set to VSVR_CNTRLD, the following are different RHI behaviors for the VIP address on the basis of RHIstate (RHI STATE) settings on the virtual servers associated with the VIP address: * If you set RHI STATE to PASSIVE on all virtual servers, the NetScaler ADC always advertises the route for the VIP address. * If you set RHI STATE to ACTIVE on all virtual servers, the NetScaler ADC advertises the route for the VIP address if at least one of the associated virtual servers is in UP state. *If you set RHI STATE to ACTIVE on some and PASSIVE on others, the NetScaler ADC advertises the route for the VIP address if at least one of the associated virtual servers, whose RHI STATE set to ACTIVE, is in UP state. Default value: ONE_VSERVER Possible values = ONE_VSERVER, ALL_VSERVERS, NONE, VSVR_CNTRLD
vserverrhimode	<String>	Read-write	Advertise the route for the Virtual IP (VIP) address using dynamic routing protocols or using RISE * DYNMAIC_ROUTING - Advertise the route for the VIP address using dynamic routing protocols (default) * RISE - Advertise the route for the VIP address using RISE. Default value: DYNAMIC_ROUTING Possible values = DYNAMIC_ROUTING, RISE
ospflsatype	<String>	Read-write	Type of LSAs to be used by the OSPF protocol, running on the NetScaler appliance, for advertising the route for this VIP address. Default value: TYPE5 Possible values = TYPE1, TYPE5
ospfarea	<Double>	Read-write	ID of the area in which the type1 link-state advertisements (LSAs) are to be advertised for this virtual IP (VIP) address by the OSPF protocol running on the NetScaler appliance. When this parameter is not set, the VIP is advertised on all areas. Default value: -1 Minimum value = 0 Maximum value = 4294967294LU

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
state	<String>	Read-write	Default value: ENABLED Possible values = ENABLED, DISABLED
vrid	<Double>	Read-write	A positive integer that uniquely identifies a VMAC address for binding to this VIP address. This binding is used to set up NetScaler appliances in an active-active configuration using VRRP. Minimum value = 1 Maximum value = 255
icmpresponse	<String>	Read-write	Respond to ICMP requests for a Virtual IP (VIP) address on the basis of the states of the virtual servers associated with that VIP. Available settings function as follows: * NONE - The NetScaler appliance responds to any ICMP request for the VIP address, irrespective of the states of the virtual servers associated with the address. * ONE VSERVER - The NetScaler appliance responds to any ICMP request for the VIP address if at least one of the associated virtual servers is in UP state. * ALL VSERVER - The NetScaler appliance responds to any ICMP request for the VIP address if all of the associated virtual servers are in UP state. * VSVR_CNTRLD - The behavior depends on the ICMP VSERVER RESPONSE setting on all the associated virtual servers. The following settings can be made for the ICMP VSERVER RESPONSE parameter on a virtual server: * If you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, NetScaler always responds. * If you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds if even one virtual server is UP. * When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds if even one virtual server set to ACTIVE is UP. Default value: 5 Possible values = NONE, ONE_VSERVER, ALL_VSERVERS, VSVR_CNTRLD
ownernode	<Double>	Read-write	The owner node in a Cluster for this IP address. Owner node can vary from 0 to 31. If ownernode is not specified then the IP is treated as Striped IP. Default value: 255
arpresponse	<String>	Read-write	Respond to ARP requests for a Virtual IP (VIP) address on the basis of the states of the virtual servers associated with that VIP. Available settings function as follows: * NONE - The NetScaler appliance responds to any ARP request for the VIP address, irrespective of the states of the virtual servers associated with the address. * ONE VSERVER - The NetScaler appliance responds to any ARP request for the VIP address if at least one of the associated virtual servers is in UP state. * ALL VSERVER - The NetScaler appliance responds to any ARP request for the VIP address if all of the associated virtual servers are in UP state. Default value: 5 Possible values = NONE, ONE_VSERVER, ALL_VSERVERS
			Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 0 Maximum value = 4094
flags	<Double>	Read-only	The flags for this entry.
hostrtgwact	<String>	Read-only	Actual Gateway used for advertising host route.
ospfareaval	<Double>	Read-only	The area ID of the area in which OSPF Type1 LSAs are advertised.
viprtadv2bsd	<Boolean>	Read-only	Whether this route is advertised to FreeBSD.
vipvsercount	<Double>	Read-only	Number of vservers bound to this VIP.
vipvserdowncount	<Double>	Read-only	Number of vservers bound to this VIP, which are down.
vipvsrvrrhiactivecount	<Double>	Read-only	Number of vservers that have RHI state ACTIVE.
vipvsrvrrhiactiveupcount	<Double>	Read-only	Number of vservers that have RHI state ACTIVE, which are UP.
freeports	<Double>	Read-only	Number of free Ports available on this IP.
riserhimsgcode	<Integer>	Read-only	The code indicating the rise rhi status.
iptype	<String[]>	Read-only	. Possible values = SNIP, VIP, NSIP, GSLBsiteIP, CLIP
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsip":{
    "ipaddress":<String_value>,
    "netmask":<String_value>,
    "type":<String_value>,
    "arp":<String_value>,
    "icmp":<String_value>,
    "vserver":<String_value>,
    "telnet":<String_value>,
    "ftp":<String_value>,
    "gui":<String_value>,
    "ssh":<String_value>,
    "snmp":<String_value>,
    "mgmtaccess":<String_value>,
    "restrictaccess":<String_value>,
    "dynamicrouting":<String_value>,
    "ospf":<String_value>,
    "bgp":<String_value>,
    "rip":<String_value>,
    "hostroute":<String_value>,
    "hostrtgw":<String_value>,
    "metric":<Integer_value>,
    "vserverrhilevel":<String_value>,
    "vserverrhimode":<String_value>,
    "ospflsatype":<String_value>,
    "ospfarea":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "state":<String_value>,
    "vrid":<Double_value>,
    "icmpresponse":<String_value>,
    "ownernode":<Double_value>,
    "arpresponse":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip/ipaddress_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsip":{
    "ipaddress":<String_value>,
    "td":<Double_value>,
    "netmask":<String_value>,
    "arp":<String_value>,
    "icmp":<String_value>,
    "vserver":<String_value>,
    "telnet":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ftp":<String_value>,
    "gui":<String_value>,
    "ssh":<String_value>,
    "snmp":<String_value>,
    "mgmtaccess":<String_value>,
    "restrictaccess":<String_value>,
    "dynamicrouting":<String_value>,
    "ospf":<String_value>,
    "bgp":<String_value>,
    "rip":<String_value>,
    "hostroute":<String_value>,
    "hostrtgw":<String_value>,
    "metric":<Integer_value>,
    "vserverrhilevel":<String_value>,
    "vserverrhimode":<String_value>,
    "ospflsatype":<String_value>,
    "ospfarea":<Double_value>,
    "vrid":<Double_value>,
    "icmpresponse":<String_value>,
    "arpresponse":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsip":{
    "ipaddress":<String_value>,
    "td":<Double_value>,
    "ospfarea":true,
    "hostrtgw":true,
    "netmask":true,
    "arp":true,
    "icmp":true,
    "vserver":true,
    "telnet":true,
    "ftp":true,
    "gui":true,
    "ssh":true,
    "snmp":true,
    "mgmtaccess":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "restrictaccess":true,
        "dynamicrouting":true,
        "ospf":true,
        "bgp":true,
        "rip":true,
        "hostroute":true,
        "metric":true,
        "vserverrhilevel":true,
        "vserverrhimode":true,
        "ospflsatype":true,
        "vrid":true,
        "icmpresponse":true,
        "arpresponse":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsip":{
    "ipaddress":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip?**action=disable**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsip":{
    "ipaddress":<String_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/nsip?**args=ipaddress:<String_value>,td:<Double_value>,type:<String_value>**

Use this query-parameter to get nsip resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsip?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsip?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsip resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsip?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nsip?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsip resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsip": [ {
ipaddress:<String_value>,td:<Double_value>,type:<String_value>      "netmask":<String_value>,
      "flags":<Double_value>,
      "arp":<String_value>,
      "icmp":<String_value>,
      "vserver":<String_value>,
      "telnet":<String_value>,
      "ssh":<String_value>,
      "gui":<String_value>,
      "snmp":<String_value>,
      "ftp":<String_value>,
      "mgmtaccess":<String_value>,
      "restrictaccess":<String_value>,
      "dynamicrouting":<String_value>,
      "bgp":<String_value>,
      "ospf":<String_value>,
      "rip":<String_value>,
      "hostroute":<String_value>,
      "hostrtgw":<String_value>,
      "hostrtgwact":<String_value>,
      "metric":<Integer_value>,
      "ospfarea":<Double_value>,
      "ospfareaval":<Double_value>,
      "vserverrhilevel":<String_value>,
      "vserverrhimode":<String_value>,
      "viprtadv2bsd":<Boolean_value>,
      "vipvsercount":<Double_value>,
      "vipvserdowncount":<Double_value>,
      "vipvsrvrrhiactivecount":<Double_value>,
      "vipvsrvrrhiactiveupcount":<Double_value>,
      "ospflsatype":<String_value>,
      "state":<String_value>,
      "freeports":<Double_value>,
      "vrid":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "riserhimsgcode":<Integer_value>,
    "iptype":<String[]_value>,
    "icmpresponse":<String_value>,
    "ownernode":<Double_value>,
    "arpresponse":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsip": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsextension_binding

May 20, 2016

Binding object which returns the resources bound to nsextension.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the extension object. Minimum length = 1 Maximum length = 31
nsextension_extensionfunction_binding	<nsextension_extensionfunction_binding[]>	Read-only	extensionfunction that can be bound to nsextension.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "nsextension_binding": [ {
    "name":<String_value>,
    "nsextension_extensionfunction_binding":<nsextension_extensionfunction_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsfeature

Jul 07, 2017

Configuration for feature resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
feature	<String[]>	Read-write	Feature to be enabled. Multiple features can be specified by providing a blank space between each feature.
wl	<Boolean>	Read-only	Web Logging.
sp	<Boolean>	Read-only	Surge Protection.
lb	<Boolean>	Read-only	Load Balancing.
cs	<Boolean>	Read-only	Content Switching.
cr	<Boolean>	Read-only	Cache Redirect.
sc	<Boolean>	Read-only	Sure Connect.
cmp	<Boolean>	Read-only	Compression.
pq	<Boolean>	Read-only	Priority Queuing.
ssl	<Boolean>	Read-only	Secure Sockets Layer.
gslb	<Boolean>	Read-only	Global Server Load Balancing.
hdosp	<Boolean>	Read-only	DOS Protection.
cf	<Boolean>	Read-only	Content Filter.
ic	<Boolean>	Read-only	Integrated Caching.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sslvpn	<Boolean>	Read-only	SSL VPN.
aaa	<Boolean>	Read-only	AAA.
ospf	<Boolean>	Read-only	OSPF Routing.
rip	<Boolean>	Read-only	RIP Routing.
bgp	<Boolean>	Read-only	BGP Routing.
rewrite	<Boolean>	Read-only	Rewrite.
ipv6pt	<Boolean>	Read-only	IPv6 protocol translation.
appfw	<Boolean>	Read-only	Application Firewall.
responder	<Boolean>	Read-only	Responder.
htmlinjection	<Boolean>	Read-only	HTML Injection.
push	<Boolean>	Read-only	NetScaler Push.
appflow	<Boolean>	Read-only	AppFlow.
cloudbridge	<Boolean>	Read-only	CloudBridge.
isis	<Boolean>	Read-only	ISIS Routing.
ch	<Boolean>	Read-only	Call Home.
appqoe	<Boolean>	Read-only	AppQoS.
vpath	<Boolean>	Read-only	Vpath.
contentaccelerator	<Boolean>	Read-only	Transparent Integrated Caching.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
reo	<Boolean>	Read-only	Optimize Web content (HTML, CSS, JavaScript, images).
lsn	<Boolean>	Read-only	Large Scale NAT.
rdpproxy	<Boolean>	Read-only	RDPPROXY.
rep	<Boolean>	Read-only	IP Reputation.

Operations (click to see Properties)

ENABLE | DISABLE | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/nsfeature?**action=enable**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsfeature":{
    "feature":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/nsfeature?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsfeature":{
    "feature":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsfeature

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsfeature": [ {
      "feature":<String[]_value>,
      "wl":<Boolean_value>,
      "sp":<Boolean_value>,
      "lb":<Boolean_value>,
      "cs":<Boolean_value>,
      "cr":<Boolean_value>,
      "sc":<Boolean_value>,
      "cmp":<Boolean_value>,
      "pq":<Boolean_value>,
      "ssl":<Boolean_value>,
      "gslb":<Boolean_value>,
      "hdosp":<Boolean_value>,
      "routing":<Boolean_value>,
      "cf":<Boolean_value>,
      "ic":<Boolean_value>,
      "sslvpn":<Boolean_value>,
      "aaa":<Boolean_value>,
      "ospf":<Boolean_value>,
      "rip":<Boolean_value>,
      "bgp":<Boolean_value>,
      "rewrite":<Boolean_value>,
      "ipv6pt":<Boolean_value>,
      "appfw":<Boolean_value>,
      "responder":<Boolean_value>,
      "htmlinjection":<Boolean_value>,
      "push":<Boolean_value>,
      "appflow":<Boolean_value>,
      "cloudbridge":<Boolean_value>,
      "isis":<Boolean_value>,
      "ch":<Boolean_value>,
      "appqoe":<Boolean_value>,
      "vpath":<Boolean_value>,
      "contentaccelerator":<Boolean_value>,
      "rise":<Boolean_value>,
      "feo":<Boolean_value>,
      "lsn":<Boolean_value>,
      "rdpproxy":<Boolean_value>,
      "rep":<Boolean_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nshardware

Jul 07, 2017

Configuration for hardware resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
hwdescription	<String>	Read-only	Hardware and it's ports detail.
sysid	<Double>	Read-only	System id.
manufactureday	<Integer>	Read-only	Manufacturing day.
manufacturemonth	<Integer>	Read-only	Manufacturing month.
manufactureyear	<Integer>	Read-only	Manufacturing year.
cpufrequncy	<Integer>	Read-only	CPU Frequency.
hostid	<Integer>	Read-only	host id.
host	<String>	Read-only	host id.
serialno	<String>	Read-only	Serial no.
encodedserialno	<String>	Read-only	Encoded serial no.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nshardware

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nshardware": [ {
    "hwdescription":<String_value>,
    "sysid":<Double_value>,
    "manufactureday":<Integer_value>,
    "manufacturemonth":<Integer_value>,
    "manufactureyear":<Integer_value>,
    "cpufrequncy":<Integer_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "hostid":<Integer_value>,
    "host":<String_value>,
    "serialno":<String_value>,
    "encodedserialno":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsextension_extensionfunction_binding

May 20, 2016

Binding object showing the extensionfunction that can be bound to nsextension.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the extension object. Minimum length = 1 Maximum length = 31
extensionfunctionlinenumber	<Double>	Read-only	Line number of the function in file.
extensionfunctionclasses	<String[]>	Read-only	List of classes (including inherited) that the function is present in.
extensionfunctionargtype	<String[]>	Read-only	List of extension function's arguments types.
activeextensionfunction	<Integer>	Read-only	Extension function is in use or not.
extensionfunctionname	<String>	Read-only	Name of extension function given in the extension. Minimum length = 1 Maximum length = 31
extensionfunctionclasstype	<String>	Read-only	Extension function class type. Minimum length = 1 Maximum length = 31
extensionfuncdescription	<String>	Read-only	Any description to preserve information about the extension function. Maximum length = 1023
extensionfunctionallparams	<String[]>	Read-only	List of parameters (including promotions) that the function can accept.
extensionfunctionargcount	<Double>	Read-only	Number of parameters in the extension function.
extensionfunctionreturntype	<String>	Read-only	Extension function return type. Minimum length = 1 Maximum length = 31
extensionfunctionclassescount	<Double>	Read-only	Number of classes the function is present in.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
__count	<Double>	Read-only	accept count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension_extensionfunction_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/nsextension_extensionfunction_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nsextension_extensionfunction_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsextension_extensionfunction_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsextension_extensionfunction_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsextension_extensionfunction_binding": [ {
    "name":<String_value>,
    "extensionfunctionlinenumber":<Double_value>,
    "extensionfunctionclasses":<String[]_value>,
    "extensionfunctionargtype":<String[]_value>,
    "activeextensionfunction":<Integer_value>,
    "extensionfunctionname":<String_value>,
    "extensionfunctionclasstype":<String_value>,
    "extensionfuncdescription":<String_value>,
    "extensionfunctionallparams":<String[]_value>,
    "extensionfunctionargcount":<Double_value>,
    "extensionfunctionreturntype":<String_value>,
    "extensionfunctionclassescount":<Double_value>,
    "extensionfunctionallparamscount":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsextension_extensionfunction_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"nsextension_extensionfunction_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nshostname

May 20, 2016

Configuration for host name resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
hostname	<String>	Read-write	Host name for the NetScaler appliance. Minimum length = 1 Maximum length = 255
ownernode	<Double>	Read-write	ID of the cluster node for which you are setting the hostname. Can be configured only through the cluster IP address. Default value: 255 Minimum value = 0 Maximum value = 31
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

UPDATE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nshostname

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nshostname":{
    "hostname":<String_value>,
    "ownernode":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nshostname

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nshostname?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nshostname?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nshostname resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/nshostname?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nshostname?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nshostname resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nshostname": [ {
      "hostname":<String_value>,
      "ownernode":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nshostname?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "nshostname": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nshttpparam

May 20, 2016

Configuration for HTTP parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
dropinvalreqs	<String>	Read-write	Drop invalid HTTP requests or responses. Default value: OFF Possible values = ON, OFF
markhttp09inval	<String>	Read-write	Mark HTTP/0.9 requests as invalid. Default value: OFF Possible values = ON, OFF
markconnreqinval	<String>	Read-write	Mark CONNECT requests as invalid. Default value: OFF Possible values = ON, OFF
insnssrvrhdr	<String>	Read-write	Enable or disable NetScaler server header insertion for NetScaler generated HTTP responses. Default value: OFF Possible values = ON, OFF
nssrvrhdr	<String>	Read-write	The server header value to be inserted. If no explicit header is specified then NSBUILD.RELEASE is used as default server header. Minimum length = 1
logerrresp	<String>	Read-write	Server header value to be inserted. Default value: ON Possible values = ON, OFF
conmultiplex	<String>	Read-write	Reuse server connections for requests from more than one client connections. Default value: ENABLED Possible values = ENABLED, DISABLED
maxreusepool	<Double>	Read-write	Maximum limit on the number of connections, from the NetScaler to a particular server that are kept in the reuse pool. This setting is helpful for optimal memory utilization and for reducing the idle connections to the server just after the peak time. Minimum value = 0 Maximum value = 360000
builtin	<String[]>	Read-only	Flag to determine if the http param is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nshttpparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nshttpparam":{
    "dropinvalreqs":<String_value>,
    "markhttp09inval":<String_value>,
    "markconnreqinval":<String_value>,
    "insnssrvrhdr":<String_value>,
    "nssrvrhdr":<String_value>,
    "logerrresp":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "conmultiplex":<String_value>,
    "maxreusepool":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nshttpparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nshttpparam":{
    "dropinvalreqs":true,
    "markhttp09inval":true,
    "markconnreqinval":true,
    "insnssrvrhdr":true,
    "nssrvrhdr":true,
    "logerrresp":true,
    "conmultiplex":true,
    "maxreusepool":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nshttpparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nshttpparam": [ {
      "dropinvalreqs":<String_value>,
      "markhttp09inval":<String_value>,
      "markconnreqinval":<String_value>,
      "insnssrvrhdr":<String_value>,
      "nssrvrhdr":<String_value>,
      "logerrresp":<String_value>,
      "conmultiplex":<String_value>,
      "maxreusepool":<Double_value>,
      "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nshttpprofile

May 20, 2016

Configuration for HTTP profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for an HTTP profile. Must begin with a letter, number, or the underscore \(_\) character. Other characters allowed, after the first character, are the hyphen \(-\), period \(.\), hash \(\#\), space \(\), at \(@\), colon \(:\), and equal \(=\) characters. The name of a HTTP profile cannot be changed after it is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks \(for example, "my http profile" or 'my http profile'\). Minimum length = 1 Maximum length = 127
dropinvalreqs	<String>	Read-write	Drop invalid HTTP requests or responses. Default value: DISABLED Possible values = ENABLED, DISABLED
markhttp09inval	<String>	Read-write	Mark HTTP/0.9 requests as invalid. Default value: DISABLED Possible values = ENABLED, DISABLED
markconnreqinval	<String>	Read-write	Mark CONNECT requests as invalid. Default value: DISABLED Possible values = ENABLED, DISABLED
cmponpush	<String>	Read-write	Start data compression on receiving a TCP packet with PUSH flag set. Default value: DISABLED Possible values = ENABLED, DISABLED
conmultiplex	<String>	Read-write	Reuse server connections for requests from more than one client connections. Default value: ENABLED Possible values = ENABLED, DISABLED
maxreusepool	<Double>	Read-write	Maximum limit on the number of connections, from the NetScaler to a particular server that are kept in the reuse pool. This setting is helpful for optimal memory utilization and for reducing the idle connections to the server just after the peak time. Zero implies no limit on reuse pool size. Default value: 0 Minimum value = 0

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
dropextracrlf	<String>	Read-write	Drop any extra 'CR' and 'LF' characters present after the header. Default value: ENABLED Possible values = ENABLED, DISABLED
incomphdrdelay	<Double>	Read-write	Maximum time to wait, in milliseconds, between incomplete header packets. If the header packets take longer to arrive at NetScaler, the connection is silently dropped. Default value: 7000 Minimum value = 1 Maximum value = 360000
websocket	<String>	Read-write	HTTP connection to be upgraded to a web socket connection. Once upgraded, NetScaler does not process Layer 7 traffic on this connection. Default value: DISABLED Possible values = ENABLED, DISABLED
rtsptunnel	<String>	Read-write	Allow RTSP tunnel in HTTP. Once application/x-rtsp-tunnelled is seen in Accept or Content-Type header, NetScaler does not process Layer 7 traffic on this connection. Default value: DISABLED Possible values = ENABLED, DISABLED
reqtimeout	<Double>	Read-write	Time, in seconds, within which the HTTP request must complete. If the request does not complete within this time, the specified request timeout action is executed. Zero disables the timeout. Default value: 0 Minimum value = 0 Maximum value = 86400
adpttimeout	<String>	Read-write	Adapts the configured request timeout based on flow conditions. The timeout is increased or decreased internally and applied on the flow. Default value: DISABLED Possible values = ENABLED, DISABLED
reqtimeoutaction	<String>	Read-write	Action to take when the HTTP request does not complete within the specified request timeout duration. You can configure the following actions: * RESET - Send RST (reset) to client when timeout occurs. * DROP - Drop silently when timeout occurs. * Custom responder action - Name of the responder action to trigger when timeout occurs, used to send custom message.
dropextradata	<String>	Read-write	Drop any extra data when server sends more data than the specified content-length. Default value: DISABLED Possible values = ENABLED, DISABLED
weblog	<String>	Read-write	Enable or disable web logging. Default value: ENABLED Possible values = ENABLED, DISABLED
clientiphdrexpr	<String>	Read-write	Name of the header that contains the real client IP address.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
maxreq	<Double>	Read-write	Maximum number of requests allowed on a single connection. Zero implies no limit on the number of requests. Default value: 0 Minimum value = 0 Maximum value = 65534
persistentetag	<String>	Read-write	Generate the persistent NetScaler specific ETag for the HTTP response with ETag header. Default value: DISABLED Possible values = ENABLED, DISABLED
spdy	<String>	Read-write	Enable SPDYv2 or SPDYv3 or both over SSL vserver. SSL will advertise SPDY support either during NPN Handshake or when client will advertises SPDY support during ALPN handshake. Both SPDY versions are enabled when this parameter is set to ENABLED. Default value: DISABLED Possible values = DISABLED, ENABLED, V2, V3
http2	<String>	Read-write	Choose whether to enable support for HTTP/2 (draft-14). Default value: DISABLED Possible values = ENABLED, DISABLED
reusepooltimeout	<Double>	Read-write	Idle timeout (in seconds) for server connections in re-use pool. Connections in the re-use pool are flushed, if they remain idle for the configured timeout. Default value: 0 Minimum value = 0 Maximum value = 31536000
maxheaderlen	<Double>	Read-write	Number of bytes to be queued to look for complete header before returning error. If complete header is not obtained after queuing these many bytes, request will be marked as invalid and no L7 processing will be done for that TCP connection. Default value: 24820 Minimum value = 2048 Maximum value = 61440
minreusepool	<Double>	Read-write	Minimum limit on the number of connections, from the NetScaler to a particular server that are kept in the reuse pool. This setting is helpful for optimal memory utilization and for reducing the idle connections to the server just after the peak time. Zero implies no limit on reuse pool size. Default value: 0 Minimum value = 0 Maximum value = 360000
http2maxheaderlistsize	<Double>	Read-write	Maximum size of header list that the NetScaler is prepared to accept, in bytes. NOTE: The actual plain text header size that the NetScaler accepts is limited by maxHeaderLen. Please change this parameter as well when modifying http2MaxHeaderListSize. Default value: 24576 Minimum value = 8192 Maximum value = 65535
			Maximum size of the frame payload that the NetScaler is willing to receive,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
http2maxframesize	<Double>	Read-write	Default value: 16384 Minimum value = 16384 Maximum value = 16777215
http2maxconcurrentstreams	<Double>	Read-write	Maximum number of concurrent streams that is allowed per connection. Default value: 100 Minimum value = 0 Maximum value = 1000
http2initialwindowsize	<Double>	Read-write	Initial window size for stream level flow control, in bytes. Default value: 65535 Minimum value = 8192 Maximum value = 20971520
http2headertablesize	<Double>	Read-write	Maximum size of the header compression table used to decode header blocks, in bytes. Default value: 4096 Minimum value = 0 Maximum value = 16384
refcnt	<Double>	Read-only	Number of entities using this profile.
builtin	<String[]>	Read-only	Flag to determine if http profile is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nshttpprofile":{
    "name":<String_value>,
    "dropinvalreqs":<String_value>,
    "markhttp09inval":<String_value>,
    "markconnreqinval":<String_value>,
    "cmponpush":<String_value>,
    "conmultiplex":<String_value>,
    "maxreusepool":<Double_value>,
    "dropextracrlf":<String_value>,
    "incomphdrdelay":<Double_value>,
    "websocket":<String_value>,
    "rtsptunnel":<String_value>,
    "reqtimeout":<Double_value>,
    "adpttimeout":<String_value>,
    "reqtimeoutaction":<String_value>,
    "dropextradata":<String_value>,
    "weblog":<String_value>,
    "clientiphdrexpr":<String_value>,
    "maxreq":<Double_value>,
    "persistentetag":<String_value>,
    "spdy":<String_value>,
    "http2":<String_value>,
    "reusepooltimeout":<Double_value>,
    "maxheaderlen":<Double_value>,
    "minreusepool":<Double_value>,
    "http2maxheaderlistsize":<Double_value>,
    "http2maxframesize":<Double_value>,
    "http2maxconcurrentstreams":<Double_value>,
    "http2initialwindowsize":<Double_value>,
    "http2headertablesize":<Double_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nshttpprofile":{
    "name":<String_value>,
    "dropinvalreqs":<String_value>,
    "markhttp09inval":<String_value>,
    "markconnreqinval":<String_value>,
    "cmponpush":<String_value>,
    "conmultiplex":<String_value>,
    "maxreusepool":<Double_value>,
    "dropextracrlf":<String_value>,
    "incomphdrdelay":<Double_value>,
    "websocket":<String_value>,
    "rtsptunnel":<String_value>,
    "reqtimeout":<Double_value>,
    "adpttimeout":<String_value>,
    "reqtimeoutaction":<String_value>,
    "dropextradata":<String_value>,
    "weblog":<String_value>,
    "clientiphdrexpr":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "maxreq":<Double_value>,
    "persistentetag":<String_value>,
    "spdy":<String_value>,
    "http2":<String_value>,
    "http2maxheaderlistsize":<Double_value>,
    "http2maxframesize":<Double_value>,
    "http2maxconcurrentstreams":<Double_value>,
    "http2initialwindowsize":<Double_value>,
    "http2headertablesize":<Double_value>,
    "reusepooltimeout":<Double_value>,
    "maxheaderlen":<Double_value>,
    "minreusepool":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nshttpprofile":{
    "name":<String_value>,
    "dropinvalreqs":true,
    "markhttp09inval":true,
    "markconnreqinval":true,
    "cmponpush":true,
    "conmultiplex":true,
    "maxreusepool":true,
    "dropextracrlf":true,
    "incomphdrdelay":true,
    "websocket":true,
    "dropextradata":true,
    "clientiphdrexpr":true,
    "reqtimeout":true,
    "adpttimeout":true,
    "reqtimeoutaction":true,
    "weblog":true,
    "maxreq":true,
    "persistentetag":true,
    "spdy":true,
    "http2":true,
    "http2maxheaderlistsize":true,
    "http2maxframesize":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "http2maxconcurrentstreams":true,
        "http2initialwindowsize":true,
        "http2headertablesize":true,
        "reusepooltimeout":true,
        "maxheaderlen":true,
        "rtsptunnel":true,
        "minreusepool":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nshttpprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nshttpprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nshttpprofile": [ {
    "name":<String_value>,
    "dropinvalreqs":<String_value>,
    "markhttp09inval":<String_value>,
    "markconnreqinval":<String_value>,
    "cmponpush":<String_value>,
    "conmultiplex":<String_value>,
    "maxreusepool":<Double_value>,
    "websocket":<String_value>,
    "refcnt":<Double_value>,
    "dropextracrlf":<String_value>,
    "incomphdrdelay":<Double_value>,
    "reqtimeout":<Double_value>,
    "adpttimeout":<String_value>,
    "reqtimeoutaction":<String_value>,
    "dropextradata":<String_value>,
    "weblog":<String_value>,
    "clientiphdrexpr":<String_value>,
    "maxreq":<Double_value>,
    "persistentetag":<String_value>,
    "spdy":<String_value>,
    "http2":<String_value>,
    "http2maxheaderlistsize":<Double_value>,
    "http2maxframesize":<Double_value>,
    "http2maxconcurrentstreams":<Double_value>,
    "http2initialwindowsize":<Double_value>,
    "http2headertablesize":<Double_value>,
    "reusepooltimeout":<Double_value>,
    "maxheaderlen":<Double_value>,
    "rtsptunnel":<String_value>,
    "minreusepool":<Double_value>,
    "builtin":<String[]_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile/name_value<String>

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nshttpprofile": [ {
    "name":<String_value>,
    "dropinvalreqs":<String_value>,
    "markhttp09inval":<String_value>,
    "markconnreqinval":<String_value>,
    "cmponpush":<String_value>,
    "conmultiplex":<String_value>,
    "maxreusepool":<Double_value>,
    "websocket":<String_value>,
    "refcnt":<Double_value>,
    "dropextracrlf":<String_value>,
    "incomphdrdelay":<Double_value>,
    "reqtimeout":<Double_value>,
    "adpttimeout":<String_value>,
    "reqtimeoutaction":<String_value>,
    "dropextradata":<String_value>,
    "weblog":<String_value>,
    "clientiphdrexpr":<String_value>,
    "maxreq":<Double_value>,
    "persistentetag":<String_value>,
    "spdy":<String_value>,
    "http2":<String_value>,
    "http2maxheaderlistsize":<Double_value>,
    "http2maxframesize":<Double_value>,
    "http2maxconcurrentstreams":<Double_value>,
    "http2initialwindowsize":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "http2headertablesize":<Double_value>,
    "reusepooltimeout":<Double_value>,
    "maxheaderlen":<Double_value>,
    "rtsptunnel":<String_value>,
    "minreusepool":<Double_value>,
    "builtin":<String[]_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nshttpprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nshttpprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsip6

May 20, 2016

Configuration for ip6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipv6address	<String>	Read-write	IPv6 address to create on the NetScaler appliance. Minimum length = 1
scope	<String>	Read-write	Scope of the IPv6 address to be created. Cannot be changed after the IP address is created. Default value: global Possible values = global, link-local
type	<String>	Read-write	Type of IP address to be created on the NetScaler appliance. Cannot be changed after the IP address is created. Default value: SNIP Possible values = NSIP, VIP, SNIP, GSLBsiteIP, ADNSsvcIP, RADIUSListenersvcIP, CLIP
vlan	<Double>	Read-write	The VLAN number. Default value: 0 Minimum value = 0 Maximum value = 4094
nd	<String>	Read-write	Respond to Neighbor Discovery (ND) requests for this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
icmp	<String>	Read-write	Respond to ICMP requests for this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
vserver	<String>	Read-write	Enable or disable the state of all the virtual servers associated with this VIP6 address. Default value: ENABLED Possible values = ENABLED, DISABLED
telnet	<String>	Read-write	Allow Telnet access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
ftp	<String>	Read-write	Allow File Transfer Protocol (FTP) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: ENABLED Possible values = ENABLED, SECUREONLY, DISABLED
ssh	<String>	Read-write	Allow secure Shell (SSH) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
snmp	<String>	Read-write	Allow Simple Network Management Protocol (SNMP) access to this IP address. Default value: ENABLED Possible values = ENABLED, DISABLED
mgmtaccess	<String>	Read-write	Allow access to management applications on this IP address. Default value: DISABLED Possible values = ENABLED, DISABLED
restrictaccess	<String>	Read-write	Block access to nonmanagement applications on this IP address. This option is applicable forMIP6s, SNIP6s, and NSIP6s, and is disabled by default. Nonmanagement applications can run on the underlying NetScaler Free BSD operating system. Default value: DISABLED Possible values = ENABLED, DISABLED
dynamicrouting	<String>	Read-write	Allow dynamic routing on this IP address. Specific to Subnet IPv6 (SNIP6) address. Default value: DISABLED Possible values = ENABLED, DISABLED
hostroute	<String>	Read-write	Advertise a route for the VIP6 address by using the dynamic routing protocols running on the NetScaler appliance. Possible values = ENABLED, DISABLED
ip6hostrtgw	<String>	Read-write	IPv6 address of the gateway for the route. If Gateway is not set, VIP uses :: as the gateway. Default value: 0
metric	<Integer>	Read-write	Integer value to add to or subtract from the cost of the route advertised for the VIP6 address. Minimum value = -16777215
vserverrhilevel	<String>	Read-write	Advertise or do not advertise the route for the Virtual IP (VIP6) address on the basis of the state of the virtual servers associated with that VIP6. * NONE - Advertise the route for the VIP6 address, irrespective of the state of the virtual servers associated with the address. * ONE VSERVER - Advertise the route for the VIP6 address if at least one of the associated virtual servers is in UP state. * ALL VSERVER - Advertise the route for the VIP6 address if all of the associated virtual servers are in UP state. * VSVR_CNTRLD. Advertise the route for the VIP address according to the RHIstate (RHI STATE) parameter setting on all the associated virtual servers of the VIP address along with their states. When Vserver RHI Level (RHI) parameter is set to VSVR_CNTRLD, the following are different RHI behaviors for the VIP address on the basis of RHIstate (RHI STATE) settings on the virtual servers associated with the VIP address: * If you set RHI STATE to PASSIVE on all virtual servers, the NetScaler ADC always

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			*If you set RHI STATE to ACTIVE on all virtual servers, the NetScaler ADC advertises the route for the VIP address if at least one of the associated virtual servers is in UP state. *If you set RHI STATE to ACTIVE on some and PASSIVE on others, the NetScaler ADC advertises the route for the VIP address if at least one of the associated virtual servers, whose RHI STATE set to ACTIVE, is in UP state. Default value: ONE_VSERVER Possible values = ONE_VSERVER, ALL_VSERVERS, NONE, VSVR_CNTRLD
ospf6lsatype	<String>	Read-write	Type of LSAs to be used by the IPv6 OSPF protocol, running on the NetScaler appliance, for advertising the route for the VIP6 address. Default value: EXTERNAL Possible values = INTRA_AREA, EXTERNAL
ospfarea	<Double>	Read-write	ID of the area in which the Intra-Area-Prefix LSAs are to be advertised for the VIP6 address by the IPv6 OSPF protocol running on the NetScaler appliance. When ospfArea is not set, VIP6 is advertised on all areas. Default value: -1 Minimum value = 0 Maximum value = 4294967294LU
state	<String>	Read-write	Enable or disable the IP address. Default value: ENABLED Possible values = DISABLED, ENABLED
map	<String>	Read-write	Mapped IPV4 address for the IPV6 address.
ownernode	<Double>	Read-write	ID of the cluster node for which you are adding the IP address. Must be used if you want the IP address to be active only on the specific node. Can be configured only through the cluster IP address. Cannot be changed after the IP address is created. Default value: 255
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
iptype	<String[]>	Read-only	The type of the IPv6 address. Possible values = NSIP, VIP, SNIP, GSLBsiteIP, ADNSsvcIP, RADIUSListenersvcIP, CLIP
curstate	<String>	Read-only	Current state of this IP. Default value: ENABLED Possible values = DISABLED, ENABLED
viprtadv2bsd	<Boolean>	Read-only	Whether this route is advertised to FreeBSD.
vipvsercount	<Double>	Read-only	Number of vservers bound to this VIP.
vipvserdowncount	<Double>	Read-only	Number of vservers bound to this VIP, which are down.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
systemtype	<String>	Read-only	The type of the system. Possible values: Standalone, HA, Cluster. Used for display purpose. Possible values = Stand-alone, HA, Cluster
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip6

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"nsip6":{
    "ipv6address":<String_value>,
    "scope":<String_value>,
    "type":<String_value>,
    "vlan":<Double_value>,
    "nd":<String_value>,
    "icmp":<String_value>,
    "vserver":<String_value>,
    "telnet":<String_value>,
    "ftp":<String_value>,
    "gui":<String_value>,
    "ssh":<String_value>,
    "snmp":<String_value>,
    "mgmtaccess":<String_value>,
    "restrictaccess":<String_value>,
    "dynamicrouting":<String_value>,
    "hostroute":<String_value>,
    "ip6hostrtgw":<String_value>,
    "metric":<Integer_value>,
    "vserverrhilevel":<String_value>,
    "ospf6lsatype":<String_value>,
    "ospfarea":<Double_value>,
    "state":<String_value>,
    "map":<String_value>,
    "ownernode":<Double_value>,
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip6/ipv6address_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip6

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsip6":{
    "ipv6address":<String_value>,
    "td":<Double_value>,
    "nd":<String_value>,
    "icmp":<String_value>,
    "vserver":<String_value>,
    "telnet":<String_value>,
    "ftp":<String_value>,
    "gui":<String_value>,
    "ssh":<String_value>,
    "snmp":<String_value>,
    "mgmtaccess":<String_value>,
    "restrictaccess":<String_value>,
    "state":<String_value>,
    "map":<String_value>,
    "dynamicrouting":<String_value>,
    "hostroute":<String_value>,
    "ip6hostrtgw":<String_value>,
    "metric":<Integer_value>,
    "vserverrhilevel":<String_value>,
    "ospf6lsatype":<String_value>,
    "ospfarea":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip6?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"nsip6":{
    "ipv6address":<String_value>,
    "td":<Double_value>,
    "ospfarea":true,
    "nd":true,
    "icmp":true,
    "vserver":true,
    "telnet":true,
    "ftp":true,
    "gui":true,
    "ssh":true,
    "snmp":true,
    "mgmtaccess":true,
    "restrictaccess":true,
    "state":true,
    "map":true,
    "dynamicrouting":true,
    "hostroute":true,
    "ip6hostrtgw":true,
    "metric":true,
    "vserverrhilevel":true,
    "ospf6lsatype":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip6

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsip6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsip6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsip6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsip6?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsip6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsip6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsip6": [ {
    "ipv6address":<String_value>,
    "td":<Double_value>,
    "scope":<String_value>,
    "type":<String_value>,
    "iptype":<String[]_value>,
    "vlan":<Double_value>,
    "nd":<String_value>,
    "icmp":<String_value>,
    "vserver":<String_value>,
    "telnet":<String_value>,
    "ssh":<String_value>,
    "gui":<String_value>,
    "snmp":<String_value>,
    "ftp":<String_value>,
    "mgmtaccess":<String_value>,
    "restrictaccess":<String_value>,
    "state":<String_value>,
    "curstate":<String_value>,
    "map":<String_value>,
    "dynamicrouting":<String_value>,
    "hostroute":<String_value>,
    "ip6hostrtgw":<String_value>,
    "metric":<Integer_value>,
    "vserverrhilevel":<String_value>,
    "viprtadv2bsd":<Boolean_value>,
    "vipvsercount":<Double_value>,
    "vipvserdowncount":<Double_value>,
    "ospf6lsatype":<String_value>,
    "ospfarea":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ownernode":<Double_value>,
    "systemtype":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsip6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsip6": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nslicense

Jul 07, 2017

Configuration for license resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
wl	<Boolean>	Read-only	Web Logging.
sp	<Boolean>	Read-only	Surge Protection.
lb	<Boolean>	Read-only	Load Balancing.
cs	<Boolean>	Read-only	Content Switching.
cr	<Boolean>	Read-only	Cache Redirect.
sc	<Boolean>	Read-only	Sure Connect.
cmp	<Boolean>	Read-only	Compression.
delta	<Boolean>	Read-only	Delta Compression.
pq	<Boolean>	Read-only	Priority Queuing.
ssl	<Boolean>	Read-only	Secure Sockets Layer.
gslb	<Boolean>	Read-only	Global Server Load Balancing.
gslbp	<Boolean>	Read-only	GSLB Proximity.
hdosp	<Boolean>	Read-only	DOS Protection.
routing	<Boolean>	Read-only	Routing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cf	<Boolean>	Read-only	Content Filter.
contentaccelerator	<Boolean>	Read-only	transparent Integrated Caching.
ic	<Boolean>	Read-only	Integrated Caching.
sslvpn	<Boolean>	Read-only	SSL VPN.
f_sslvpn_users	<Double>	Read-only	Number of licensed users allowed by this license.
f_ica_users	<Double>	Read-only	Number of licensed users allowed by ICAONLY license. As long as the AG Feature is licensed, unlimited number of ICA connections are accepted. (In API, 0 value for this parameter means unlimited when AG license in ON).
aaa	<Boolean>	Read-only	AAA.
ospf	<Boolean>	Read-only	OSPF Routing.
rip	<Boolean>	Read-only	RIP Routing.
bgp	<Boolean>	Read-only	BGP Routing.
rewrite	<Boolean>	Read-only	Rewrite.
ipv6pt	<Boolean>	Read-only	IPv6 protocol translation.
appfw	<Boolean>	Read-only	Application Firewall.
responder	<Boolean>	Read-only	Responder.
agee	<Boolean>	Read-only	.
nsxn	<Boolean>	Read-only	.
htmlinjection	<Boolean>	Read-only	HTML Injection.
modelid	<Double>	Read-only	Model Number ID.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
push	<Boolean>	Read-only	NetScaler Push.
wionns	<Boolean>	Read-only	WI on NS.
appflow	<Boolean>	Read-only	AppFlow.
cloudbridge	<Boolean>	Read-only	CloudBridge.
cloudbridgeappliance	<Boolean>	Read-only	.
cloudextenderappliance	<Boolean>	Read-only	.
isis	<Boolean>	Read-only	ISIS Routing.
cluster	<Boolean>	Read-only	Clustering.
ch	<Boolean>	Read-only	Call Home.
appqoe	<Boolean>	Read-only	AppQoS.
appflowica	<Boolean>	Read-only	Appflow for ICA.
isstandardlic	<Boolean>	Read-only	Standard License.
isenterpriselic	<Boolean>	Read-only	Enterprise License.
isplatinumlic	<Boolean>	Read-only	Platinum License.
rise	<Boolean>	Read-only	RISE.
vpath	<Boolean>	Read-only	Vpath.
feo	<Boolean>	Read-only	Front End Optimization.
lsn	<Boolean>	Read-only	Large Scale NAT.
rdpproxy	<Boolean>	Read-only	RDPPROXY.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rep	<Boolean>	Read-only	IP Reputation.
Name	**Data Type**	**Permissions**	**Description**

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nslicense

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslicense": [ {
    "wl":<Boolean_value>,
    "sp":<Boolean_value>,
    "lb":<Boolean_value>,
    "cs":<Boolean_value>,
    "cr":<Boolean_value>,
    "sc":<Boolean_value>,
    "cmp":<Boolean_value>,
    "delta":<Boolean_value>,
    "pq":<Boolean_value>,
    "ssl":<Boolean_value>,
    "gslb":<Boolean_value>,
    "gslbp":<Boolean_value>,
    "hdosp":<Boolean_value>,
    "routing":<Boolean_value>,
    "cf":<Boolean_value>,
    "contentaccelerator":<Boolean_value>,
    "ic":<Boolean_value>,
    "sslvpn":<Boolean_value>,
    "f_sslvpn_users":<Double_value>,
    "f_ica_users":<Double_value>,
    "aaa":<Boolean_value>,
    "ospf":<Boolean_value>,
    "rip":<Boolean_value>,
    "bgp":<Boolean_value>,
    "rewrite":<Boolean_value>,
    "ipv6pt":<Boolean_value>,
    "appfw":<Boolean_value>,
    "responder":<Boolean_value>,
    "agee":<Boolean_value>,
    "nsxn":<Boolean_value>,
    "htmlinjection":<Boolean_value>,
    "modelid":<Double_value>,
    "push":<Boolean_value>,
    "wionns":<Boolean_value>,
    "appflow":<Boolean_value>,
    "cloudbridge":<Boolean_value>,
    "cloudbridgeappliance":<Boolean_value>,
    "cloudextenderappliance":<Boolean_value>,
    "isis":<Boolean_value>,
    "cluster":<Boolean_value>,
    "ch":<Boolean_value>,
    "appqoe":<Boolean_value>,
    "appflowica":<Boolean_value>,
    "isstandardlic":<Boolean_value>,
    "isenterpriselic":<Boolean_value>,
    "isplatinumlic":<Boolean_value>,
    "rise":<Boolean_value>,
    "vpath":<Boolean_value>,
    "feo":<Boolean_value>,
    "lsn":<Boolean_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "rdpproxy":<Boolean_value>,
    "rep":<Boolean_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nslicenseproxyserver

May 20, 2016

Configuration for licenseproxyserver resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
serverip	<String>	Read-write	IP address of the License proxy server. Minimum length = 1
servername	<String>	Read-write	Fully qualified domain name of the License proxy server.
port	<Double>	Read-write	License proxy server port.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nslicenseproxyserver

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nslicenseproxyserver":{
    "serverip":<String_value>,
    "servername":<String_value>,
    "port":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nslicenseproxyserver/serverip_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nslicenseproxyserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nslicenseproxyserver":{
    "serverip":<String_value>,
    "servername":<String_value>,
    "port":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nslicenseproxyserver

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nslicenseproxyserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nslicenseproxyserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nslicenseproxyserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nslicenseproxyserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nslicenseproxyserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nslicenseproxyserver resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslicenseproxyserver": [ {
    "serverip":<String_value>,
    "servername":<String_value>,
    "port":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nslicenseproxyserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslicenseproxyserver": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nslimitidentifier

May 20, 2016

Configuration for limit Indetifier resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
limitidentifier	<String>	Read-write	Name for a rate limit identifier. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Reserved words must not be used.
threshold	<Double>	Read-write	Maximum number of requests that are allowed in the given timeslice when requests (mode is set as REQUEST_RATE) are tracked per timeslice. When connections (mode is set as CONNECTION) are tracked, it is the total number of connections that would be let through. Default value: 1 Minimum value = 1
timeslice	<Double>	Read-write	Time interval, in milliseconds, specified in multiples of 10, during which requests are tracked to check if they cross the threshold. This argument is needed only when the mode is set to REQUEST_RATE. Default value: 1000 Minimum value = 10
mode	<String>	Read-write	Defines the type of traffic to be tracked. * REQUEST_RATE - Tracks requests/timeslice. * CONNECTION - Tracks active transactions. Examples 1. To permit 20 requests in 10 ms and 2 traps in 10 ms: add limitidentifier limit_req -mode request_rate -limitType smooth -timeslice 1000 -Threshold 2000 -trapsInTimeSlice 200 2. To permit 50 requests in 10 ms: set limitidentifier limit_req -mode request_rate -timeslice 1000 -Threshold 5000 -limitType smooth 3. To permit 1 request in 40 ms: set limitidentifier limit_req -mode request_rate -timeslice 2000 -Threshold 50 -limitType smooth 4. To permit 1 request in 200 ms and 1 trap in 130 ms: set limitidentifier limit_req -mode request_rate -timeslice 1000 -Threshold 5 -limitType smooth -trapsInTimeSlice 8

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			set limitidentifier limit_req -mode request_rate -timeslice 1000 -threshold 5000 -limitType BURSTY. Default value: REQUEST_RATE Possible values = CONNECTION, REQUEST_RATE, NONE
limittype	<String>	Read-write	Smooth or bursty request type. * SMOOTH - When you want the permitted number of requests in a given interval of time to be spread evenly across the timeslice * BURSTY - When you want the permitted number of requests to exhaust the quota anytime within the timeslice. This argument is needed only when the mode is set to REQUEST_RATE. Default value: BURSTY Possible values = BURSTY, SMOOTH
selectorname	<String>	Read-write	Name of the rate limit selector. If this argument is NULL, rate limiting will be applied on all traffic received by the virtual server or the NetScaler (depending on whether the limit identifier is bound to a virtual server or globally) without any filtering. Minimum length = 1
maxbandwidth	<Double>	Read-write	Maximum bandwidth permitted, in kbps. Minimum value = 0 Maximum value = 4294967287
trapsintimeslice	<Double>	Read-write	Number of traps to be sent in the timeslice configured. A value of 0 indicates that traps are disabled. Minimum value = 0 Maximum value = 65535
ngname	<String>	Read-only	Nodegroup name to which this identifier belongs to.
hits	<Double>	Read-only	The number of times this identifier was evaluated.
drop	<Double>	Read-only	The number of times action was taken.
rule	<String[]>	Read-only	Rule.
time	<Double>	Read-only	Time interval considered for rate limiting.
total	<Double>	Read-only	Maximum number of requests permitted in the computed timeslice.
trapscomputedintimeslice	<Double>	Read-only	The number of traps that would be sent in the timeslice configured. .
computedtraptimeslice	<Double>	Read-only	The time interval computed for sending traps.
referencecount	<Double>	Read-only	Total number of transactions pointing to this entry.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nslimitidentifier":{
    "limitidentifier":<String_value>,
    "threshold":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "timeslice":<Double_value>,
    "mode":<String_value>,
    "limittype":<String_value>,
    "selectorname":<String_value>,
    "maxbandwidth":<Double_value>,
    "trapsintimeslice":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier/limitidentifier_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nslimitidentifier":{
    "limitidentifier":<String_value>,
    "threshold":<Double_value>,
    "timeslice":<Double_value>,
    "mode":<String_value>,
    "limittype":<String_value>,
    "selectorname":<String_value>,
    "maxbandwidth":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "trapsintimeslice":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nslimitidentifier":{
    "limitidentifier":<String_value>,
    "selectorname":true,
    "threshold":true,
    "timeslice":true,
    "mode":true,
    "limittype":true,
    "maxbandwidth":true,
    "trapsintimeslice":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nslimitidentifier resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nslimitidentifier resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslimitidentifier": [ {
    "limitidentifier":<String_value>,
    "ngname":<String_value>,
    "threshold":<Double_value>,
    "timeslice":<Double_value>,
    "mode":<String_value>,
    "limittype":<String_value>,
    "selectorname":<String_value>,
    "hits":<Double_value>,
    "drop":<Double_value>,
    "rule":<String[]_value>,
    "time":<Double_value>,
    "total":<Double_value>,
    "maxbandwidth":<Double_value>,
    "trapsintimeslice":<Double_value>,
    "trapscomputedintimeslice":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "computedtraptimeslice":<Double_value>,
    "referencecount":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier/limitidentifier_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier/limitidentifier_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier/limitidentifier_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslimitidentifier": [ {
    "limitidentifier":<String_value>,
    "ngname":<String_value>,
    "threshold":<Double_value>,
    "timeslice":<Double_value>,
    "mode":<String_value>,
    "limittype":<String_value>,
    "selectorname":<String_value>,
    "hits":<Double_value>,
    "drop":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "rule":<String[]_value>,
    "time":<Double_value>,
    "total":<Double_value>,
    "maxbandwidth":<Double_value>,
    "trapsintimeslice":<Double_value>,
    "trapscomputedintimeslice":<Double_value>,
    "computedtraptimeslice":<Double_value>,
    "referencecount":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslimitidentifier": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nslimitidentifier_binding

Jul 07, 2017

Binding object which returns the resources bound to nslimitidentifier.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
limitidentifier	<String>	Read-write	Name of the rate limit identifier about which to display information. If a name is not provided, information about all rate limit identifiers is shown.
nslimitidentifier_nslimitsessions_binding	<nslimitidentifier_nslimitsessions_binding[]>	Read-only	nslimitsessions that can be bound to nslimitidentifier.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier_binding/limitidentifier_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "nslimitidentifier_binding": [ {
      "limitidentifier":<String_value>,
      "nslimitidentifier_nslimitsessions_binding":<nslimitidentifier_nslimitsessions_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nslimitidentifier_nslimitsessions_binding

May 20, 2016

Binding object showing the nslimitsessions that can be bound to nslimitidentifier.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
limitidentifier	<String>	Read-write	Name of the rate limit identifier about which to display information. If a name is not provided, information about all rate limit identifiers is shown.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier_nslimitsessions_binding/limitidentifier_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier_nslimitsessions_binding/limitidentifier_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nslimitidentifier_nslimitsessions_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier_nslimitsessions_binding/limitidentifier_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nslimitidentifier_nslimitsessions_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslimitidentifier_nslimitsessions_binding": [ {
    "limitidentifier":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitidentifier_nslimitsessions_binding/limitidentifier_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"nslimitidentifier_nslimitsessions_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nslimitselector

May 20, 2016

Configuration for limit selector resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
selectorname	<String>	Read-write	.
rule	<String[]>	Read-write	. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitselector

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nslimitselector":{
    "selectorname":<String_value>,
    "rule":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitselector/selectorname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitselector

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nslimitselector":{
    "selectorname":<String_value>,
    "rule":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitselector?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nslimitselector":{
    "selectorname":<String_value>,
    "rule":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitselector

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nslimitselector?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nslimitselector?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nslimitselector resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nslimitselector?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nslimitselector?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nslimitselector resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslimitselector": [ {
      "selectorname":<String_value>,
      "rule":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitselector/selectorname_value<String>

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nslimitselector/selectorname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nslimitselector/selectorname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslimitselector": [ {
      "selectorname":<String_value>,
      "rule":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitselector?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "nslimitselector": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nslimitsessions

May 20, 2016

Configuration for limit sessions resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
limitidentifier	<String>	Read-write	Name of the rate limit identifier for which to display the sessions. Minimum length = 1
detail	<Boolean>	Read-write	Show the individual hash values.
timeout	<Double>	Read-only	The time remaining on the session before a flush can be attempted. If active transactions are present the session will not be flushed.
hits	<Double>	Read-only	The number of times this entry was hit.
drop	<Double>	Read-only	The number of times action was taken.
number	<Double[]>	Read-only	The hash of the matched selectlets.
name	<String>	Read-only	The string formed by gathering selectlet values.
unit	<Double>	Read-only	Total computed hash of the matched selectlets.
flags	<Double>	Read-only	Used internally to identify ip addresses.
referencecount	<Double>	Read-only	Total number of transactions pointing to this entry. Its the sum total of the connection and bandwidth references.
maxbandwidth	<Double>	Read-only	The current bandwidth.
selectoripv61	<String>	Read-only	First IPV6 address gathered.
selectoripv62	<String>	Read-only	Second IPV6 address gathered.
flag	<Double>	Read-only	Used internally to identify ipv6 addresses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
count	<Double>	Read-only	count parameter

Operations (click to see Properties)

CLEAR | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitsessions?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nslimitsessions":{
    "limitidentifier":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitsessions

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/nslimitsessions?**args=limitidentifier:<String_value>,detail:<Boolean_value>**

Use this query-parameter to get nslimitsessions resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/nslimitsessions?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nslimitsessions?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nslimitsessions resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nslimitsessions?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nslimitsessions?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nslimitsessions resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslimitsessions": [ {
limitidentifier:<String_value>,detail:<Boolean_value>      "timeout":<Double_value>,
      "hits":<Double_value>,
      "drop":<Double_value>,
      "number":<Double[]_value>,
      "name":<String_value>,
      "unit":<Double_value>,
      "flags":<Double_value>,
      "referencecount":<Double_value>,
      "maxbandwidth":<Double_value>,
      "selectoripv61":<String_value>,
      "selectoripv62":<String_value>,
      "flag":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nslimitsessions?**args=limitidentifier:<String_value>,detail:<Boolean_value>&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "nslimitsessions": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsmode

May 20, 2016

Configuration for ns mode resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
mode	<String[]>	Read-write	Mode to be enabled. Multiple modes can be specified by providing a blank space between each mode. Possible values = FR, FastRamp, L2, L2mode, L3, L3mode, USIP, UseSourceIP, CKA, ClientKeepAlive, TCPB, TCPBuffering, MBF, MACbasedforwarding, Edge, USNIP, SRADV, DRADV, IRADV, SRADV6, DRADV6, PMTUD, RISE_APBR, RISE_RHI, BridgeBPDUs, Mediaclassification
fr	<Boolean>	Read-only	Fast Ramp.
l2	<Boolean>	Read-only	Layer 2 mode.
usip	<Boolean>	Read-only	Use Source IP.
cka	<Boolean>	Read-only	Client Keep-alive.
tcpb	<Boolean>	Read-only	TCP Buffering.
mbf	<Boolean>	Read-only	MAC-based forwarding.
edge	<Boolean>	Read-only	Edge configuration.
usnip	<Boolean>	Read-only	Use Subnet IP.
l3	<Boolean>	Read-only	Layer 3 mode (ip forwarding).
pmtud	<Boolean>	Read-only	Path MTU Discovery.
mediaclassification	<Boolean>	Read-only	mediaclassification.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
dradv	<Boolean>	Read-only	Direct Route Advertisement.
iradv	<Boolean>	Read-only	Intranet Route Advertisement.
sradv6	<Boolean>	Read-only	Ipv6 Static Route Advertisement.
dradv6	<Boolean>	Read-only	Ipv6 Direct Route Advertisement.
bridgebpdus	<Boolean>	Read-only	BPDU Bridging Mode.
rise_apbr	<Boolean>	Read-only	APBR Publishing Mode.
rise_rhi	<Boolean>	Read-only	RHI Publishing Mode.

Operations (click to see Properties)

ENABLE | DISABLE | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/nsmode?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsmode":{
    "mode":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/nsmode?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsmode":{
    "mode":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsmode

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsmode": [ {
      "mode":<String[]_value>,
      "fr":<Boolean_value>,
      "l2":<Boolean_value>,
      "usip":<Boolean_value>,
      "cka":<Boolean_value>,
      "tcpb":<Boolean_value>,
      "mbf":<Boolean_value>,
      "edge":<Boolean_value>,
      "usnip":<Boolean_value>,
      "l3":<Boolean_value>,
      "pmtud":<Boolean_value>,
      "mediaclassification":<Boolean_value>,
      "sradv":<Boolean_value>,
      "dradv":<Boolean_value>,
      "iradv":<Boolean_value>,
      "sradv6":<Boolean_value>,
      "dradv6":<Boolean_value>,
      "bridgebpdus":<Boolean_value>,
      "rise_apbr":<Boolean_value>,
      "rise_rhi":<Boolean_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsparam

May 20, 2016

Configuration for netscaler parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
httpport	<Integer[]>	Read-write	HTTP ports on the web server. This allows the system to perform connection off-load for any client request that has a destination port matching one of these configured ports. Minimum value = 1 Maximum value = 65535
maxconn	<Double>	Read-write	Maximum number of connections that will be made from the appliance to the web server(s) attached to it. The value entered here is applied globally to all attached servers. Default value: 0 Minimum value = 0 Maximum value = 4294967294
maxreq	<Double>	Read-write	Maximum number of requests that the system can pass on a particular connection between the appliance and a server attached to it. Setting this value to 0 allows an unlimited number of requests to be passed. This value is overridden by the maximum number of requests configured on the individual service. Minimum value = 0 Maximum value = 65535
cip	<String>	Read-write	Enable or disable the insertion of the actual client IP address into the HTTP header request passed from the client to one, some, or all servers attached to the system. The passed address can then be accessed through a minor modification to the server. * If the CIP header is specified, it will be used as the client IP header. * If the CIP header is not specified, the value that has been set will be used as the client IP header. Possible values = ENABLED, DISABLED
cipheader	<String>	Read-write	Text that will be used as the client IP address header. Minimum length = 1
cookieversion	<String>	Read-write	Version of the cookie inserted by the system. Possible values = 0, 1
securecookie	<String>	Read-write	Enable or disable secure flag for persistence cookie. Default value: ENABLED Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
pmtumin	<Double>	Read-write	fragmentation needed message. If the ICMP message contains a value less than this value, then this value is used instead. Default value: 576 Minimum value = 168 Maximum value = 1500
pmtutimeout	<Double>	Read-write	Interval, in minutes, for flushing the PMTU entries. Default value: 10 Minimum value = 1 Maximum value = 1440
ftpportrange	<String>	Read-write	Minimum and maximum port (port range) that FTP services are allowed to use. Minimum length = 1024 Maximum length = 64000
crportrange	<String>	Read-write	Port range for cache redirection services. Minimum length = 1 Maximum length = 65535
			Time zone for the NetScaler appliance. Name of the time zone should be specified as argument. Possible values = CoordinatedUniversalTime, GMT+01:00-CET-Europe/Andorra, GMT+04:00-GST-Asia/Dubai, GMT+04:30-AFT-Asia/Kabul, GMT-04:00-AST-America/Antigua, GMT-04:00-AST-America/Anguilla, GMT+01:00-CET-Europe/Tirane, GMT+04:00-AMT-Asia/Yerevan, GMT+01:00-WAT-Africa/Luanda, GMT+13:00-NZDT-Antarctica/McMurdo, GMT+13:00-NZDT-Antarctica/South_Pole, GMT-03:00-ROTT-Antarctica/Rothera, GMT-04:00-CLT-Antarctica/Palmer, GMT+05:00-MAWT-Antarctica/Mawson, GMT+07:00-DAVT-Antarctica/Davis, GMT+08:00-WST-Antarctica/Casey, GMT+06:00-VOST-Antarctica/Vostok, GMT+10:00-DDUT-Antarctica/DumontDUrville, GMT+03:00-SYOT-Antarctica/Syowa, GMT+11:00-MIST-Antarctica/Macquarie, GMT-03:00-ART-America/Argentina/Buenos_Aires, GMT-03:00-ART-America/Argentina/Cordoba, GMT-03:00-ART-America/Argentina/Salta, GMT-03:00-ART-America/Argentina/Jujuy, GMT-03:00-ART-America/Argentina/Tucuman, GMT-03:00-ART-America/Argentina/Catamarca, GMT-03:00-ART-America/Argentina/La_Rioja, GMT-03:00-ART-America/Argentina/San_Juan, GMT-03:00-ART-America/Argentina/Mendoza, GMT-03:00-WARST-America/Argentina/San_Luis, GMT-03:00-ART-America/Argentina/Rio_Gallegos, GMT-03:00-ART-America/Argentina/Ushuaia, GMT-11:00-SST-Pacific/Pago_Pago, GMT+01:00-CET-Europe/Vienna, GMT+11:00-LHST-Australia/Lord_Howe, GMT+11:00-EST-Australia/Hobart, GMT+11:00-EST-Australia/Currie, GMT+11:00-EST-Australia/Melbourne, GMT+11:00-EST-Australia/Sydney, GMT+10:30-CST-Australia/Broken_Hill, GMT+10:00-EST-Australia/Brisbane, GMT+10:00-EST-Australia/Lindeman, GMT+10:30-CST-Australia/Adelaide, GMT+09:30-CST-Australia/Darwin, GMT+08:00-WST-Australia/Perth, GMT+08:45-CWST-Australia/Eucla, GMT-04:00-AST-America/Aruba, GMT+02:00-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Europe/Sarajevo, GMT-04:00-AST-America/Barbados, GMT+06:00-BDT-Asia/Dhaka, GMT+01:00-CET-Europe/Brussels, GMT+00:00-GMT-Africa/Ouagadougou, GMT+02:00-EET-Europe/Sofia, GMT+03:00-AST-Asia/Bahrain, GMT+02:00-CAT-Africa/Bujumbura, GMT+01:00-WAT-Africa/Porto-Novo, GMT-04:00-AST-America/St_Barthelemy, GMT-03:00-ADT-Atlantic/Bermuda, GMT+08:00-BNT-Asia/Brunei, GMT-04:00-BOT-America/La_Paz, GMT-02:00-FNT-America/Noronha, GMT-03:00-BRT-America/Belem, GMT-03:00-BRT-America/Fortaleza, GMT-03:00-BRT-America/Recife, GMT-03:00-BRT-America/Araguaina, GMT-03:00-BRT-America/Maceio, GMT-03:00-BRT-America/Bahia, GMT-03:00-BRT-America/Sao_Paulo, GMT-04:00-AMT-America/Campo_Grande, GMT-04:00-AMT-America/Cuiaba, GMT-03:00-BRT-America/Santarem, GMT-04:00-AMT-America/Porto_Velho, GMT-04:00-AMT-America/Boa_Vista, GMT-04:00-AMT-America/Manaus, GMT-04:00-AMT-America/Eirunepe, GMT-04:00-AMT-America/Rio_Branco, GMT-04:00-EDT-America/Nassau, GMT+06:00-BTT-Asia/Thimphu, GMT+02:00-CAT-Africa/Gaborone, GMT+03:00-FET-Europe/Minsk, GMT-06:00-CST-America/Belize, GMT-02:30-NDT-America/St_Johns, GMT-03:00-ADT-America/Halifax, GMT-03:00-ADT-America/Glace_Bay, GMT-03:00-ADT-America/Moncton, GMT-03:00-ADT-America/Goose_Bay, GMT-04:00-AST-America/Blanc-Sablon, GMT-04:00-EDT-America/Montreal, GMT-04:00-EDT-America/Toronto, GMT-04:00-EDT-America/Nipigon, GMT-04:00-EDT-America/Thunder_Bay, GMT-04:00-EDT-America/Iqaluit, GMT-04:00-EDT-America/Pangnirtung, GMT-05:00-CDT-America/Resolute, GMT-05:00-EST-America/Atikokan, GMT-05:00-CDT-America/Rankin_Inlet, GMT-05:00-CDT-America/Winnipeg, GMT-05:00-CDT-America/Rainy_River, GMT-06:00-CST-America/Regina, GMT-06:00-CST-America/Swift_Current, GMT-06:00-MDT-America/Edmonton, GMT-06:00-MDT-America/Cambridge_Bay, GMT-06:00-MDT-America/Yellowknife, GMT-06:00-MDT-America/Inuvik, GMT-07:00-MST-America/Dawson_Creek, GMT-07:00-PDT-America/Vancouver, GMT-07:00-PDT-America/Whitehorse, GMT-07:00-PDT-America/Dawson, GMT+06:30-CCT-Indian/Cocos, GMT+01:00-WAT-Africa/Kinshasa, GMT+02:00-CAT-Africa/Lubumbashi, GMT+01:00-WAT-Africa/Bangui, GMT+01:00-WAT-Africa/Brazzaville, GMT+01:00-CET-Europe/Zurich, GMT+00:00-GMT-Africa/Abidjan, GMT-10:00-CKT-Pacific/Rarotonga, GMT-04:00-CLT-America/Santiago, GMT-06:00-EAST-Pacific/Easter, GMT+01:00-WAT-Africa/Douala, GMT+08:00-CST-Asia/Shanghai, GMT+08:00-CST-Asia/Harbin, GMT+08:00-CST-Asia/Chongqing, GMT+08:00-CST-Asia/Urumqi, GMT+08:00-CST-Asia/Kashgar, GMT-05:00-COT-America/Bogota, GMT-06:00-CST-America/Costa_Rica, GMT-04:00-CDT-America/Havana, GMT-01:00-CVT-Atlantic/Cape_Verde, GMT+07:00-CXT-Indian/Christmas, GMT+02:00-EET-Asia/Nicosia, GMT+01:00-CET-Europe/Prague, GMT+01:00-CET-Europe/Berlin, GMT+03:00-EAT-Africa/Djibouti, GMT+01:00-CET-Europe/Copenhagen, GMT-04:00-AST-America/Dominica, GMT-04:00-AST-America/Santo_Domingo, GMT+01:00-CET-Africa/Algiers, GMT-05:00-ECT-America/Guayaquil, GMT-06:00-GALT-Pacific/Galapagos, GMT+02:00-EET-Europe/Tallinn, GMT+02:00-EET-Africa/Cairo, GMT+00:00-WET-Africa/El_Aaiun, GMT+03:00-EAT-Africa/Asmara, GMT+01:00-CET-Europe/Madrid, GMT+01:00-CET-Africa/Ceuta, GMT+00:00-WET-Atlantic/Canary, GMT+03:00-EAT-Africa/Addis_Ababa, GMT+02:00-EET-Europe/Helsinki, GMT+12:00-FJT-Pacific/Fiji, GMT-03:00-FKST-Atlantic/Stanley,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
timezone	<String>	Read-write	GMT+11:00-KOST-Pacific/Kosrae, GMT+00:00-WET-Atlantic/Faroe, GMT+01:00-CET-Europe/Paris, GMT+01:00-WAT-Africa/Libreville, GMT+00:00-GMT-Europe/London, GMT-04:00-AST-America/Grenada, GMT+04:00-GET-Asia/Tbilisi, GMT-03:00-GFT-America/Cayenne, GMT+00:00-GMT-Europe/Guernsey, GMT+00:00-GMT-Africa/Accra, GMT+01:00-CET-Europe/Gibraltar, GMT-03:00-WGT-America/Godthab, GMT+00:00-GMT-America/Danmarkshavn, GMT-01:00-EGT-America/Scoresbysund, GMT-03:00-ADT-America/Thule, GMT+00:00-GMT-Africa/Banjul, GMT+00:00-GMT-Africa/Conakry, GMT-04:00-AST-America/Guadeloupe, GMT+01:00-WAT-Africa/Malabo, GMT+02:00-EET-Europe/Athens, GMT-02:00-GST-Atlantic/South_Georgia, GMT-06:00-CST-America/Guatemala, GMT+10:00-ChST-Pacific/Guam, GMT+00:00-GMT-Africa/Bissau, GMT-04:00-GYT-America/Guyana, GMT+08:00-HKT-Asia/Hong_Kong, GMT-06:00-CST-America/Tegucigalpa, GMT+01:00-CET-Europe/Zagreb, GMT-05:00-EST-America/Port-au-Prince, GMT+01:00-CET-Europe/Budapest, GMT+07:00-WIT-Asia/Jakarta, GMT+07:00-WIT-Asia/Pontianak, GMT+08:00-CIT-Asia/Makassar, GMT+09:00-EIT-Asia/Jayapura, GMT+00:00-GMT-Europe/Dublin, GMT+02:00-IST-Asia/Jerusalem, GMT+00:00-GMT-Europe/Isle_of_Man, GMT+05:30-IST-Asia/Kolkata, GMT+06:00-IOT-Indian/Chagos, GMT+03:00-AST-Asia/Baghdad, GMT+03:30-IRST-Asia/Tehran, GMT+00:00-GMT-Atlantic/Reykjavik, GMT+01:00-CET-Europe/Rome, GMT+00:00-GMT-Europe/Jersey, GMT-05:00-EST-America/Jamaica, GMT+02:00-EET-Asia/Amman, GMT+09:00-JST-Asia/Tokyo, GMT+03:00-EAT-Africa/Nairobi, GMT+06:00-KGT-Asia/Bishkek, GMT+07:00-ICT-Asia/Phnom_Penh, GMT+12:00-GILT-Pacific/Tarawa, GMT+13:00-PHOT-Pacific/Enderbury, GMT+14:00-LINT-Pacific/Kiritimati, GMT+03:00-EAT-Indian/Comoro, GMT-04:00-AST-America/St_Kitts, GMT+09:00-KST-Asia/Pyongyang, GMT+09:00-KST-Asia/Seoul, GMT+03:00-AST-Asia/Kuwait, GMT-05:00-EST-America/Cayman, GMT+06:00-ALMT-Asia/Almaty, GMT+06:00-QYZT-Asia/Qyzylorda, GMT+05:00-AQTT-Asia/Aqtobe, GMT+05:00-AQTT-Asia/Aqtau, GMT+05:00-ORAT-Asia/Oral, GMT+07:00-ICT-Asia/Vientiane, GMT+02:00-EET-Asia/Beirut, GMT-04:00-AST-America/St_Lucia, GMT+01:00-CET-Europe/Vaduz, GMT+05:30-IST-Asia/Colombo, GMT+00:00-GMT-Africa/Monrovia, GMT+02:00-SAST-Africa/Maseru, GMT+02:00-EET-Europe/Vilnius, GMT+01:00-CET-Europe/Luxembourg, GMT+02:00-EET-Europe/Riga, GMT+02:00-EET-Africa/Tripoli, GMT+00:00-WET-Africa/Casablanca, GMT+01:00-CET-Europe/Monaco, GMT+02:00-EET-Europe/Chisinau, GMT+01:00-CET-Europe/Podgorica, GMT-04:00-AST-America/Marigot, GMT+03:00-EAT-Indian/Antananarivo, GMT+12:00-MHT-Pacific/Majuro, GMT+12:00-MHT-Pacific/Kwajalein, GMT+01:00-CET-Europe/Skopje, GMT+00:00-GMT-Africa/Bamako, GMT+06:30-MMT-Asia/Rangoon, GMT+08:00-ULAT-Asia/Ulaanbaatar, GMT+07:00-HOVT-Asia/Hovd, GMT+08:00-CHOT-Asia/Choibalsan, GMT+08:00-CST-Asia/Macau, GMT+10:00-ChST-Pacific/Saipan, GMT-04:00-AST-America/Martinique, GMT+00:00-GMT-Africa/Nouakchott, GMT-04:00-AST-America/Montserrat, GMT+01:00-CET-Europe/Malta, GMT+04:00-MUT-Indian/Mauritius, GMT+05:00-MVT-Indian/Maldives, GMT+02:00-CAT-Africa/Blantyre, GMT-06:00-CST-America/Mexico_City, GMT-06:00-CST-America/Cancun, GMT-06:00-CST-America/Merida, GMT-06:00-CST-America/Monterrey, GMT-05:00-CDT-America/Matamoros, GMT-07:00-MST-America/Mazatlan, GMT-07:00-MST-America/Chihuahua, GMT-06:00-MDT-America/Ojinaga,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			America/Tijuana, GMT-08:00-PST-America/Santa_Isabel, GMT-06:00-CST-America/Bahia_Banderas, GMT+08:00-MYT-Asia/Kuala_Lumpur, GMT+08:00-MYT-Asia/Kuching, GMT+02:00-CAT-Africa/Maputo, GMT+02:00-WAST-Africa/Windhoek, GMT+11:00-NCT-Pacific/Noumea, GMT+01:00-WAT-Africa/Niamey, GMT+11:30-NFT-Pacific/Norfolk, GMT+01:00-WAT-Africa/Lagos, GMT-06:00-CST-America/Managua, GMT+01:00-CET-Europe/Amsterdam, GMT+01:00-CET-Europe/Oslo, GMT+05:45-NPT-Asia/Kathmandu, GMT+12:00-NRT-Pacific/Nauru, GMT-11:00-NUT-Pacific/Niue, GMT+13:00-NZDT-Pacific/Auckland, GMT+13:45-CHADT-Pacific/Chatham, GMT+04:00-GST-Asia/Muscat, GMT-05:00-EST-America/Panama, GMT-05:00-PET-America/Lima, GMT-10:00-TAHT-Pacific/Tahiti, GMT-09:30-MART-Pacific/Marquesas, GMT-09:00-GAMT-Pacific/Gambier, GMT+10:00-PGT-Pacific/Port_Moresby, GMT+08:00-PHT-Asia/Manila, GMT+05:00-PKT-Asia/Karachi, GMT+01:00-CET-Europe/Warsaw, GMT-02:00-PMDT-America/Miquelon, GMT-08:00-PST-Pacific/Pitcairn, GMT-04:00-AST-America/Puerto_Rico, GMT+02:00-EET-Asia/Gaza, GMT+02:00-EET-Asia/Hebron, GMT+00:00-WET-Europe/Lisbon, GMT+00:00-WET-Atlantic/Madeira, GMT-01:00-AZOT-Atlantic/Azores, GMT+09:00-PWT-Pacific/Palau, GMT-03:00-PYST-America/Asuncion, GMT+03:00-AST-Asia/Qatar, GMT+04:00-RET-Indian/Reunion, GMT+02:00-EET-Europe/Bucharest, GMT+01:00-CET-Europe/Belgrade, GMT+03:00-FET-Europe/Kaliningrad, GMT+04:00-MSK-Europe/Moscow, GMT+04:00-VOLT-Europe/Volgograd, GMT+04:00-SAMT-Europe/Samara, GMT+06:00-YEKT-Asia/Yekaterinburg, GMT+07:00-OMST-Asia/Omsk, GMT+07:00-NOVT-Asia/Novosibirsk, GMT+07:00-NOVT-Asia/Novokuznetsk, GMT+08:00-KRAT-Asia/Krasnoyarsk, GMT+09:00-IRKT-Asia/Irkutsk, GMT+10:00-YAKT-Asia/Yakutsk, GMT+11:00-VLAT-Asia/Vladivostok, GMT+11:00-SAKT-Asia/Sakhalin, GMT+12:00-MAGT-Asia/Magadan, GMT+12:00-PETT-Asia/Kamchatka, GMT+12:00-ANAT-Asia/Anadyr, GMT+02:00-CAT-Africa/Kigali, GMT+03:00-AST-Asia/Riyadh, GMT+11:00-SBT-Pacific/Guadalcanal, GMT+04:00-SCT-Indian/Mahe, GMT+03:00-EAT-Africa/Khartoum, GMT+01:00-CET-Europe/Stockholm, GMT+08:00-SGT-Asia/Singapore, GMT+00:00-GMT-Atlantic/St_Helena, GMT+01:00-CET-Europe/Ljubljana, GMT+01:00-CET-Arctic/Longyearbyen, GMT+01:00-CET-Europe/Bratislava, GMT+00:00-GMT-Africa/Freetown, GMT+01:00-CET-Europe/San_Marino, GMT+00:00-GMT-Africa/Dakar, GMT+03:00-EAT-Africa/Mogadishu, GMT-03:00-SRT-America/Paramaribo, GMT+00:00-GMT-Africa/Sao_Tome, GMT-06:00-CST-America/El_Salvador, GMT+02:00-EET-Asia/Damascus, GMT+02:00-SAST-Africa/Mbabane, GMT-04:00-EDT-America/Grand_Turk, GMT+01:00-WAT-Africa/Ndjamena, GMT+05:00-TFT-Indian/Kerguelen, GMT+00:00-GMT-Africa/Lome, GMT+07:00-ICT-Asia/Bangkok, GMT+05:00-TJT-Asia/Dushanbe, GMT-10:00-TKT-Pacific/Fakaofo, GMT+09:00-TLT-Asia/Dili, GMT+05:00-TMT-Asia/Ashgabat, GMT+01:00-CET-Africa/Tunis, GMT+13:00-TOT-Pacific/Tongatapu, GMT+02:00-EET-Europe/Istanbul, GMT-04:00-AST-America/Port_of_Spain, GMT+12:00-TVT-Pacific/Funafuti, GMT+08:00-CST-Asia/Taipei, GMT+03:00-EAT-Africa/Dar_es_Salaam, GMT+02:00-EET-Europe/Kiev, GMT+02:00-EET-Europe/Uzhgorod, GMT+02:00-EET-Europe/Zaporozhye, GMT+02:00-EET-Europe/Simferopol, GMT+03:00-EAT-Africa/Kampala, GMT-10:00-HST-Pacific/Johnston, GMT-11:00-SST-Pacific/Midway, GMT+12:00-WAKT-Pacific/Wake, GMT-04:00-EDT-America/New_York, GMT-04:00-EDT-America/Detroit, GMT-04:00-EDT-America/Kentucky/Louisville,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			America/Indiana/Indianapolis, GMT-04:00-EDT-America/Indiana/Vincennes, GMT-04:00-EDT-America/Indiana/Winamac, GMT-04:00-EDT-America/Indiana/Marengo, GMT-04:00-EDT-America/Indiana/Petersburg, GMT-04:00-EDT-America/Indiana/Vevay, GMT-05:00-CDT-America/Chicago, GMT-05:00-CDT-America/Indiana/Tell_City, GMT-05:00-CDT-America/Indiana/Knox, GMT-05:00-CDT-America/Menominee, GMT-05:00-CDT-America/North_Dakota/Center, GMT-05:00-CDT-America/North_Dakota/New_Salem, GMT-05:00-CDT-America/North_Dakota/Beulah, GMT-06:00-MDT-America/Denver, GMT-06:00-MDT-America/Boise, GMT-06:00-MDT-America/Shiprock, GMT-07:00-MST-America/Phoenix, GMT-07:00-PDT-America/Los_Angeles, GMT-08:00-AKDT-America/Anchorage, GMT-08:00-AKDT-America/Juneau, GMT-08:00-AKDT-America/Sitka, GMT-08:00-AKDT-America/Yakutat, GMT-08:00-AKDT-America/Nome, GMT-09:00-HADT-America/Adak, GMT-08:00-MeST-America/Metlakatla, GMT-10:00-HST-Pacific/Honolulu, GMT-03:00-UYT-America/Montevideo, GMT+05:00-UZT-Asia/Samarkand, GMT+05:00-UZT-Asia/Tashkent, GMT+01:00-CET-Europe/Vatican, GMT-04:00-AST-America/St_Vincent, GMT-04:30-VET-America/Caracas, GMT-04:00-AST-America/Tortola, GMT-04:00-AST-America/St_Thomas, GMT+07:00-ICT-Asia/Ho_Chi_Minh, GMT+11:00-VUT-Pacific/Efate, GMT+12:00-WFT-Pacific/Wallis, GMT+14:00-WSDT-Pacific/Apia, GMT+03:00-AST-Asia/Aden, GMT+03:00-EAT-Indian/Mayotte, GMT+02:00-SAST-Africa/Johannesburg, GMT+02:00-CAT-Africa/Lusaka, GMT+02:00-CAT-Africa/Harare
grantquotamaxclient	<Double>	Read-write	Percentage of shared quota to be granted at a time for maxClient. Default value: 10 Minimum value = 0 Maximum value = 100
exclusivequotamaxclient	<Double>	Read-write	Percentage of maxClient to be given to PEs. Default value: 80 Minimum value = 0 Maximum value = 100
grantquotaspillover	<Double>	Read-write	Percentage of shared quota to be granted at a time for spillover. Default value: 10 Minimum value = 0 Maximum value = 100
exclusivequotaspillover	<Double>	Read-write	Percentage of maximum limit to be given to PEs. Default value: 80 Minimum value = 0 Maximum value = 100
useproxyport	<String>	Read-write	Enable/Disable use_proxy_port setting. Default value: ENABLED Possible values = ENABLED, DISABLED
			Enables/disables the internal user from logging in to the appliance. Before disabling internal user login, you must have key-based authentication set up on the appliance. The file name for the key pair

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: ENABLED Possible values = ENABLED, DISABLED
aftpallowrandomsourceport	<String>	Read-write	Allow the FTP server to come from a random source port for active FTP data connections. Default value: DISABLED Possible values = ENABLED, DISABLED
icaports	<Integer[]>	Read-write	The ICA ports on the Web server. This allows the system to perform connection off-load for any client request that has a destination port matching one of these configured ports. Minimum value = 1
tcpcip	<String>	Read-write	Enable or disable the insertion of the client TCP/IP header in TCP payload passed from the client to one, some, or all servers attached to the system. The passed address can then be accessed through a minor modification to the server. Default value: DISABLED Possible values = ENABLED, DISABLED
servicepathingressvlan	<Double>	Read-write	VLAN on which the subscriber traffic arrives on the appliance. Minimum value = 1

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsparam":{
    "httpport":<Integer[]_value>,
    "maxconn":<Double_value>,
    "maxreq":<Double_value>,
    "cip":<String_value>,
    "cipheader":<String_value>,
    "cookieversion":<String_value>,
    "securecookie":<String_value>,
    "pmtumin":<Double_value>,
    "pmtutimeout":<Double_value>,
    "ftpportrange":<String_value>,
    "crportrange":<String_value>,
    "timezone":<String_value>,
    "grantquotamaxclient":<Double_value>,
    "exclusivequotamaxclient":<Double_value>,
    "grantquotaspillover":<Double_value>,
    "exclusivequotaspillover":<Double_value>,
    "useproxyport":<String_value>,
    "internaluserlogin":<String_value>,
    "aftpallowrandomsourceport":<String_value>,
    "icaports":<Integer[]_value>,
    "tcpcip":<String_value>,
    "servicepathingressvlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsparam?**action=unset**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsparam":{
    "ftpportrange":true,
    "crportrange":true,
    "timezone":true,
    "aftpallowrandomsourceport":true,
    "httpport":true,
    "maxconn":true,
    "maxreq":true,
    "cip":true,
    "cipheader":true,
    "cookieversion":true,
    "securecookie":true,
    "pmtumin":true,
    "pmtutimeout":true,
    "grantquotamaxclient":true,
    "exclusivequotamaxclient":true,
    "grantquotaspillover":true,
    "exclusivequotaspillover":true,
    "useproxyport":true,
    "internaluserlogin":true,
    "icaports":true,
    "tcpcip":true,
    "servicepathingressvlan":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsparam": [ {
      "httpport":<Integer[]_value>,
      "maxconn":<Double_value>,
      "maxreq":<Double_value>,
      "cip":<String_value>,
      "cipheader":<String_value>,
      "cookieversion":<String_value>,
      "securecookie":<String_value>,
      "pmtumin":<Double_value>,
      "pmtutimeout":<Double_value>,
      "ftpportrange":<String_value>,
      "crportrange":<String_value>,
      "timezone":<String_value>,
      "grantquotamaxclient":<Double_value>,
      "exclusivequotamaxclient":<Double_value>,
      "grantquotaspillover":<Double_value>,
      "exclusivequotaspillover":<Double_value>,
      "useproxyport":<String_value>,
      "internaluserlogin":<String_value>,
      "aftpallowrandomsourceport":<String_value>,
      "icaports":<Integer[]_value>,
      "tcpcip":<String_value>,
      "servicepathingressvlan":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nspartition

May 20, 2016

Configuration for admin partition resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
partitionname	<String>	Read-write	Name of the Partition. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
maxbandwidth	<Double>	Read-write	Maximum bandwidth in Kbps, permitted on this partition. Default value: 10240
minbandwidth	<Double>	Read-write	Minimum bandwidth in Kbps, permitted on this partition. Default value: 10240
maxconn	<Double>	Read-write	Maximum number of connections permitted on this partition. Default value: 1024
maxmemlimit	<Double>	Read-write	Maximum memory allowed for this partition in megabytes. Default value: 10 Minimum value = 5
partitionid	<Double>	Read-only	Partition Id. Minimum value = 1
partitiontype	<String>	Read-only	Type of the Partition. Possible values = Default Partition, Current Partition
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | SWITCH | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspartition":{
    "partitionname":<String_value>,
    "maxbandwidth":<Double_value>,
    "minbandwidth":<Double_value>,
    "maxconn":<Double_value>,
    "maxmemlimit":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition/partitionname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspartition":{
    "partitionname":<String_value>,
    "maxbandwidth":<Double_value>,
    "minbandwidth":<Double_value>,
    "maxconn":<Double_value>,
    "maxmemlimit":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"nspartition":{
    "partitionname":<String_value>,
    "maxbandwidth":true,
    "minbandwidth":true,
    "maxconn":true,
    "maxmemlimit":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Switch

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition?**action=Switch**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspartition":{
    "partitionname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nspartition?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nspartition?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nspartition resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nspartition?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nspartition?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nspartition resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspartition": [ {
    "partitionname":<String_value>,
    "partitionid":<Double_value>,
    "partitiontype":<String_value>,
    "maxbandwidth":<Double_value>,
    "minbandwidth":<Double_value>,
    "maxconn":<Double_value>,
    "maxmemlimit":<Double_value>
}]}
```

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition/partitionname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nspartition/partitionname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nspartition/partitionname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspartition": [ {
      "partitionname":<String_value>,
      "partitionid":<Double_value>,
      "partitiontype":<String_value>,
      "maxbandwidth":<Double_value>,
      "minbandwidth":<Double_value>,
      "maxconn":<Double_value>,
      "maxmemlimit":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "nspartition": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nspartition_binding

Jul 07, 2017

Binding object which returns the resources bound to nspartition.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
partitionname	<String>	Read-write	Name of partition for which to display parameters. Minimum length = 1
nspartition_bridgegroup_binding	<nspartition_bridgegroup_binding[]>	Read-only	bridgegroup that can be bound to nspartition.
nspartition_vlan_binding	<nspartition_vlan_binding[]>	Read-only	vlan that can be bound to nspartition.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition_binding/partitionname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "nspartition_binding": [ {
    "partitionname":<String_value>,
    "nspartition_vlan_binding":<nspartition_vlan_binding[]_value>,
    "nspartition_bridgegroup_binding":<nspartition_bridgegroup_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nspartition_bridgegroup_binding

May 20, 2016

Binding object showing the bridgegroup that can be bound to nspartition.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
bridgegroup	<Double>	Read-write	Identifier of the bridge group that is assigned to this partition. Minimum value = 1 Maximum value = 1000
partitionname	<String>	Read-write	Name of the Partition. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/nspartition_bridgegroup_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"nspartition_bridgegroup_binding":{
    "partitionname":<String_value>,
    "bridgegroup":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nspartition_bridgegroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nspartition_bridgegroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspartition_bridgegroup_binding": [ {
      "bridgegroup":<Double_value>,
      "partitionname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition_bridgegroup_binding/partitionname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"nspartition_bridgegroup_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nspartition_vlan_binding

Jul 07, 2017

Binding object showing the vlan that can be bound to nspartition.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vlan	<Double>	Read-write	Identifier of the vlan that is assigned to this partition. Minimum value = 1 Maximum value = 4094
partitionname	<String>	Read-write	Name of the Partition. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/nspartition_vlan_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"nspartition_vlan_binding":{
    "partitionname":<String_value>,
    "vlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nspartition_vlan_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nspartition_vlan_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspartition_vlan_binding": [ {
    "vlan":<Double_value>,
    "partitionname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nspartition_vlan_binding/partitionname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"nspartition_vlan_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nspbr

May 20, 2016

Configuration for Policy Based Routing(PBR) entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the PBR. Must begin with an ASCII alphabetic or underscore \(_\) character, and must contain only ASCII alphanumeric, underscore, hash \(\#\), period \(.\), space, colon \(:\), at \(@\), equals \(=\), and hyphen \(-\) characters. Cannot be changed after the PBR is created. Minimum length = 1
action	<String>	Read-write	Action to perform on the outgoing IPv4 packets that match the PBR. Available settings function as follows: * ALLOW - The NetScaler appliance sends the packet to the designated next-hop router. * DENY - The NetScaler appliance applies the routing table for normal destination-based routing. Possible values = ALLOW, DENY
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcip	<Boolean>	Read-write	IP address or range of IP addresses to match against the source IP address of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [10.102.29.30-10.102.29.189].
srcipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
srcipval	<String>	Read-write	IP address or range of IP addresses to match against the source IP address of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [10.102.29.30-10.102.29.189].
srcport	<Boolean>	Read-write	Port number or range of port numbers to match against the source port number of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90]. Note: The destination port can be specified only for TCP and UDP protocols.
			Logical operator.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
srcportval	<String>	Read-write	Port number or range of port numbers to match against the source port number of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90]. Note: The destination port can be specified only for TCP and UDP protocols. Maximum length = 65535
destip	<Boolean>	Read-write	IP address or range of IP addresses to match against the destination IP address of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [10.102.29.30-10.102.29.189].
destipop	<String>	Read-write	Logical operator. Possible values = =, !=, EQ, NEQ
destipval	<String>	Read-write	IP address or range of IP addresses to match against the destination IP address of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [10.102.29.30-10.102.29.189].
destport	<Boolean>	Read-write	Port number or range of port numbers to match against the destination port number of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90]. Note: The destination port can be specified only for TCP and UDP protocols.
destportop	<String>	Read-write	Logical operator. Possible values = =, !=, EQ, NEQ
destportval	<String>	Read-write	Port number or range of port numbers to match against the destination port number of an outgoing IPv4 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90]. Note: The destination port can be specified only for TCP and UDP protocols. Maximum length = 65535
nexthop	<Boolean>	Read-write	IP address of the next hop router or the name of the link load balancing virtual server to which to send matching packets if action is set to ALLOW. If you specify a link load balancing (LLB) virtual server, which can provide a backup if a next hop link fails, first make sure that the next hops bound to the LLB virtual server are actually next hops that are directly connected to the NetScaler appliance. Otherwise, the NetScaler throws an error when you attempt to create the PBR.
nexthopval	<String>	Read-write	The Next Hop IP address or gateway name.
iptunnel	<Boolean>	Read-write	The Tunnel name.
iptunnelname	<String>	Read-write	The iptunnel name where packets need to be forwarded upon.
srcmac	<String>	Read-write	MAC address to match against the source MAC address of an outgoing IPv4 packet.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
srcmacmask	<String>	Read-write	exact match and 1s for wildcard. For matching first 3 bytes of MAC address, srcMacMask value "000000111111". . Default value: "000000000000"
protocol	<String>	Read-write	Protocol, identified by protocol name, to match against the protocol of an outgoing IPv4 packet. Possible values = ICMP, IGMP, TCP, EGP, IGP, ARGUS, UDP, RDP, RSVP, EIGRP, L2TP, ISIS
protocolnumber	<Double>	Read-write	Protocol, identified by protocol number, to match against the protocol of an outgoing IPv4 packet. Minimum value = 1 Maximum value = 255
vlan	<Double>	Read-write	ID of the VLAN. The NetScaler appliance compares the PBR only to the outgoing packets on the specified VLAN. If you do not specify any interface ID, the appliance compares the PBR to the outgoing packets on all VLANs. Minimum value = 1 Maximum value = 4094
vxlan	<Double>	Read-write	ID of the VXLAN. The NetScaler appliance compares the PBR only to the outgoing packets on the specified VXLAN. If you do not specify any interface ID, the appliance compares the PBR to the outgoing packets on all VXLANs. Minimum value = 1 Maximum value = 16777215
Interface	<String>	Read-write	ID of an interface. The NetScaler appliance compares the PBR only to the outgoing packets on the specified interface. If you do not specify any value, the appliance compares the PBR to the outgoing packets on all interfaces.
priority	<Double>	Read-write	Priority of the PBR, which determines the order in which it is evaluated relative to the other PBRs. If you do not specify priorities while creating PBRs, the PBRs are evaluated in the order in which they are created. Minimum value = 1 Maximum value = 81920
msr	<String>	Read-write	Monitor the route specified byte Next Hop parameter. This parameter is not applicable if you specify a link load balancing (LLB) virtual server name with the Next Hop parameter. Default value: DISABLED Possible values = ENABLED, DISABLED
monitor	<String>	Read-write	The name of the monitor.(Can be only of type ping or ARP). Minimum length = 1
state	<String>	Read-write	Enable or disable the PBR. After you apply the PBRs, the NetScaler appliance compares outgoing packets to the enabled PBRs. Default value: ENABLED Possible values = ENABLED, DISABLED
detail	<Boolean>	Read-write	To get a detailed view.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
kernelstate	<String>	Read-only	The commit status of the PBR. Default value: NOT APPLIED Possible values = APPLIED, NOT APPLIED, RE-APPLY, SFAPPLIED, SFNOT APPLIED, SFAPPLIED61, SFNOT APPLIED61
curstate	<Double>	Read-only	If this route is UP/DOWN.
totalprobes	<Double>	Read-only	The total number of probes sent.
totalfailedprobes	<Double>	Read-only	The total number of failed probes.
failedprobes	<Double>	Read-only	Number of the current failed monitoring probes.
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
data	<Boolean>	Read-only	Internal data of this route.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr":{
    "name":<String_value>,
    "action":<String_value>,
    "td":<Double_value>,
    "srcip":<Boolean_value>,
    "srcipop":<String_value>,
    "srcipval":<String_value>,
    "srcport":<Boolean_value>,
    "srcportop":<String_value>,
    "srcportval":<String_value>,
    "destip":<Boolean_value>,
    "destipop":<String_value>,
    "destipval":<String_value>,
    "destport":<Boolean_value>,
    "destportop":<String_value>,
    "destportval":<String_value>,
    "nexthop":<Boolean_value>,
    "nexthopval":<String_value>,
    "iptunnel":<Boolean_value>,
    "iptunnelname":<String_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "Interface":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "msr":<String_value>,
    "monitor":<String_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr":{
    "name":<String_value>,
    "action":<String_value>,
    "srcip":<Boolean_value>,
    "srcipop":<String_value>,
    "srcipval":<String_value>,
    "srcport":<Boolean_value>,
    "srcportop":<String_value>,
    "srcportval":<String_value>,
    "destip":<Boolean_value>,
    "destipop":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "destipval":<String_value>,
    "destport":<Boolean_value>,
    "destportop":<String_value>,
    "destportval":<String_value>,
    "nexthop":<Boolean_value>,
    "nexthopval":<String_value>,
    "iptunnel":<Boolean_value>,
    "iptunnelname":<String_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "Interface":<String_value>,
    "priority":<Double_value>,
    "msr":<String_value>,
    "monitor":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr":{
    "name":<String_value>,
    "srcip":true,
    "srcport":true,
    "destip":true,
    "destport":true,
    "nexthop":true,
    "iptunnel":true,
    "srcmac":true,
    "srcmacmask":true,
    "protocol":true,
    "vlan":true,
    "vxlan":true,
    "Interface":true,
    "msr":true,
    "monitor":true
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/nspbr?**args=name:<String_value>,detail:<Boolean_value>**

Use this query-parameter to get nspbr resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/nspbr?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nspbr?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nspbr resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nspbr?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nspbr?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nspbr resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspbr": [ {
name:<String_value>,detail:<Boolean_value>      "action":<String_value>,
      "srcmac":<String_value>,
      "srcmacmask":<String_value>,
      "protocol":<String_value>,
      "protocolnumber":<Double_value>,
      "srcportval":<String_value>,
      "td":<Double_value>,
      "destportval":<String_value>,
      "srcipval":<String_value>,
      "destipval":<String_value>,
      "vlan":<Double_value>,
      "vxlan":<Double_value>,
      "state":<String_value>,
      "Interface":<String_value>,
      "hits":<Double_value>,
      "priority":<Double_value>,
      "srcipop":<String_value>,
      "destipop":<String_value>,
      "srcportop":<String_value>,
      "destportop":<String_value>,
      "kernelstate":<String_value>,
      "nexthopval":<String_value>,
      "iptunnelname":<String_value>,
      "msr":<String_value>,
      "monitor":<String_value>,
      "curstate":<Double_value>,
      "totalprobes":<Double_value>,
      "totalfailedprobes":<Double_value>,
      "failedprobes":<Double_value>,
      "monstatcode":<Integer_value>,
      "monstatparam1":<Integer_value>,
      "monstatparam2":<Integer_value>,
      "monstatparam3":<Integer_value>,
      "data":<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nspbr/name_value<String>?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nspbr/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspbr": [ {
name:<String_value>,detail:<Boolean_value>      "action":<String_value>,
      "srcmac":<String_value>,
      "srcmacmask":<String_value>,
      "protocol":<String_value>,
      "protocolnumber":<Double_value>,
      "srcportval":<String_value>,
      "td":<Double_value>,
      "destportval":<String_value>,
      "srcipval":<String_value>,
      "destipval":<String_value>,
      "vlan":<Double_value>,
      "vxlan":<Double_value>,
      "state":<String_value>,
      "Interface":<String_value>,
      "hits":<Double_value>,
      "priority":<Double_value>,
      "srcipop":<String_value>,
      "destipop":<String_value>,
      "srcportop":<String_value>,
      "destportop":<String_value>,
      "kernelstate":<String_value>,
      "nexthopval":<String_value>,
      "iptunnelname":<String_value>,
      "msr":<String_value>,
      "monitor":<String_value>,
      "curstate":<Double_value>,
      "totalprobes":<Double_value>,
      "totalfailedprobes":<Double_value>,
      "failedprobes":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "monstatcode":<Integer_value>,
    "monstatparam1":<Integer_value>,
    "monstatparam2":<Integer_value>,
    "monstatparam3":<Integer_value>,
    "data":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspbr": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nspbr6

May 20, 2016

Configuration for PBR6 entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the PBR6. Must begin with an ASCII alphabetic or underscore \(_\) character, and must contain only ASCII alphanumeric, underscore, hash \(\#\), period \(.\), space, colon \(:\), at \(@\), equals \(=\), and hyphen \(-\) characters. Cannot be changed after the PBR6 is created. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
action	<String>	Read-write	Action to perform on the outgoing IPv6 packets that match the PBR6. Available settings function as follows: * ALLOW - The NetScaler appliance sends the packet to the designated next-hop router. * DENY - The NetScaler appliance applies the routing table for normal destination-based routing. Possible values = ALLOW, DENY
srcipv6	<Boolean>	Read-write	IP address or range of IP addresses to match against the source IP address of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen and enclose within brackets.
srcipop	<String>	Read-write	Either the equals (=) or does not equal (!=) logical operator. Possible values = =, !=, EQ, NEQ
srcipv6val	<String>	Read-write	IP address or range of IP addresses to match against the source IP address of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen and enclose within brackets.
srcport	<Boolean>	Read-write	Port number or range of port numbers to match against the source port number of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90].
srcportop	<String>	Read-write	Logical operator. Possible values = =, !=, EQ, NEQ

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum length = 65535
destipv6	<Boolean>	Read-write	IP address or range of IP addresses to match against the destination IP address of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen and enclose within brackets.
destipop	<String>	Read-write	Logical operator. Possible values = =, !=, EQ, NEQ
destipv6val	<String>	Read-write	IP address or range of IP addresses to match against the destination IP address of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen and enclose within brackets.
destport	<Boolean>	Read-write	Port number or range of port numbers to match against the destination port number of an outgoing IPv6 packet. In the command line interface, separate the range with a hyphen and enclose within brackets. For example: [40-90]. Note: The destination port can be specified only for TCP and UDP protocols.
destportop	<String>	Read-write	Logical operator. Possible values = =, !=, EQ, NEQ
destportval	<String>	Read-write	Destination port (range). Maximum length = 65535
srcmac	<String>	Read-write	MAC address to match against the source MAC address of an outgoing IPv6 packet.
srcmacmask	<String>	Read-write	Used to define range of Source MAC address. It takes string of 0 and 1, 0s are for exact match and 1s for wildcard. For matching first 3 bytes of MAC address, srcMacMask value "000000111111". . Default value: "000000000000"
protocol	<String>	Read-write	Protocol, identified by protocol name, to match against the protocol of an outgoing IPv6 packet. Possible values = ICMPV6, TCP, UDP
protocolnumber	<Double>	Read-write	Protocol, identified by protocol number, to match against the protocol of an outgoing IPv6 packet. Minimum value = 1 Maximum value = 255
vlan	<Double>	Read-write	ID of the VLAN. The NetScaler appliance compares the PBR6 only to the outgoing packets on the specified VLAN. If you do not specify an interface ID, the appliance compares the PBR6 to the outgoing packets on all VLANs. Minimum value = 1 Maximum value = 4094
vxlan	<Double>	Read-write	ID of the VXLAN. The NetScaler appliance compares the PBR6 only to the outgoing packets on the specified VXLAN. If you do not specify an interface ID, the appliance compares the PBR6 to the outgoing packets on all VXLANs. Minimum value = 1 Maximum value = 16777215

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Interface	<String>	Read-write	packets on the specified interface. If you do not specify a value, the appliance compares the PBR6 to the outgoing packets on all interfaces.
Name	**Data Type**	**Permissions**	**Description**
priority	<Double>	Read-write	Priority of the PBR6, which determines the order in which it is evaluated relative to the other PBR6s. If you do not specify priorities while creating PBR6s, the PBR6s are evaluated in the order in which they are created. Minimum value = 1 Maximum value = 80000
state	<String>	Read-write	Enable or disable the PBR6. After you apply the PBR6s, the NetScaler appliance compares outgoing packets to the enabled PBR6s. Default value: ENABLED Possible values = ENABLED, DISABLED
msr	<String>	Read-write	Monitor the route specified by the Next Hop parameter. Default value: DISABLED Possible values = ENABLED, DISABLED
monitor	<String>	Read-write	The name of the monitor.(Can be only of type ping or ARP). Minimum length = 1
nexthop	<Boolean>	Read-write	IP address of the next hop router to which to send matching packets if action is set to ALLOW. This next hop should be directly reachable from the appliance.
nexthopval	<String>	Read-write	The Next Hop IPv6 address.
iptunnel	<String>	Read-write	The iptunnel name where packets need to be forwarded upon.
nexthopvlan	<Double>	Read-write	VLAN number to be used for link local nexthop . Minimum value = 1 Maximum value = 4094
detail	<Boolean>	Read-write	To get a detailed view.
kernelstate	<String>	Read-only	Commit status of the PBR6. Default value: NOT APPLIED Possible values = APPLIED, NOT APPLIED, RE-APPLY, SFAPPLIED, SFNOT APPLIED
hits	<Double>	Read-only	Number of hits of this PBR6.
curstate	<Double>	Read-only	If this route is UP/DOWN.
totalprobes	<Double>	Read-only	The total number of probes sent.
totalfailedprobes	<Double>	Read-only	The total number of failed probes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
monstatcode	<Integer>	Read-only	The code indicating the monitor response.
monstatparam1	<Integer>	Read-only	First parameter for use with message code.
monstatparam2	<Integer>	Read-only	Second parameter for use with message code.
monstatparam3	<Integer>	Read-only	Third parameter for use with message code.
data	<Boolean>	Read-only	Internal data of this route.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | RENUMBER | DELETE | UPDATE | UNSET | ENABLE | DISABLE | CLEAR | APPLY | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr6":{
    "name":<String_value>,
    "td":<Double_value>,
    "action":<String_value>,
    "srcipv6":<Boolean_value>,
    "srcipop":<String_value>,
    "srcipv6val":<String_value>,
    "srcport":<Boolean_value>,
    "srcportop":<String_value>,
    "srcportval":<String_value>,
    "destipv6":<Boolean_value>,
    "destipop":<String_value>,
    "destipv6val":<String_value>,
    "destport":<Boolean_value>,
    "destportop":<String_value>,
    "destportval":<String_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "Interface":<String_value>,
    "priority":<Double_value>,
    "state":<String_value>,
    "msr":<String_value>,
    "monitor":<String_value>,
    "nexthop":<Boolean_value>,
    "nexthopval":<String_value>,
    "iptunnel":<String_value>,
    "nexthopvlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

renumber

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**action=renumber**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr6":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr6":{
    "name":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "action":<String_value>,
    "srcipv6":<Boolean_value>,
    "srcipop":<String_value>,
    "srcipv6val":<String_value>,
    "srcport":<Boolean_value>,
    "srcportop":<String_value>,
    "srcportval":<String_value>,
    "destipv6":<Boolean_value>,
    "destipop":<String_value>,
    "destipv6val":<String_value>,
    "destport":<Boolean_value>,
    "destportop":<String_value>,
    "destportval":<String_value>,
    "srcmac":<String_value>,
    "srcmacmask":<String_value>,
    "protocol":<String_value>,
    "protocolnumber":<Double_value>,
    "vlan":<Double_value>,
    "vxlan":<Double_value>,
    "Interface":<String_value>,
    "priority":<Double_value>,
    "msr":<String_value>,
    "monitor":<String_value>,
    "nexthop":<Boolean_value>,
    "nexthopval":<String_value>,
    "iptunnel":<String_value>,
    "nexthopvlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr6":{
    "name":<String_value>,
    "srcipv6":true,
    "srcport":true,
    "destipv6":true,
    "destport":true,
    "srcmac":true,
    "srcmacmask":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "protocol":true,
    "Interface":true,
    "vlan":true,
    "vxlan":true,
    "msr":true,
    "monitor":true,
    "nexthop":true,
    "iptunnel":true,
    "nexthopvlan":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr6":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"nspbr6":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr6":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

apply

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**action=apply**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbr6":{
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**args=name:<String_value>,detail:<Boolean_value>**

Use this query-parameter to get nspbr6 resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nspbr6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nspbr6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspbr6": [ {
name:<String_value>,detail:<Boolean_value>      "td":<Double_value>,
      "action":<String_value>,
      "srcmac":<String_value>,
      "srcmacmask":<String_value>,
      "protocol":<String_value>,
      "protocolnumber":<Double_value>,
      "srcportval":<String_value>,
      "destportval":<String_value>,
      "srcipv6val":<String_value>,
      "destipv6val":<String_value>,
      "vlan":<Double_value>,
      "vxlan":<Double_value>,
      "state":<String_value>,
      "kernelstate":<String_value>,
      "Interface":<String_value>,
      "hits":<Double_value>,
      "priority":<Double_value>,
      "srcipop":<String_value>,
      "destipop":<String_value>,
      "srcportop":<String_value>,
      "destportop":<String_value>,
      "msr":<String_value>,
      "monitor":<String_value>,
      "curstate":<Double_value>,
      "totalprobes":<Double_value>,
      "totalfailedprobes":<Double_value>,
      "failedprobes":<Double_value>,
      "monstatcode":<Integer_value>,
      "monstatparam1":<Integer_value>,
      "monstatparam2":<Integer_value>,
      "monstatparam3":<Integer_value>,
      "nexthopval":<String_value>,
      "iptunnel":<String_value>,
      "nexthopvlan":<Double_value>,
      "data":<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nspbr6/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nspbr6/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspbr6": [ {
name:<String_value>,detail:<Boolean_value>      "td":<Double_value>,
      "action":<String_value>,
      "srcmac":<String_value>,
      "srcmacmask":<String_value>,
      "protocol":<String_value>,
      "protocolnumber":<Double_value>,
      "srcportval":<String_value>,
      "destportval":<String_value>,
      "srcipv6val":<String_value>,
      "destipv6val":<String_value>,
      "vlan":<Double_value>,
      "vxlan":<Double_value>,
      "state":<String_value>,
      "kernelstate":<String_value>,
      "Interface":<String_value>,
      "hits":<Double_value>,
      "priority":<Double_value>,
      "srcipop":<String_value>,
      "destipop":<String_value>,
      "srcportop":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "destportop":<String_value>,
    "msr":<String_value>,
    "monitor":<String_value>,
    "curstate":<Double_value>,
    "totalprobes":<Double_value>,
    "totalfailedprobes":<Double_value>,
    "failedprobes":<Double_value>,
    "monstatcode":<Integer_value>,
    "monstatparam1":<Integer_value>,
    "monstatparam2":<Integer_value>,
    "monstatparam3":<Integer_value>,
    "nexthopval":<String_value>,
    "iptunnel":<String_value>,
    "nexthopvlan":<Double_value>,
    "data":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbr6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspbr6": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nspbrs

May 20, 2016

Configuration for Policy based routing resource.

Operations (click to see Properties)

RENUMBER | CLEAR | APPLY

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

renumber

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbrs?**action=renumber**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"nspbrs":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbrs?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbrs":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

apply

URL: http://<netscaler-ip-address>/nitro/v1/config/nspbrs?**action=apply**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nspbrs":{
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsratecontrol

May 20, 2016

Configuration for rate control resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
tcpthreshold	<Double>	Read-write	Number of SYNs permitted per 10 milliseconds.
udpthreshold	<Double>	Read-write	Number of UDP packets permitted per 10 milliseconds.
icmpthreshold	<Double>	Read-write	Number of ICMP packets permitted per 10 milliseconds. Default value: 100
tcprstthreshold	<Double>	Read-write	The number of TCP RST packets permitted per 10 milli second. zero means rate control is disabled and 0xffffffff means every thing is rate controlled. Default value: 100

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsratecontrol

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsratecontrol":{
    "tcpthreshold":<Double_value>,
    "udpthreshold":<Double_value>,
    "icmpthreshold":<Double_value>,
    "tcprstthreshold":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsratecontrol?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsratecontrol":{
    "tcpthreshold":true,
    "udpthreshold":true,
    "icmpthreshold":true,
    "tcprstthreshold":true
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsratecontrol

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsratecontrol": [ {
      "tcpthreshold":<Double_value>,
      "udpthreshold":<Double_value>,
      "icmpthreshold":<Double_value>,
      "tcprstthreshold":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsrollbackcmd

May 20, 2016

Configuration for Generate rollback commands resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
filename	<String>	Read-write	File that contains the commands for which the rollback commands must be generated. Specify the full path of the file name.
outtype	<String>	Read-write	Format in which the rollback commands must be generated. Possible values = cli, xml

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsrollbackcmd

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/nsrollbackcmd?**args=filename:<String_value>,outtype:<String_value>**

Use this query-parameter to get nsrollbackcmd resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsrollbackcmd": [ {
filename:<String_value>,outtype:<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsrpcnode

May 20, 2016

Configuration for rpc node resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipaddress	<String>	Read-write	IP address of the node. This has to be in the same subnet as the NSIP address. Minimum length = 1
password	<String>	Read-write	Password to be used in authentication with the peer system node. Minimum length = 1
srcip	<String>	Read-write	Source IP address to be used to communicate with the peer system node. The default value is 0, which means that the appliance uses the NSIP address as the source IP address.
secure	<String>	Read-write	State of the channel when talking to the node. Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsrpcnode":{
    "ipaddress":<String_value>,
    "password":<String_value>,
    "srcip":<String_value>,
    "secure":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsrpcnode":{
    "ipaddress":<String_value>,
    "password":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "srcip":true,
    "secure":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsrpcnode resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsrpcnode resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsrpcnode": [ {
      "ipaddress":<String_value>,
      "password":<String_value>,
      "srcip":<String_value>,
      "secure":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode/ipaddress_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode/ipaddress_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode/ipaddress_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "nsrpcnode": [ {
      "ipaddress":<String_value>,
      "password":<String_value>,
      "srcip":<String_value>,
      "secure":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsrpcnode?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsrpcnode": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsrunningconfig

May 20, 2016

Configuration for running configuration resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
withdefaults	<Boolean>	Read-write	Include default values of parameters that have not been explicitly configured. If this argument is disabled, such parameters are not included.
response	<String>	Read-only	running config data as text blob.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsrunningconfig

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/nsrunningconfig?**args=withdefaults:<Boolean_value>**

Use this query-parameter to get nsrunningconfig resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsrunningconfig": [ {
withdefaults:<Boolean_value>      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nssavedconfig

Jul 07, 2017

Configuration for last saved configuration resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
textblob	<String>	Read-only	Text of the last saved configuration.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nssavedconfig

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nssavedconfig": [ {
      "textblob":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nservicefunction

May 20, 2016

Configuration for service Function resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicefunctionname	<String>	Read-write	Name of the service function to be created. Leading character must be a number or letter. Other characters allowed, after the first character, are @ _ - . (period) : (colon) # and space ( ). Minimum length = 1
ingressvlan	<Double>	Read-write	VLAN ID on which the traffic from service function reaches Netscaler. Minimum value = 1 Maximum value = 4094
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsservicefunction":{
    "servicefunctionname":<String_value>,
    "ingressvlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsservicefunction":{
    "servicefunctionname":<String_value>,
    "ingressvlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction/servicefunctionname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsservicefunction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsservicefunction resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsservicefunction": [ {
      "servicefunctionname":<String_value>,
      "ingressvlan":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction/servicefunctionname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction/servicefunctionname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction/servicefunctionname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsservicefunction": [ {
      "servicefunctionname":<String_value>,
      "ingressvlan":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicefunction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsservicefunction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsservicepath

May 20, 2016

Configuration for service Chain resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicepathname	<String>	Read-write	Name for the Service path. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicepath

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsservicepath":{
    "servicepathname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicepath/servicepathname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicepath

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsservicepath?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsservicepath?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsservicepath resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsservicepath?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsservicepath?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsservicepath resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsservicepath": [ {
      "servicepathname":<String_value>,

}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicepath/servicepathname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsservicepath/servicepathname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nsservicepath/servicepathname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsservicepath": [ {
      "servicepathname":<String_value>,

}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicepath?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsservicepath": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsservicepath_binding

May 20, 2016

Binding object which returns the resources bound to nsservicepath.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicepathname	<String>	Read-write	Name for the Service path. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
nsservicepath_nsservicefunction_binding	<nsservicepath_nsservicefunction_binding[]>	Read-only	nsservicefunction that can be bound to nsservicepath.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicepath_binding/servicepathname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "nsservicepath_binding": [ {
      "servicepathname":<String_value>,
      "nsservicepath_nsservicefunction_binding":<nsservicepath_nsservicefunction_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nssimpleacl

May 20, 2016

Configuration for simple ACL resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
aclname	<String>	Read-write	Name for the simple ACL rule. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the simple ACL rule is created. Minimum length = 1
aclaction	<String>	Read-write	Drop incoming IPv4 packets that match the simple ACL rule. Possible values = DENY
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
srcip	<String>	Read-write	IP address to match against the source IP address of an incoming IPv4 packet.
destport	<Integer>	Read-write	Port number to match against the destination port number of an incoming IPv4 packet. Omitting the port number creates an all-ports simple ACL rule, which matches any port. In that case, you cannot create another simple ACL rule specifying a specific port and the same source IPv4 address.
protocol	<String>	Read-write	Protocol to match against the protocol of an incoming IPv4 packet. You must set this parameter if you have set the Destination Port parameter. Possible values = TCP, UDP
ttl	<Double>	Read-write	Number of seconds, in multiples of four, after which the simple ACL rule expires. If you do not want the simple ACL rule to expire, do not specify a TTL value. Minimum value = 4 Maximum value = 2147483647
estsessions	<Boolean>	Read-write	.
hits	<Double>	Read-only	Number of hits for this ACL rule.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD | CLEAR | DELETE | FLUSH | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nssimpleacl":{
    "aclname":<String_value>,
    "aclaction":<String_value>,
    "td":<Double_value>,
    "srcip":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "destport":<Integer_value>,
    "protocol":<String_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nssimpleacl":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl/aclname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

flush

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl?**action=flush**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nssimpleacl":{
    "estsessions":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nssimpleacl resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl?**pagesize=#no&pageno=#no**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the nssimpleacl resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nssimpleacl": [ {
    "aclname":<String_value>,
    "aclaction":<String_value>,
    "td":<Double_value>,
    "srcip":<String_value>,
    "destport":<Integer_value>,
    "protocol":<String_value>,
    "ttl":<Double_value>,
    "hits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl/aclname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl/aclname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl/aclname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nssimpleacl": [ {
      "aclname":<String_value>,
      "aclaction":<String_value>,
      "td":<Double_value>,
      "srcip":<String_value>,
      "destport":<Integer_value>,
      "protocol":<String_value>,
      "ttl":<Double_value>,
      "hits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "nssimpleacl": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsservicepath_nsservicefunction_binding

May 20, 2016

Binding object showing the nsservicefunction that can be bound to nsservicepath.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicepathname	<String>	Read-write	Name for the Service path. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1
servicefunction	<String>	Read-write	List of service functions constituting the chain. Minimum length = 1
index	<Double>	Read-write	The serviceindex of each servicefunction in path. Minimum value = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/nsservicepath_nsservicefunction_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"nsservicepath_nsservicefunction_binding":{
    "servicepathname":<String_value>,
    "servicefunction":<String_value>,
    "index":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of nsservicepath_nsservicefunction_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsservicepath_nsservicefunction_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsservicepath_nsservicefunction_binding": [ {
      "servicepathname":<String_value>,
      "servicefunction":<String_value>,
      "index":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsservicepath_nsservicefunction_binding/servicepathname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{"nsservicepath_nsservicefunction_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nssimpleacl6

May 20, 2016

Configuration for simple ACL6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
aclname	<String>	Read-write	Name for the simple ACL6 rule. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the simple ACL6 rule is created. Minimum length = 1
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
aclaction	<String>	Read-write	Drop incoming IPv6 packets that match the simple ACL6 rule. Possible values = DENY
srcipv6	<String>	Read-write	IP address to match against the source IP address of an incoming IPv6 packet.
destport	<Integer>	Read-write	Port number to match against the destination port number of an incoming IPv6 packet. Omitting the port number creates an all-ports simple ACL6 rule, which matches any port. In that case, you cannot create another simple ACL6 rule specifying a specific port and the same source IPv6 address.
protocol	<String>	Read-write	Protocol to match against the protocol of an incoming IPv6 packet. You must set this parameter if you set the Destination Port parameter. Possible values = TCP, UDP
ttl	<Double>	Read-write	Number of seconds, in multiples of four, after which the simple ACL6 rule expires. If you do not want the simple ACL6 rule to expire, do not specify a TTL value. Minimum value = 4 Maximum value = 2147483647
estsessions	<Boolean>	Read-write	.
hits	<Double>	Read-only	Number of hits for this SACL6 rule.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD | CLEAR | FLUSH | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nssimpleacl6":{
    "aclname":<String_value>,
    "td":<Double_value>,
    "aclaction":<String_value>,
    "srcipv6":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "destport":<Integer_value>,
    "protocol":<String_value>,
    "ttl":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nssimpleacl6":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

flush

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6?**action=flush**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nssimpleacl6":{
    "estsessions":<Boolean_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6/aclname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nssimpleacl6 resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nssimpleacl6 resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nssimpleacl6": [ {
      "aclname":<String_value>,
      "aclaction":<String_value>,
      "td":<Double_value>,
      "srcipv6":<String_value>,
      "destport":<Integer_value>,
      "protocol":<String_value>,
      "ttl":<Double_value>,
      "hits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6/aclname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6/aclname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6/aclname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nssimpleacl6": [ {
      "aclname":<String_value>,
      "aclaction":<String_value>,
      "td":<Double_value>,
      "srcipv6":<String_value>,
      "destport":<Integer_value>,
      "protocol":<String_value>,
      "ttl":<Double_value>,
      "hits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nssimpleacl6?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{ "nssimpleacl6": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsspparams

May 20, 2016

Configuration for Surge Protection parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
basethreshold	<Integer>	Read-write	Maximum number of server connections that can be opened before surge protection is activated. Default value: 200 Minimum value = 0 Maximum value = 32767
throttle	<String>	Read-write	Rate at which the system opens connections to the server. Default value: Normal Possible values = Aggressive, Normal, Relaxed
table0	<Integer[]>	Read-only	Table.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsspparams

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsspparams":{
    "basethreshold":<Integer_value>,
    "throttle":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsspparams?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsspparams":{
    "basethreshold":true,
    "throttle":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsspparams

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsspparams": [ {
      "basethreshold":<Integer_value>,
      "throttle":<String_value>,
      "table0":<Integer[]_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsstats

Jul 07, 2017

Configuration for stats resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cleanuplevel	<String>	Read-write	The level of stats to be cleared. 'global' option will clear global counters only, 'all' option will clear all device counters also along with global counters. For both the cases only 'ever incrementing counters' i.e. total counters will be cleared. Possible values = global, all

Operations (click to see Properties)

CLEAR

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nsstats?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsstats":{
    "cleanuplevel":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nssurgeq

May 20, 2016

Configuration for surge queue resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of a virtual server, service or service group for which the SurgeQ must be flushed.
servername	<String>	Read-write	Name of a service group member. This argument is needed when you want to flush the SurgeQ of a service group.
port	<Integer>	Read-write	port on which server is bound to the entity(Servicegroup). Range 1 - 65535

Operations (click to see Properties)

FLUSH

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

flush

URL: http://<netscaler-ip-address>/nitro/v1/config/nssurgeq?**action=flush**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nssurgeq":{
    "name":<String_value>,
    "servername":<String_value>,
    "port":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstcpbufparam

May 20, 2016

Configuration for tcp buffer parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
size	<Double>	Read-write	TCP buffering size per connection, in kilobytes. Default value: 64 Minimum value = 4 Maximum value = 20480
memlimit	<Double>	Read-write	Maximum memory, in megabytes, that can be used for buffering. Default value: 64

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpbufparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstcpbufparam":{
    "size":<Double_value>,
    "memlimit":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpbufparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstcpbufparam":{
    "size":true,
    "memlimit":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpbufparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstcpbufparam": [ {
      "size":<Double_value>,
      "memlimit":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstcpparam

May 20, 2016

Configuration for tcp parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ws	<String>	Read-write	Enable or disable window scaling. Default value: DISABLED Possible values = ENABLED, DISABLED
wsval	<Double>	Read-write	Factor used to calculate the new window size. This argument is needed only when the window scaling is enabled. Default value: 4 Minimum value = 0 Maximum value = 14
sack	<String>	Read-write	Enable or disable Selective ACKnowledgement (SACK). Default value: DISABLED Possible values = ENABLED, DISABLED
learnvsvrmss	<String>	Read-write	Enable or disable maximum segment size (MSS) learning for virtual servers. Default value: DISABLED Possible values = ENABLED, DISABLED
maxburst	<Double>	Read-write	Maximum number of TCP segments allowed in a burst. Default value: 6 Minimum value = 1 Maximum value = 255
initialcwnd	<Double>	Read-write	Initial maximum upper limit on the number of TCP packets that can be outstanding on the TCP link to the server. Default value: 4 Minimum value = 1 Maximum value = 44
recvbuffsize	<Double>	Read-write	TCP Receive buffer size. Default value: 8190 Minimum value = 8190 Maximum value = 20971520
delayedack	<Double>	Read-write	Timeout for TCP delayed ACK, in milliseconds. Default value: 100 Minimum value = 10

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
downstaterst	<String>	Read-write	Flag to switch on RST on down services. Default value: DISABLED Possible values = ENABLED, DISABLED
nagle	<String>	Read-write	Enable or disable the Nagle algorithm on TCP connections. Default value: DISABLED Possible values = ENABLED, DISABLED
limitedpersist	<String>	Read-write	Limit the number of persist (zero window) probes. Default value: ENABLED Possible values = ENABLED, DISABLED
oooqsize	<Double>	Read-write	Maximum size of out-of-order packets queue. A value of 0 means no limit. Default value: 64 Minimum value = 0 Maximum value = 65535
ackonpush	<String>	Read-write	Send immediate positive acknowledgement (ACK) on receipt of TCP packets with PUSH flag. Default value: ENABLED Possible values = ENABLED, DISABLED
maxpktpermss	<Integer>	Read-write	Maximum number of TCP packets allowed per maximum segment size (MSS). Minimum value = 0 Maximum value = 1460
pktperretx	<Integer>	Read-write	Maximum limit on the number of packets that should be retransmitted on receiving a partial ACK. Default value: 1 Minimum value = 1 Maximum value = 100
minrto	<Integer>	Read-write	Minimum retransmission timeout, in milliseconds, specified in 10-millisecond increments (value must yield a whole number if divided by 10). Default value: 1000 Minimum value = 10 Maximum value = 64000
slowstartincr	<Integer>	Read-write	Multiplier that determines the rate at which slow start increases the size of the TCP transmission window after each acknowledgement of successful transmission. Default value: 2 Minimum value = 1 Maximum value = 100
maxdynserverprobes	<Double>	Read-write	Maximum number of probes that NetScaler can send out in 10 milliseconds, to dynamically learn a service. NetScaler probes for the existence of the origin in case of wildcard virtual server or services. Default value: 7

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum value = 65535
synholdfastgiveup	<Double>	Read-write	Maximum threshold. After crossing this threshold number of outstanding probes for origin, the NetScaler reduces the number of connection retries for probe connections. Default value: 1024 Minimum value = 256 Maximum value = 65535
maxsynholdperprobe	<Double>	Read-write	Limit the number of client connections (SYN) waiting for status of single probe. Any new SYN packets will be dropped. Default value: 128 Minimum value = 1 Maximum value = 255
maxsynhold	<Double>	Read-write	Limit the number of client connections (SYN) waiting for status of probe system wide. Any new SYN packets will be dropped. Default value: 16384 Minimum value = 256 Maximum value = 65535
msslearninterval	<Double>	Read-write	Duration, in seconds, to sample the Maximum Segment Size (MSS) of the services. The NetScaler appliance determines the best MSS to set for the virtual server based on this sampling. The argument to enable maximum segment size (MSS) for virtual servers must be enabled. Default value: 180 Minimum value = 1 Maximum value = 1048576
msslearndelay	<Double>	Read-write	Frequency, in seconds, at which the virtual servers learn the Maximum segment size (MSS) from the services. The argument to enable maximum segment size (MSS) for virtual servers must be enabled. Default value: 3600 Minimum value = 1 Maximum value = 1048576
maxtimewaitconn	<Double>	Read-write	Maximum number of connections to hold in the TCP TIME_WAIT state on a packet engine. New connections entering TIME_WAIT state are proactively cleaned up. Default value: 7000 Minimum value = 1
kaprobeupdatelastactivity	<String>	Read-write	Update last activity for KA probes. Default value: ENABLED Possible values = ENABLED, DISABLED
maxsynackretx	<Double>	Read-write	When 'syncookie' is disabled in the TCP profile that is bound to the virtual server or service, and the number of TCP SYN+ACK retransmission by NetScaler for that virtual server or service crosses this threshold, the NetScaler appliance responds by using the TCP SYN-Cookie mechanism.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 100 Maximum value = 1048576
synattackdetection	<String>	Read-write	Detect TCP SYN packet flood and send an SNMP trap. Default value: ENABLED Possible values = ENABLED, DISABLED
connflushifnomem	<String>	Read-write	Flush an existing connection if no memory can be obtained for new connection. HALF_CLOSED_AND_IDLE: Flush a connection that is closed by us but not by peer, or failing that, a connection that is past configured idle time. New connection fails if no such connection can be found. FIFO: If no half-closed or idle connection can be found, flush the oldest non-management connection, even if it is active. New connection fails if the oldest few connections are management connections. Note: If you enable this setting, you should also consider lowering the zombie timeout and half-close timeout, while setting the NetScaler timeout. See Also: connFlushThres argument below. Default value: 5 Possible values = NONE, HALFCLOSED_AND_IDLE, FIFO
connflushthres	<Double>	Read-write	Flush an existing connection (as configured through -connFlushIfNoMem FIFO) if the system has more than specified number of connections, and a new connection is to be established. Note: This value may be rounded down to be a whole multiple of the number of packet engines running. Minimum value = 1
mptcpconcloseonpassivesf	<String>	Read-write	Accept DATA_FIN/FAST_CLOSE on passive subflow. Default value: ENABLED Possible values = ENABLED, DISABLED
mptcpchecksum	<String>	Read-write	Use MPTCP DSS checksum. Default value: ENABLED Possible values = ENABLED, DISABLED
mptcpsftimeout	<Double>	Read-write	The timeout value in seconds for idle mptcp subflows. If this timeout is not set, idle subflows are cleared after cltTimeout of vserver. Default value: 0 Minimum value = 0 Maximum value = 31536000
			The minimum idle time value in seconds for idle mptcp subflows after which the sublow is replaced by new incoming subflow if maximum subflow limit is reached. The priority for

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: 10 Minimum value = 0 Maximum value = 31536000
mptcpmaxsf	<Double>	Read-write	Maximum number of subflow connections supported in established state per mptcp connection. Default value: 4 Minimum value = 2 Maximum value = 6
mptcpmaxpendingsf	<Double>	Read-write	Maximum number of subflow connections supported in pending join state per mptcp connection. Default value: 4 Minimum value = 0 Maximum value = 4
mptcppendingjointhreshold	<Double>	Read-write	Maximum system level pending join connections allowed. Default value: 0 Minimum value = 0 Maximum value = 4294967294
mptcprtostoswitchsf	<Double>	Read-write	Number of RTO's at subflow level, after which MPCTP should start using other subflow. Default value: 2 Minimum value = 1 Maximum value = 6
mptcpusebackupondss	<String>	Read-write	When enabled, if NS receives a DSS on a backup subflow, NS will start using that subflow to send data. And if disabled, NS will continue to transmit on current chosen subflow. In case there is some error on a subflow (like RTO's/RST etc.) then NS can choose a backup subflow irrespective of this tunable. Default value: ENABLED Possible values = ENABLED, DISABLED
tcpmaxretries	<Double>	Read-write	Number of RTO's after which a connection should be freed. Default value: 7 Minimum value = 1 Maximum value = 7
mptcpimmediatesfcloseonfin	<String>	Read-write	Allow subflows to close immediately on FIN before the DATA_FIN exchange is completed at mptcp level. Default value: DISABLED Possible values = ENABLED, DISABLED
mptcpclosemptcpsessiononlastsfclose	<String>	Read-write	Allow to send DATA FIN or FAST CLOSE on mptcp connection while sending FIN or RST on the last subflow. Default value: DISABLED Possible values = ENABLED, DISABLED
builtin	<String[]>	Read-only	Flag to determine if the tcp param is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstcpparam":{
    "ws":<String_value>,
    "wsval":<Double_value>,
    "sack":<String_value>,
    "learnvsvrmss":<String_value>,
    "maxburst":<Double_value>,
    "initialcwnd":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "recvbuffsize":<Double_value>,
    "delayedack":<Double_value>,
    "downstaterst":<String_value>,
    "nagle":<String_value>,
    "limitedpersist":<String_value>,
    "oooqsize":<Double_value>,
    "ackonpush":<String_value>,
    "maxpktpermss":<Integer_value>,
    "pktperretx":<Integer_value>,
    "minrto":<Integer_value>,
    "slowstartincr":<Integer_value>,
    "maxdynserverprobes":<Double_value>,
    "synholdfastgiveup":<Double_value>,
    "maxsynholdperprobe":<Double_value>,
    "maxsynhold":<Double_value>,
    "msslearninterval":<Double_value>,
    "msslearndelay":<Double_value>,
    "maxtimewaitconn":<Double_value>,
    "kaprobeupdatelastactivity":<String_value>,
    "maxsynackretx":<Double_value>,
    "synattackdetection":<String_value>,
    "connflushifnomem":<String_value>,
    "connflushthres":<Double_value>,
    "mptcpconcloseonpassivesf":<String_value>,
    "mptcpchecksum":<String_value>,
    "mptcpsftimeout":<Double_value>,
    "mptcpsfreplacetimeout":<Double_value>,
    "mptcpmaxsf":<Double_value>,
    "mptcpmaxpendingsf":<Double_value>,
    "mptcppendingjointhreshold":<Double_value>,
    "mptcprtostoswitchsf":<Double_value>,
    "mptcpusebackupondss":<String_value>,
    "tcpmaxretries":<Double_value>,
    "mptcpimmediatesfcloseonfin":<String_value>,
    "mptcpclosemptcpsessiononlastsfclose":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"nstcpparam":{
    "ws":true,
    "wsval":true,
    "sack":true,
    "learnvsvrmss":true,
    "maxburst":true,
    "initialcwnd":true,
    "recvbuffsize":true,
    "delayedack":true,
    "downstaterst":true,
    "nagle":true,
    "limitedpersist":true,
    "oooqsize":true,
    "ackonpush":true,
    "maxpktpermss":true,
    "pktperretx":true,
    "minrto":true,
    "slowstartincr":true,
    "maxdynserverprobes":true,
    "synholdfastgiveup":true,
    "maxsynholdperprobe":true,
    "maxsynhold":true,
    "msslearninterval":true,
    "msslearndelay":true,
    "maxtimewaitconn":true,
    "kaprobeupdatelastactivity":true,
    "maxsynackretx":true,
    "synattackdetection":true,
    "connflushifnomem":true,
    "connflushthres":true,
    "mptcpconcloseonpassivesf":true,
    "mptcpchecksum":true,
    "mptcpsftimeout":true,
    "mptcpsfreplacetimeout":true,
    "mptcpmaxsf":true,
    "mptcpmaxpendingsf":true,
    "mptcppendingjointhreshold":true,
    "mptcprtostoswitchsf":true,
    "mptcpusebackupondss":true,
    "tcpmaxretries":true,
    "mptcpimmediatesfcloseonfin":true,
    "mptcpclosemptcpsessiononlastsfclose":true
}}
```


Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpparam

HTTP Method: GET

Request Headers:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "nstcpparam": [ {
    "ws":<String_value>,
    "wsval":<Double_value>,
    "sack":<String_value>,
    "learnvsvrmss":<String_value>,
    "maxburst":<Double_value>,
    "initialcwnd":<Double_value>,
    "recvbuffsize":<Double_value>,
    "delayedack":<Double_value>,
    "downstaterst":<String_value>,
    "nagle":<String_value>,
    "limitedpersist":<String_value>,
    "oooqsize":<Double_value>,
    "ackonpush":<String_value>,
    "maxpktpermss":<Integer_value>,
    "pktperretx":<Integer_value>,
    "minrto":<Integer_value>,
    "slowstartincr":<Integer_value>,
    "maxdynserverprobes":<Double_value>,
    "synholdfastgiveup":<Double_value>,
    "maxsynholdperprobe":<Double_value>,
    "maxsynhold":<Double_value>,
    "msslearninterval":<Double_value>,
    "msslearndelay":<Double_value>,
    "maxtimewaitconn":<Double_value>,
    "kaprobeupdatelastactivity":<String_value>,
    "maxsynackretx":<Double_value>,
    "synattackdetection":<String_value>,
    "connflushifnomem":<String_value>,
    "connflushthres":<Double_value>,
    "mptcpconcloseonpassivesf":<String_value>,
    "mptcpchecksum":<String_value>,
    "mptcpsftimeout":<Double_value>,
    "mptcpsfreplacetimeout":<Double_value>,
    "mptcpmaxsf":<Double_value>,
    "mptcpmaxpendingsf":<Double_value>,
    "mptcppendingjointhreshold":<Double_value>,
    "mptcprtostoswitchsf":<Double_value>,
    "mptcpusebackupondss":<String_value>,
    "tcpmaxretries":<Double_value>,
    "mptcpimmediatesfcloseonfin":<String_value>,
    "mptcpclosemptcpsessiononlastsfclose":<String_value>,
    "builtin":<String[]_value>
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

}]}

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstcpprofile

May 20, 2016

Configuration for TCP profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for a TCP profile. Must begin with a letter, number, or the underscore \(_\) character. Other characters allowed, after the first character, are the hyphen \(-\), period \(.\), hash \(\#\), space \(\), at \(@\), colon \(:\), and equal \(=\) characters. The name of a TCP profile cannot be changed after it is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks \(for example, "my tcp profile" or 'my tcp profile'\). Minimum length = 1 Maximum length = 127
ws	<String>	Read-write	Enable or disable window scaling. Default value: DISABLED Possible values = ENABLED, DISABLED
sack	<String>	Read-write	Enable or disable Selective ACKnowledgement (SACK). Default value: DISABLED Possible values = ENABLED, DISABLED
wsval	<Double>	Read-write	Factor used to calculate the new window size. This argument is needed only when window scaling is enabled. Default value: 4 Minimum value = 0 Maximum value = 14
nagle	<String>	Read-write	Enable or disable the Nagle algorithm on TCP connections. Default value: DISABLED Possible values = ENABLED, DISABLED
ackonpush	<String>	Read-write	Send immediate positive acknowledgement (ACK) on receipt of TCP packets with PUSH flag. Default value: ENABLED Possible values = ENABLED, DISABLED
mss	<Double>	Read-write	Maximum number of octets to allow in a TCP data segment. Minimum value = 0 Maximum value = 9176

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
maxburst	<Double>	Read-write	Default value: 6 Minimum value = 1 Maximum value = 255
initialcwnd	<Double>	Read-write	Initial maximum upper limit on the number of TCP packets that can be outstanding on the TCP link to the server. Default value: 4 Minimum value = 1 Maximum value = 44
delayedack	<Double>	Read-write	Timeout for TCP delayed ACK, in milliseconds. Default value: 100 Minimum value = 10 Maximum value = 300
oooqsize	<Double>	Read-write	Maximum size of out-of-order packets queue. A value of 0 means no limit. Default value: 64 Minimum value = 0 Maximum value = 65535
maxpktpermss	<Double>	Read-write	Maximum number of TCP packets allowed per maximum segment size (MSS). Minimum value = 0 Maximum value = 1460
pktperretx	<Double>	Read-write	Maximum limit on the number of packets that should be retransmitted on receiving a partial ACK. Default value: 1 Minimum value = 1 Maximum value = 512
minrto	<Double>	Read-write	Minimum retransmission timeout, in milliseconds, specified in 10-millisecond increments (value must yield a whole number if divided by 10). Default value: 1000 Minimum value = 10 Maximum value = 64000
slowstartincr	<Double>	Read-write	Multiplier that determines the rate at which slow start increases the size of the TCP transmission window after each acknowledgement of successful transmission. Default value: 2 Minimum value = 1 Maximum value = 100
buffersize	<Double>	Read-write	TCP buffering size, in bytes. Default value: 8190 Minimum value = 8190 Maximum value = 20971520
syncookie	<String>	Read-write	Enable or disable the SYNCOOKIE mechanism for TCP handshake with clients. Disabling SYNCOOKIE prevents SYN attack protection on the NetScaler appliance. Default value: ENABLED Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
kaprobeupdatelastactivity	<String>	Read-write	Update last activity for the connection after receiving keep-alive (KA) probes. Default value: ENABLED Possible values = ENABLED, DISABLED
flavor	<String>	Read-write	Set TCP congestion control algorithm. Default value: Default Possible values = Default, Westwood, BIC, CUBIC, Nile
dynamicreceivebuffering	<String>	Read-write	Enable or disable dynamic receive buffering. When enabled, allows the receive buffer to be adjusted dynamically based on memory and network conditions. Note: The buffer size argument must be set for dynamic adjustments to take place. Default value: ENABLED Possible values = ENABLED, DISABLED
ka	<String>	Read-write	Send periodic TCP keep-alive (KA) probes to check if peer is still up. Default value: DISABLED Possible values = ENABLED, DISABLED
kaconnidletime	<Double>	Read-write	Duration, in seconds, for the connection to be idle, before sending a keep-alive (KA) probe. Minimum value = 1 Maximum value = 4095
kamaxprobes	<Double>	Read-write	Number of keep-alive (KA) probes to be sent when not acknowledged, before assuming the peer to be down. Minimum value = 1 Maximum value = 255
kaprobeinterval	<Double>	Read-write	Time interval, in seconds, before the next keep-alive (KA) probe, if the peer does not respond. Minimum value = 1 Maximum value = 4095
sendbuffsize	<Double>	Read-write	TCP Send Buffer Size. Default value: 8190 Minimum value = 8190 Maximum value = 20971520
mptcp	<String>	Read-write	Enable or disable Multipath TCP. Default value: DISABLED Possible values = ENABLED, DISABLED
establishclientconn	<String>	Read-write	Establishing Client Client connection on First data/ Final-ACK / Automatic. Default value: AUTOMATIC Possible values = AUTOMATIC, CONN_ESTABLISHED, ON_FIRST_DATA
tcpsegoffload	<String>	Read-write	Offload TCP segmentation to the NIC. If set to AUTOMATIC, TCP segmentation will be offloaded to the NIC, if the NIC supports it. Default value: AUTOMATIC Possible values = AUTOMATIC, DISABLED
			Enable or disable RST window attenuation to protect against spoofing.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
rstwindowattenuate	<String>	Read-write	invalid. Default value: DISABLED Possible values = ENABLED, DISABLED
rstmaxack	<String>	Read-write	Enable or disable acceptance of RST that is out of window yet echoes highest ACK sequence number. Useful only in proxy mode. Default value: DISABLED Possible values = ENABLED, DISABLED
spoofsyndrop	<String>	Read-write	Enable or disable drop of invalid SYN packets to protect against spoofing. When disabled, established connections will be reset when a SYN packet is received. Default value: ENABLED Possible values = ENABLED, DISABLED
ecn	<String>	Read-write	Enable or disable TCP Explicit Congestion Notification. Default value: DISABLED Possible values = ENABLED, DISABLED
mptcpdropdataonpreestsf	<String>	Read-write	Enable or disable silently dropping the data on Pre-Established subflow. When enabled, DSS data packets are dropped silently instead of dropping the connection when data is received on pre established subflow. Default value: DISABLED Possible values = ENABLED, DISABLED
mptcpfastopen	<String>	Read-write	Enable or disable Multipath TCP fastopen. When enabled, DSS data packets are accepted before receiving the third ack of SYN handshake. Default value: DISABLED Possible values = ENABLED, DISABLED
mptcpsessiontimeout	<Double>	Read-write	MPTCP session timeout in seconds. If this value is not set, idle MPTCP sessions are flushed after vserver's client idle timeout. Default value: 0 Minimum value = 0 Maximum value = 86400
timestamp	<String>	Read-write	Enable or Disable TCP Timestamp option (RFC 1323). Default value: DISABLED Possible values = ENABLED, DISABLED
dsack	<String>	Read-write	Enable or disable DSACK. Default value: ENABLED Possible values = ENABLED, DISABLED
ackaggregation	<String>	Read-write	Enable or disable ACK Aggregation. Default value: DISABLED Possible values = ENABLED, DISABLED
frto	<String>	Read-write	Enable or disable FRTO (Forward RTO-Recovery). Default value: DISABLED Possible values = ENABLED, DISABLED
			TCP Maximum Congestion Window.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 8190 Maximum value = 20971520
fack	<String>	Read-write	Enable or disable FACK (Forward ACK). Default value: DISABLED Possible values = ENABLED, DISABLED
tcpmode	<String>	Read-write	TCP Optimization modes TRANSPARENT / ENDPOINT. Default value: TRANSPARENT Possible values = TRANSPARENT, ENDPOINT
refcnt	<Double>	Read-only	Number of entities using this profile.
builtin	<String[]>	Read-only	Flag to determine if tcp profile is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstcpprofile":{
    "name":<String_value>,
    "ws":<String_value>,
    "sack":<String_value>,
    "wsval":<Double_value>,
    "nagle":<String_value>,
    "ackonpush":<String_value>,
    "mss":<Double_value>,
    "maxburst":<Double_value>,
    "initialcwnd":<Double_value>,
    "delayedack":<Double_value>,
    "oooqsize":<Double_value>,
    "maxpktpermss":<Double_value>,
    "pktperretx":<Double_value>,
    "minrto":<Double_value>,
    "slowstartincr":<Double_value>,
    "buffersize":<Double_value>,
    "syncookie":<String_value>,
    "kaprobeupdatelastactivity":<String_value>,
    "flavor":<String_value>,
    "dynamicreceivebuffering":<String_value>,
    "ka":<String_value>,
    "kaconnidletime":<Double_value>,
    "kamaxprobes":<Double_value>,
    "kaprobeinterval":<Double_value>,
    "sendbuffsize":<Double_value>,
    "mptcp":<String_value>,
    "establishclientconn":<String_value>,
    "tcpsegoffload":<String_value>,
    "rstwindowattenuate":<String_value>,
    "rstmaxack":<String_value>,
    "spoofsyndrop":<String_value>,
    "ecn":<String_value>,
    "mptcpdropdataonpreestsf":<String_value>,
    "mptcpfastopen":<String_value>,
    "mptcpsessiontimeout":<Double_value>,
    "timestamp":<String_value>,
    "dsack":<String_value>,
    "ackaggregation":<String_value>,
    "frto":<String_value>,
    "maxcwnd":<Double_value>,
    "fack":<String_value>,
    "tcpmode":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstcpprofile":{
    "name":<String_value>,
    "ws":<String_value>,
    "sack":<String_value>,
    "wsval":<Double_value>,
    "nagle":<String_value>,
    "ackonpush":<String_value>,
    "mss":<Double_value>,
    "maxburst":<Double_value>,
    "initialcwnd":<Double_value>,
    "delayedack":<Double_value>,
    "oooqsize":<Double_value>,
    "maxpktpermss":<Double_value>,
    "pktperretx":<Double_value>,
    "minrto":<Double_value>,
    "slowstartincr":<Double_value>,
    "buffersize":<Double_value>,
    "syncookie":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "kaprobeupdatelastactivity":<String_value>,
    "flavor":<String_value>,
    "dynamicreceivebuffering":<String_value>,
    "ka":<String_value>,
    "kaconnidletime":<Double_value>,
    "kamaxprobes":<Double_value>,
    "kaprobeinterval":<Double_value>,
    "sendbuffsize":<Double_value>,
    "mptcp":<String_value>,
    "establishclientconn":<String_value>,
    "tcpsegoffload":<String_value>,
    "rstwindowattenuate":<String_value>,
    "rstmaxack":<String_value>,
    "spoofsyndrop":<String_value>,
    "ecn":<String_value>,
    "mptcpdropdataonpreestsf":<String_value>,
    "mptcpfastopen":<String_value>,
    "mptcpsessiontimeout":<Double_value>,
    "timestamp":<String_value>,
    "dsack":<String_value>,
    "ackaggregation":<String_value>,
    "frto":<String_value>,
    "maxcwnd":<Double_value>,
    "fack":<String_value>,
    "tcpmode":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstcpprofile":{
    "name":<String_value>,
    "ws":true,
    "sack":true,
    "wsval":true,
    "nagle":true,
    "ackonpush":true,
    "mss":true,
    "maxburst":true,
    "initialcwnd":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "delayedack":true,
        "oooqsize":true,
        "maxpktpermss":true,
        "pktperretx":true,
        "minrto":true,
        "slowstartincr":true,
        "buffersize":true,
        "syncookie":true,
        "kaprobeupdatelastactivity":true,
        "flavor":true,
        "dynamicreceivebuffering":true,
        "ka":true,
        "kamaxprobes":true,
        "kaconnidletime":true,
        "kaprobeinterval":true,
        "sendbuffsize":true,
        "mptcp":true,
        "establishclientconn":true,
        "tcpsegoffload":true,
        "rstwindowattenuate":true,
        "rstmaxack":true,
        "spoofsyndrop":true,
        "ecn":true,
        "mptcpdropdataonpreestsf":true,
        "mptcpfastopen":true,
        "mptcpsessiontimeout":true,
        "timestamp":true,
        "dsack":true,
        "ackaggregation":true,
        "frto":true,
        "maxcwnd":true,
        "fack":true,
        "tcpmode":true
    }}
```


Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile?**filter=property-name1:property-val1,property-**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

name2:property-val2

Use this query-parameter to get the filtered set of nstcpprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nstcpprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstcpprofile": [ {
    "name":<String_value>,
    "ws":<String_value>,
    "sack":<String_value>,
    "wsval":<Double_value>,
    "nagle":<String_value>,
    "ackonpush":<String_value>,
    "mss":<Double_value>,
    "maxburst":<Double_value>,
    "initialcwnd":<Double_value>,
    "delayedack":<Double_value>,
    "oooqsize":<Double_value>,
    "maxpktpermss":<Double_value>,
    "pktperretx":<Double_value>,
    "minrto":<Double_value>,
    "slowstartincr":<Double_value>,
    "buffersize":<Double_value>,
    "flavor":<String_value>,
    "refcnt":<Double_value>,
    "syncookie":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "kaprobeupdatelastactivity":<String_value>,
    "dynamicreceivebuffering":<String_value>,
    "ka":<String_value>,
    "kaconnidletime":<Double_value>,
    "kamaxprobes":<Double_value>,
    "kaprobeinterval":<Double_value>,
    "sendbuffsize":<Double_value>,
    "mptcp":<String_value>,
    "establishclientconn":<String_value>,
    "tcpsegoffload":<String_value>,
    "rstwindowattenuate":<String_value>,
    "rstmaxack":<String_value>,
    "timestamp":<String_value>,
    "spoofsyndrop":<String_value>,
    "ecn":<String_value>,
    "mptcpdropdataonpreestsf":<String_value>,
    "mptcpfastopen":<String_value>,
    "mptcpsessiontimeout":<Double_value>,
    "dsack":<String_value>,
    "ackaggregation":<String_value>,
    "frto":<String_value>,
    "maxcwnd":<Double_value>,
    "fack":<String_value>,
    "tcpmode":<String_value>,
    "builtin":<String[]_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstcpprofile": [ {
    "name":<String_value>,
    "ws":<String_value>,
    "sack":<String_value>,
    "wsval":<Double_value>,
    "nagle":<String_value>,
    "ackonpush":<String_value>,
    "mss":<Double_value>,
    "maxburst":<Double_value>,
    "initialcwnd":<Double_value>,
    "delayedack":<Double_value>,
    "oooqsize":<Double_value>,
    "maxpktpermss":<Double_value>,
    "pktperretx":<Double_value>,
    "minrto":<Double_value>,
    "slowstartincr":<Double_value>,
    "buffersize":<Double_value>,
    "flavor":<String_value>,
    "refcnt":<Double_value>,
    "syncookie":<String_value>,
    "kaprobeupdatelastactivity":<String_value>,
    "dynamicreceivebuffering":<String_value>,
    "ka":<String_value>,
    "kaconnidletime":<Double_value>,
    "kamaxprobes":<Double_value>,
    "kaprobeinterval":<Double_value>,
    "sendbuffsize":<Double_value>,
    "mptcp":<String_value>,
    "establishclientconn":<String_value>,
    "tcpsegoffload":<String_value>,
    "rstwindowattenuate":<String_value>,
    "rstmaxack":<String_value>,
    "timestamp":<String_value>,
    "spoofsyndrop":<String_value>,
    "ecn":<String_value>,
    "mptcpdropdataonpreestsf":<String_value>,
    "mptcpfastopen":<String_value>,
    "mptcpsessiontimeout":<Double_value>,
    "dsack":<String_value>,
    "ackaggregation":<String_value>,
    "frto":<String_value>,
    "maxcwnd":<Double_value>,
    "fack":<String_value>,
    "tcpmode":<String_value>,
    "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nstcpprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstcpprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstimeout

May 20, 2016

Configuration for timeout resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
zombie	<Double>	Read-write	Interval, in seconds, at which the NetScaler zombie cleanup process must run. This process cleans up inactive TCP connections. Default value: 120 Minimum value = 1 Maximum value = 600
client	<Double>	Read-write	Client idle timeout (in seconds). If zero, the service-type default value is taken when service is created. Default value: 0 Minimum value = 0 Maximum value = 18000
server	<Double>	Read-write	Server idle timeout (in seconds). If zero, the service-type default value is taken when service is created. Default value: 0 Minimum value = 0 Maximum value = 18000
httpclient	<Double>	Read-write	Global idle timeout, in seconds, for client connections of HTTP service type. This value is over ridden by the client timeout that is configured on individual entities. Default value: 0 Minimum value = 0 Maximum value = 18000
httpserver	<Double>	Read-write	Global idle timeout, in seconds, for server connections of HTTP service type. This value is over ridden by the server timeout that is configured on individual entities. Default value: 0 Minimum value = 0 Maximum value = 18000
tcpclient	<Double>	Read-write	Global idle timeout, in seconds, for non-HTTP client connections of TCP service type. This value is over ridden by the client timeout that is configured on individual entities. Default value: 0 Minimum value = 0 Maximum value = 18000
			Global idle timeout, in seconds, for non-HTTP server connections of TCP service type. This value is over ridden by the server timeout that is configured on entities.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 0 Maximum value = 18000
anyclient	<Double>	Read-write	Global idle timeout, in seconds, for non-TCP client connections. This value is over ridden by the client timeout that is configured on individual entities. Default value: 0 Minimum value = 0 Maximum value = 31536000
anyserver	<Double>	Read-write	Global idle timeout, in seconds, for non TCP server connections. This value is over ridden by the server timeout that is configured on individual entities. Default value: 0 Minimum value = 0 Maximum value = 31536000
anytcpclient	<Double>	Read-write	Global idle timeout, in seconds, for TCP client connections. This value takes precedence over entity level timeout settings (vserver/service). This is applicable only to transport protocol TCP. Default value: 0 Minimum value = 0 Maximum value = 31536000
anytcpserver	<Double>	Read-write	Global idle timeout, in seconds, for TCP server connections. This value takes precedence over entity level timeout settings (vserver/service). This is applicable only to transport protocol TCP. Default value: 0 Minimum value = 0 Maximum value = 31536000
halfclose	<Double>	Read-write	Idle timeout, in seconds, for connections that are in TCP half-closed state. Default value: 10 Minimum value = 1 Maximum value = 600
nontcpzombie	<Double>	Read-write	Interval at which the zombie clean-up process for non-TCP connections should run. Inactive IP NAT connections will be cleaned up. Default value: 60 Minimum value = 1 Maximum value = 600
reducedfintimeout	<Double>	Read-write	Alternative idle timeout, in seconds, for closed TCP NATPCB connections. Default value: 30 Minimum value = 1 Maximum value = 300
reducedrsttimeout	<Double>	Read-write	Timer interval, in seconds, for abruptly terminated TCP NATPCB connections. Default value: 0 Minimum value = 0 Maximum value = 300
newconnidletimeout	<Double>	Read-write	Timer interval, in seconds, for new TCP NATPCB connections on which no data was received. Default value: 4 Minimum value = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimeout

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstimeout":{
    "zombie":<Double_value>,
    "client":<Double_value>,
    "server":<Double_value>,
    "httpclient":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "httpserver":<Double_value>,
    "tcpclient":<Double_value>,
    "tcpserver":<Double_value>,
    "anyclient":<Double_value>,
    "anyserver":<Double_value>,
    "anytcpclient":<Double_value>,
    "anytcpserver":<Double_value>,
    "halfclose":<Double_value>,
    "nontcpzombie":<Double_value>,
    "reducedfintimeout":<Double_value>,
    "reducedrsttimeout":<Double_value>,
    "newconnidletimeout":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimeout?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstimeout":{
    "zombie":true,
    "client":true,
    "server":true,
    "httpclient":true,
    "httpserver":true,
    "tcpclient":true,
    "tcpserver":true,
    "anyclient":true,
    "anyserver":true,
    "anytcpclient":true,
    "anytcpserver":true,
    "halfclose":true,
    "nontcpzombie":true,
    "reducedfintimeout":true,
    "reducedrsttimeout":true,
    "newconnidletimeout":true
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimeout

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstimeout": [ {
      "zombie":<Double_value>,
      "client":<Double_value>,
      "server":<Double_value>,
      "httpclient":<Double_value>,
      "httpserver":<Double_value>,
      "tcpclient":<Double_value>,
      "tcpserver":<Double_value>,
      "anyclient":<Double_value>,
      "anyserver":<Double_value>,
      "halfclose":<Double_value>,
      "anytcpclient":<Double_value>,
      "anytcpserver":<Double_value>,
      "nontcpzombie":<Double_value>,
      "reducedfintimeout":<Double_value>,
      "reducedrsttimeout":<Double_value>,
      "newconnidletimeout":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstimer

May 20, 2016

Configuration for Timer resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Timer name. Minimum length = 1
interval	<Integer>	Read-write	The frequency at which the policies bound to this timer are invoked. The minimum value is 20 msec. The maximum value is 20940 in seconds and 349 in minutes. Default value: 5 Minimum value = 1 Maximum value = 20940000
unit	<String>	Read-write	Timer interval unit. Default value: SEC Possible values = SEC, MIN
comment	<String>	Read-write	Comments associated with this timer.
newname	<String>	Read-write	The new name of the timer. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstimer":{
    "name":<String_value>,
    "interval":<Integer_value>,
    "unit":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer/name_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstimer":{
    "name":<String_value>,
    "interval":<Integer_value>,
    "unit":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstimer":{
    "name":<String_value>,
    "interval":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "unit":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstimer":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nstimer?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nstimer?**filter=property-name1:property-val1,property-name2:property-val2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of nstimer resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nstimer?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nstimer?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nstimer resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstimer": [ {
      "name":<String_value>,
      "interval":<Integer_value>,
      "unit":<String_value>,
      "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nstimer/name_value<String>?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nstimer/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstimer": [ {
      "name":<String_value>,
      "interval":<Integer_value>,
      "unit":<String_value>,
      "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "nstimer": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstimer_autoscalepolicy_binding

May 20, 2016

Binding object showing the autoscalepolicy that can be bound to nstimer.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the timer policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	The timer policy associated with the timer.
name	<String>	Read-write	Timer name. Minimum length = 1
threshold	<Double>	Read-write	Denotes the threshold. If the rule of the policy in the binding relation evaluates 'threshold size' number of times in 'sample size' to true, then the corresponding action is taken. Its value needs to be less than or equal to the sample size value. Default value: 3 Minimum value = 1 Maximum value = 32
samplesize	<Double>	Read-write	Denotes the sample size. Sample size value of 'x' means that previous '(x - 1)' policy's rule evaluation results and the current evaluation result are present with the binding. For example, sample size of 10 means that there is a state of previous 9 policy evaluation results and also the current policy evaluation result. Default value: 3 Minimum value = 1 Maximum value = 32
vserver	<String>	Read-write	Name of the vserver which provides the context for the rule in timer policy. When not specified it is treated as a Global Default context.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/nstimer_autoscalepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"nstimer_autoscalepolicy_binding":{
    "name":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "vserver":<String_value>,
    "samplesize":<Double_value>,
    "threshold":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nstimer_autoscalepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nstimer_autoscalepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstimer_autoscalepolicy_binding": [ {
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "policyname":<String_value>,
    "name":<String_value>,
    "threshold":<Double_value>,
    "samplesize":<Double_value>,
    "vserver":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer_autoscalepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"nstimer_autoscalepolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstimer_binding

May 20, 2016

Binding object which returns the resources bound to nstimer.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Timer name. Minimum length = 1
nstimer_autoscalepolicy_binding	<nstimer_autoscalepolicy_binding[]>	Read-only	autoscalepolicy that can be bound to nstimer.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nstimer_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "nstimer_binding": [ {
      "name":<String_value>,
      "nstimer_autoscalepolicy_binding":<nstimer_autoscalepolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstrafficdomain

May 20, 2016

Configuration for Traffic Domain resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
td	<Double>	Read-write	Integer value that uniquely identifies a traffic domain. Minimum value = 1 Maximum value = 4094
aliasname	<String>	Read-write	Name of traffic domain being added. Minimum length = 1 Maximum length = 31
vmac	<String>	Read-write	Associate the traffic domain with a VMAC address instead of with VLANs. The NetScaler ADC then sends the VMAC address of the traffic domain in all responses to ARP queries for network entities in that domain. As a result, the ADC can segregate subsequent incoming traffic for this traffic domain on the basis of the destination MAC address, because the destination MAC address is the VMAC address of the traffic domain. After creating entities on a traffic domain, you can easily manage and monitor them by performing traffic domain level operations. Default value: DISABLED Possible values = ENABLED, DISABLED
state	<String>	Read-only	The state of TrafficDmain. Default value: ENABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | CLEAR | ENABLE | DISABLE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstrafficdomain":{
    "td":<Double_value>,
    "aliasname":<String_value>,
    "vmac":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain/td_value<Double>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstrafficdomain":{
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstrafficdomain":{
    "td":<Double_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nstrafficdomain":{
    "td":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nstrafficdomain resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nstrafficdomain resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstrafficdomain": [ {
      "td":<Double_value>,
      "aliasname":<String_value>,
      "vmac":<String_value>,
      "state":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain/td_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain/td_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain/td_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstrafficdomain": [ {
    "td":<Double_value>,
    "aliasname":<String_value>,
    "vmac":<String_value>,
    "state":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{ "nstrafficdomain": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstrafficdomain_binding

May 20, 2016

Binding object which returns the resources bound to nstrafficdomain.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
td	<Double>	Read-write	Integer value that uniquely identifies a traffic domain. Minimum value = 1 Maximum value = 4094
nstrafficdomain_bridgegroup_binding	<nstrafficdomain_bridgegroup_binding[]>	Read-only	bridgegroup that can be bound to nstrafficdomain.
nstrafficdomain_vlan_binding	<nstrafficdomain_vlan_binding[]>	Read-only	vlan that can be bound to nstrafficdomain.
nstrafficdomain_vxlan_binding	<nstrafficdomain_vxlan_binding[]>	Read-only	vxlan that can be bound to nstrafficdomain.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_binding/td_value<Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "nstrafficdomain_binding": [ {
      "td":<Double_value>,
      "nstrafficdomain_vlan_binding":<nstrafficdomain_vlan_binding[]_value>,
      "nstrafficdomain_bridgegroup_binding":<nstrafficdomain_bridgegroup_binding[]_value>,
      "nstrafficdomain_vxlan_binding":<nstrafficdomain_vxlan_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstrafficdomain_bridgegroup_binding

May 20, 2016

Binding object showing the bridgegroup that can be bound to nstrafficdomain.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
bridgegroup	<Double>	Read-write	ID of the configured bridge to bind to this traffic domain. More than one bridge group can be bound to a traffic domain, but the same bridge group cannot be a part of multiple traffic domains. Minimum value = 1 Maximum value = 1000
td	<Double>	Read-write	Integer value that uniquely identifies a traffic domain. Minimum value = 1 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/nstrafficdomain_bridgegroup_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"nstrafficdomain_bridgegroup_binding":{
    "td":<Double_value>,
    "bridgegroup":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nstrafficdomain_bridgegroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nstrafficdomain_bridgegroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstrafficdomain_bridgegroup_binding": [ {
    "bridgegroup":<Double_value>,
    "td":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_bridgegroup_binding/td_value<Double>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"nstrafficdomain_bridgegroup_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsvariable

May 20, 2016

Configuration for variable resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Variable name. This follows the same syntax rules as other default syntax expression entity names: It must begin with an alpha character (A-Z or a-z) or an underscore (_). The rest of the characters must be alpha, numeric (0-9) or underscores. It cannot be re or xp (reserved for regular and XPath expressions). It cannot be a default syntax expression reserved word (e.g. SYS or HTTP). It cannot be used for an existing default syntax expression object (HTTP callout, patset, dataset, stringmap, or named expression). Minimum length = 1
type	<String>	Read-write	Specification of the variable type; one of the following: ulong - singleton variable with an unsigned 64-bit value. text(value-max-size) - singleton variable with a text string value. map(text(key-max-size),ulong,max-entries) - map of text string keys to unsigned 64-bit values. map(text(key-max-size),text(value-max-size),max-entries) - map of text string keys to text string values. where value-max-size is a positive integer that is the maximum number of bytes in a text string value. key-max-size is a positive integer that is the maximum number of bytes in a text string key. max-entries is a positive integer that is the maximum number of entries in a map variable. For a global singleton text variable, value-max-size <= 64000. For a global map with ulong values, key-max-size <= 64000. For a global map with text values, key-max-size + value-max-size <= 64000. max-entries is a positive integer that is the maximum number of entries in a map variable. This has a theoretical maximum of 2^64-1, but in actual use will be much smaller, considering the memory available for use by the map. Example: map(text(10),text(20),100) specifies a map of text string keys (max size 10 bytes) to text string values (max size 20 bytes), with 100 max entries. Minimum length = 1
scope	<String>	Read-write	Scope of the variable: global - (default) one set of values visible across all Packet Engines and, in a cluster, all nodes transaction - one value for each request-response transaction (singleton variables

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: global Possible values = global, transaction
iffull	<String>	Read-write	Action to perform if an assignment to a map exceeds its configured max-entries: lru - (default) reuse the least recently used entry in the map. undef - force the assignment to return an undefined (Undef) result to the policy executing the assignment. Default value: lru Possible values = undef, lru
ifvaluetoobig	<String>	Read-write	Action to perform if an value is assigned to a text variable that exceeds its configured max-size, or if a key is used that exceeds its configured max-size: truncate - (default) truncate the text string to the first max-size bytes and proceed. undef - force the assignment or expression evaluation to return an undefined (Undef) result to the policy executing the assignment or expression. Default value: truncate Possible values = undef, truncate
ifnovalue	<String>	Read-write	Action to perform if on a variable reference in an expression if the variable is single-valued and uninitialized or if the variable is a map and there is no value for the specified key: init - (default) initialize the single-value variable, or create a map entry for the key and the initial value, using the -init value or its default. undef - force the expression evaluation to return an undefined (Undef) result to the policy executing the expression. Default value: init Possible values = undef, init
init	<String>	Read-write	Initialization value for values in this variable. Default: 0 for ulong, NULL for text.
expires	<Double>	Read-write	Value expiration in seconds. If the value is not referenced within the expiration period it will be deleted. 0 (the default) means no expiration. Minimum value = 0 Maximum value = 31622400
comment	<String>	Read-write	Comments associated with this variable.
referencecount	<Double>	Read-only	The number of references to the variable in expressions and assignments.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsvariable

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsvariable":{
    "name":<String_value>,
    "type":<String_value>,
    "scope":<String_value>,
    "iffull":<String_value>,
    "ifvaluetoobig":<String_value>,
    "ifnovalue":<String_value>,
    "init":<String_value>,
    "expires":<Double_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsvariable

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsvariable":{
    "name":<String_value>,
    "type":<String_value>,
    "iffull":<String_value>,
    "ifvaluetoobig":<String_value>,
    "ifnovalue":<String_value>,
    "init":<String_value>,
    "expires":<Double_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsvariable?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsvariable":{
    "name":<String_value>,
    "iffull":true,
    "ifvaluetoobig":true,
    "ifnovalue":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "init":true,
        "expires":true,
        "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsvariable/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsvariable

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsvariable?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsvariable?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsvariable resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsvariable?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsvariable?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsvariable resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsvariable": [ {
      "name":<String_value>,
      "type":<String_value>,
      "scope":<String_value>,
      "iffull":<String_value>,
      "ifvaluetoobig":<String_value>,
      "ifnovalue":<String_value>,
      "expires":<Double_value>,
      "init":<String_value>,
      "comment":<String_value>,
      "referencecount":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsvariable/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsvariable/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/nsvariable/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsvariable": [ {
    "name":<String_value>,
    "type":<String_value>,
    "scope":<String_value>,
    "iffull":<String_value>,
    "ifvaluetoobig":<String_value>,
    "ifnovalue":<String_value>,
    "expires":<Double_value>,
    "init":<String_value>,
    "comment":<String_value>,
    "referencecount":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsvariable?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "nsvariable": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstrafficdomain_vlan_binding

May 20, 2016

Binding object showing the vlan that can be bound to nstrafficdomain.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vlan	<Double>	Read-write	ID of the VLAN to bind to this traffic domain. More than one VLAN can be bound to a traffic domain, but the same VLAN cannot be a part of multiple traffic domains. Minimum value = 1 Maximum value = 4094
td	<Double>	Read-write	Integer value that uniquely identifies a traffic domain. Minimum value = 1 Maximum value = 4094
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/nstrafficdomain_vlan_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"nstrafficdomain_vlan_binding":{
    "td":<Double_value>,
    "vlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nstrafficdomain_vlan_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nstrafficdomain_vlan_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstrafficdomain_vlan_binding": [ {
      "vlan":<Double_value>,
      "td":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_vlan_binding/td_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"nstrafficdomain_vlan_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstrafficdomain_vxlan_binding

May 20, 2016

Binding object showing the vxlan that can be bound to nstrafficdomain.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
td	<Double>	Read-write	Integer value that uniquely identifies a traffic domain. Minimum value = 1 Maximum value = 4094
vxlan	<Double>	Read-write	ID of the VXLAN to bind to this traffic domain. More than one VXLAN can be bound to a traffic domain, but the same VXLAN cannot be a part of multiple traffic domains. Minimum value = 1 Maximum value = 16777215
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/nstrafficdomain_vxlan_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"nstrafficdomain_vxlan_binding":{
    "td":<Double_value>,
    "vxlan":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of nstrafficdomain_vxlan_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nstrafficdomain_vxlan_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstrafficdomain_vxlan_binding": [ {
    "td":<Double_value>,
    "vxlan":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nstrafficdomain_vxlan_binding/td_value<Double>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"nstrafficdomain_vxlan_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsversion

May 20, 2016

Configuration for version resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
version	<String>	Read-only	Version.
mode	<Double>	Read-only	Kernel mode (KMPE/VMPE).

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsversion

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsversion": [ {
      "version":<String_value>,
      "mode":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsweblogparam

May 20, 2016

Configuration for Web log parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
buffersizemb	<Double>	Read-write	Buffer size, in MB, allocated for log transaction data on the system. The maximum value is limited to the memory available on the system. Default value: 16 Minimum value = 1 Maximum value = 4294967294LU
customreqhdrs	<String[]>	Read-write	Name(s) of HTTP request headers whose values should be exported by the Web Logging feature. Minimum length = 1
customrsphdrs	<String[]>	Read-write	Name(s) of HTTP response headers whose values should be exported by the Web Logging feature. Minimum length = 1

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/nsweblogparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsweblogparam":{
    "buffersizemb":<Double_value>,
    "customreqhdrs":<String[]_value>,
    "customrsphdrs":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsweblogparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsweblogparam":{
    "buffersizemb":true,
    "customreqhdrs":true,
    "customrsphdrs":true
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/nsweblogparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsweblogparam": [ {
      "buffersizemb":<Double_value>,
      "customreqhdrs":<String[]_value>,
      "customrsphdrs":<String[]_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsxmlnamespace

May 20, 2016

Configuration for XML namespace resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
prefix	<String>	Read-write	XML prefix. Minimum length = 1
Namespace	<String>	Read-write	Expanded namespace for which the XML prefix is provided. Minimum length = 1
description	<String>	Read-write	Description for the prefix. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsxmlnamespace":{
    "prefix":<String_value>,
    "Namespace":<String_value>,
    "description":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace/prefix_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsxmlnamespace":{
    "prefix":<String_value>,
    "Namespace":<String_value>,
    "description":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"nsxmlnamespace":{
    "prefix":<String_value>,
    "Namespace":true,
    "description":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of nsxmlnamespace resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace?**pagesize=#no&pageno=#no**

Use this query-parameter to get the nsxmlnamespace resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsxmlnamespace": [ {
      "prefix":<String_value>,
      "Namespace":<String_value>,
      "description":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace/prefix_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace/prefix_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace/prefix_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsxmlnamespace": [ {
      "prefix":<String_value>,
      "Namespace":<String_value>,
      "description":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/nsxmlnamespace?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "nsxmlnamespace": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

reboot

May 20, 2016

Configuration for 0 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
warm	<Boolean>	Read-write	Restarts the NetScaler software without rebooting the underlying operating system. The session terminates and you must log on to the appliance after it has restarted. Note: This argument is required only for nCore appliances. Classic appliances ignore this argument.

Operations (click to see Properties)

REBOOT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Reboot

URL: http://<netscaler-ip-address>/nitro/v1/config/reboot

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"reboot":{
    "warm":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

shutdown

May 20, 2016

Configuration for 0 resource.

Operations (click to see Properties)

SHUTDOWN

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Shutdown

URL: http://<netscaler-ip-address>/nitro/v1/config/shutdown

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"shutdown":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NTP

May 20, 2016

NTP configuration.

Name	Description
ntpparam	Configuration for NTP parameter
ntpserver	Configuration for NTP server
ntpstatus	Configuration for ntp status
ntpsync	Configuration for NTP sync

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ntpparam

May 20, 2016

Configuration for NTP parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
authentication	<String>	Read-write	Apply NTP authentication, which enables the NTP client (NetScaler) to verify that the server is in fact known and trusted. Default value: YES Possible values = YES, NO
trustedkey	<Double[]>	Read-write	Key identifiers that are trusted for server authentication with symmetric key cryptography in the keys file. Minimum value = 1 Maximum value = 65534
autokeylogsec	<Double>	Read-write	Autokey protocol requires the keys to be refreshed periodically. This parameter specifies the interval between regenerations of new session keys. In seconds, expressed as a power of 2. Default value: 12 Minimum value = 0 Maximum value = 32
revokelogsec	<Double>	Read-write	Interval between re-randomizations of the autokey seeds to prevent brute-force attacks on the autokey algorithms. Default value: 16 Minimum value = 0 Maximum value = 32

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ntpparam":{
    "authentication":<String_value>,
    "trustedkey":<Double[]_value>,
    "autokeylogsec":<Double_value>,
    "revokelogsec":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpparam?**action=unset**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ntpparam":{
    "authentication":true,
    "trustedkey":true,
    "autokeylogsec":true,
    "revokelogsec":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ntpparam": [ {
    "authentication":<String_value>,
    "trustedkey":<Double[]_value>,
    "autokeylogsec":<Double_value>,
    "revokelogsec":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ntpserver

May 20, 2016

Configuration for NTP server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
serverip	<String>	Read-write	IP address of the NTP server. Minimum length = 1
servername	<String>	Read-write	Fully qualified domain name of the NTP server.
minpoll	<Double>	Read-write	Minimum time after which the NTP server must poll the NTP messages. In seconds, expressed as a power of 2. Minimum value = 4 Maximum value = 17
maxpoll	<Double>	Read-write	Maximum time after which the NTP server must poll the NTP messages. In seconds, expressed as a power of 2. Minimum value = 4 Maximum value = 17
autokey	<Boolean>	Read-write	Use the Autokey protocol for key management for this server, with the cryptographic values (for example, symmetric key, host and public certificate files, and sign key) generated by the ntp-keygen utility. To require authentication for communication with the server, you must set either the value of this parameter or the key parameter.
key	<Double>	Read-write	Key to use for encrypting authentication fields. All packets sent to and received from the server must include authentication fields encrypted by using this key. To require authentication for communication with the server, you must set either the value of this parameter or the autokey parameter. Minimum value = 1 Maximum value = 65534
preferredntpserver	<String>	Read-write	Preferred NTP server. The NetScaler appliance chooses this NTP server for time synchronization among a set of correctly operating hosts. Default value: NO Possible values = YES, NO
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD | DELETE | UPDATE | UNSET | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpserver

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ntpserver":{
    "serverip":<String_value>,
    "servername":<String_value>,
    "minpoll":<Double_value>,
    "maxpoll":<Double_value>,
    "autokey":<Boolean_value>,
    "key":<Double_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpserver/serverip_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpserver

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ntpserver":{
    "serverip":<String_value>,
    "servername":<String_value>,
    "minpoll":<Double_value>,
    "maxpoll":<Double_value>,
    "preferredntpserver":<String_value>,
    "autokey":<Boolean_value>,
    "key":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpserver?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ntpserver":{
    "serverip":<String_value>,
    "servername":<String_value>,
    "autokey":true,
    "minpoll":true,
    "maxpoll":true,
    "preferredntpserver":true,
    "key":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpserver

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ntpserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/ntpserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ntpserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/ntpserver?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/ntpserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the ntpserver resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ntpserver": [ {
      "serverip":<String_value>,
      "servername":<String_value>,
      "minpoll":<Double_value>,
      "maxpoll":<Double_value>,
      "preferredntpserver":<String_value>,
      "autokey":<Boolean_value>,
      "key":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "ntpserver": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ntpstatus

May 20, 2016

Configuration for ntp status resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
response	<String>	Read-only	.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpstatus

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ntpstatus": [ {
      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ntpsync

May 20, 2016

Configuration for NTP sync resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
state	<String>	Read-only	Show NTP status. Possible values = ENABLED, DISABLED

Operations (click to see Properties)

ENABLE | DISABLE | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

enable

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpsync?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ntpsync":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpsync?**action=disable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ntpsync":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ntpsync

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ntpsync": [ {
      "state":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Policy

May 20, 2016

Policy configuration.

Name	Description
policydataset	Configuration for TYPE set
policydataset_binding	Binding object showing the resources that can be bound to policydataset
policydataset_value_binding	Binding object showing the value that can be bound to policydataset
policyevaluation	Configuration for expr evaluation
policyexpression	Configuration for expression
policyhttpcallout	Configuration for HTTP callout
policymap	Configuration for map policy
policypatset	Configuration for PAT set
policypatset_binding	Binding object showing the resources that can be bound to policypatset
policypatset_pattern_binding	Binding object showing the pattern that can be bound to policypatset
policystringmap	Configuration for string map
policystringmap_binding	Binding object showing the resources that can be bound to policystringmap
policystringmap_pattern_binding	Binding object showing the pattern that can be bound to policystringmap

© 1999-2017 Citrix Systems, Inc. All rights reserved.

policydataset

Jul 07, 2017

Configuration for TYPE set resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the dataset. Must not exceed 127 characters. Minimum length = 1
type	<String>	Read-write	Type of value to bind to the dataset. Possible values = ipv4, number, ipv6, ulong, double, mac
indextype	<String>	Read-write	Index type. Possible values = Auto-generated, User-defined
comment	<String>	Read-write	Any comments to preserve information about this dataset.
description	<String>	Read-only	Description of the set.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/policydataset

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policydataset":{
    "name":<String_value>,
    "type":<String_value>,
    "indextype":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/policydataset/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/policydataset

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policydataset?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/policydataset?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policydataset resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/policydataset?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/policydataset?**pagesize=#no&pageno=#no**

Use this query-parameter to get the policydataset resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "policydataset": [ {
      "name":<String_value>,
      "description":<String_value>,
      "type":<String_value>,
      "indextype":<String_value>,
      "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policydataset/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policydataset/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/policydataset/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policydataset": [ {
      "name":<String_value>,
      "description":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "type":<String_value>,
    "indextype":<String_value>,
    "comment":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/policydataset?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policydataset": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

policydataset_binding

May 20, 2016

Binding object which returns the resources bound to policydataset.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the dataset. Must not exceed 127 characters. Minimum length = 1
policydataset_value_binding	<policydataset_value_binding[]>	Read-only	value that can be bound to policydataset.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policydataset_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "policydataset_binding": [ {
      "name":<String_value>,
      "policydataset_value_binding":<policydataset_value_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

policydataset_value_binding

May 20, 2016

Binding object showing the value that can be bound to policydataset.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
value	<String>	Read-write	Value of the specified type that is associated with the dataset.
name	<String>	Read-write	Name of the dataset to which to bind the value. Minimum length = 1
index	<Double>	Read-write	The index of the value (ipv4, ipv6, number) associated with the set.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/policydataset_value_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"policydataset_value_binding":{
    "name":<String_value>,
    "value":<String_value>,
    "index":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/policydataset_value_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/policydataset_value_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/policydataset_value_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of policydataset_value_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/policydataset_value_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the policydataset_value_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policydataset_value_binding": [ {
    "value":<String_value>,
    "name":<String_value>,
    "index":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/policydataset_value_binding/name_value<String>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"policydataset_value_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

policyevaluation

May 20, 2016

Configuration for expr evaluation resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
expression	<String>	Read-write	Expression string. For example: http.req.body(100).contains("this").
action	<String>	Read-write	Rewrite action name. Supported rewrite action types are: -delete -delete_all -delete_http_header -insert_after -insert_after_all -insert_before -insert_before_all -insert_http_header -replace -replace_all . Minimum length = 1
type	<String>	Read-write	Indicates request or response input packet. Possible values = HTTP_REQ, HTTP_RES, TEXT
input	<String>	Read-write	Text representation of input packet.
pitmodifiedinputdata	<String>	Read-only	Text representation of packet after evaluating expression or rewrite action.
pitboolresult	<Boolean>	Read-only	Result of the expression in bool format.
pitnumresult	<Double>	Read-only	Result of the expression in num format.
pitdoubleresult	<Double>	Read-only	Result of the expression in double format.
pitulongresult	<Double>	Read-only	Result of the expression in unsigned long format.
pitrefresult	<String>	Read-only	Result of the expression in string format.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pitoffsetresult	<Double>	Read-only	Offset of the resultant sting.
Name	**Data Type**	**Permissions**	**Description**
pitoffsetresultlen	<Double>	Read-only	Offset length of the resultant sting.
istruncatedrefresult	<Boolean>	Read-only	Identify whether ref result is truncated result.
pitboolevaltime	<Double>	Read-only	Average evaluation time of bool type expression in nanoseconds.
pitnumevaltime	<Double>	Read-only	Average evaluation time of num type expression in nanoseconds.
pitdoubleevaltime	<Double>	Read-only	Average evaluation time of double type expression in nanoseconds.
pitulongevaltime	<Double>	Read-only	Average evaluation time of unsigned long type expression in nanoseconds.
pitrefevaltime	<Double>	Read-only	Average evaluation time of string type expression in nanoseconds.
pitoffsetevaltime	<Double>	Read-only	Average evaluation time in finding offset of the resultant string in the input. Time is in nanoseconds.
pitactionevaltime	<Double>	Read-only	Average evaluation time of rewrite action in nanoseconds.
pitoperationperformerarray	<String[]>	Read-only	Details of the operation NS performed at various offsets during applying of rewrite action on input data. Operation can be insertion, modfication or deletion. Possible values = INSERT, MODIFY, DELETE
pitoldoffsetarray	<Double[]>	Read-only	Details of the offsets in the input data at which NS either inserted or modified or deleted data during applying of rewrite action.
pitnewoffsetarray	<Double[]>	Read-only	Details of the offsets in the output data at which NS either inserted or modified or deleted data during applying of rewrite action.
pitoffsetlengtharray	<Integer[]>	Read-only	Details of the lengths of the data which NS either inserted or modified or deleted during applying of rewrite action.
pitboolerrorresult	<String>	Read-only	Result of the bool type expression if any error occurs during evaluation. Result will be in string format.
pitnumerrorresult	<String>	Read-only	Result of the num type expression if any error occurs during evaluation. Result will be in string format.
pitdoubleerrorresult	<String>	Read-only	Result of the double type expression if any error occurs during evaluation. Result will be in string format.
			Result of the unsigned long type expression if any error occurs during

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
pitreferrorresult	<String>	Read-only	Result of the ref type expression if any error occurs during evaluation. Result will be in string format.
pitoffseterrorresult	<String>	Read-only	Result of the expression if any error occurs in calculating offset. Result will be in string format.
pitactionerrorresult	<String>	Read-only	Result of the action if any error occurs in evaluation. Result will be in string format.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/policyevaluation

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

args

http://<netscaler-ip-address>/nitro/v1/config/policyevaluation?**args=expression:<String_value>,action:<String_value>,type:<String_value>,input:<String_value>,**

Use this query-parameter to get policyevaluation resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/policyevaluation?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/policyevaluation?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policyevaluation resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/policyevaluation?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/policyevaluation?**pagesize=#no&pageno=#no**

Use this query-parameter to get the policyevaluation resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policyevaluation": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
expression:<String_value>,action:<String_value>,type:<String_value>,input:<String_value>,      "pitmodifiedinputdata":<String_value>,
      "pitboolresult":<Boolean_value>,
      "pitnumresult":<Double_value>,
      "pitdoubleresult":<Double_value>,
      "pitulongresult":<Double_value>,
      "pitrefresult":<String_value>,
      "pitoffsetresult":<Double_value>,
      "pitoffsetresultlen":<Double_value>,
      "istruncatedrefresult":<Boolean_value>,
      "pitboolevaltime":<Double_value>,
      "pitnumevaltime":<Double_value>,
      "pitdoubleevaltime":<Double_value>,
      "pitulongevaltime":<Double_value>,
      "pitrefevaltime":<Double_value>,
      "pitoffsetevaltime":<Double_value>,
      "pitactionevaltime":<Double_value>,
      "pitoperationperformerarray":<String[]_value>,
      "pitoldoffsetarray":<Double[]_value>,
      "pitnewoffsetarray":<Double[]_value>,
      "pitoffsetlengtharray":<Integer[]_value>,
      "pitboolerrorresult":<String_value>,
      "pitnumerrorresult":<String_value>,
      "pitdoubleerrorresult":<String_value>,
      "pitulongerrorresult":<String_value>,
      "pitreferrorresult":<String_value>,
      "pitoffseterrorresult":<String_value>,
      "pitactionerrorresult":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/policyevaluation?**args=expression:<String_value>,action:<String_value>,type:<String_value>,input:<String_value>,&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "policyevaluation": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

policyexpression

May 20, 2016

Configuration for expression resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Unique name for the expression. Not case sensitive. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Must not begin with 're' or 'xp' or be a word reserved for use as a default syntax expression qualifier prefix (such as HTTP) or enumeration value (such as ASCII). Must not be the name of an existing named expression, pattern set, dataset, stringmap, or HTTP callout. Minimum length = 1
value	<String>	Read-write	Expression string. For example: http.req.body(100).contains("this").
description	<String>	Read-write	Description for the expression.
comment	<String>	Read-write	Any comments associated with the expression. Displayed upon viewing the policy expression.
clientsecuritymessage	<String>	Read-write	Message to display if the expression fails. Allowed for classic end-point check expressions only. Minimum length = 1
type	<String>	Read-write	Type of expression. Can be a classic or default syntax (advanced) expression. Possible values = CLASSIC, ADVANCED
hits	<Double>	Read-only	The total number of hits.
pihits	<Double>	Read-only	The total number of hits.
type1	<String>	Read-only	The type of expression. This is for output only. Possible values = CLASSIC, ADVANCED
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default policy expression.
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/policyexpression

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policyexpression":{
    "name":<String_value>,
    "value":<String_value>,
    "description":<String_value>,
    "comment":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "clientsecuritymessage":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/policyexpression/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/policyexpression

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policyexpression":{
    "name":<String_value>,
    "value":<String_value>,
    "description":<String_value>,
    "comment":<String_value>,
    "clientsecuritymessage":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/policyexpression?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policyexpression":{
    "name":<String_value>,
    "description":true,
    "comment":true,
    "clientsecuritymessage":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/policyexpression

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/policyexpression?**args=name:<String_value>,type:<String_value>**

Use this query-parameter to get policyexpression resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/policyexpression?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/policyexpression?**filter=property-name1:property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

val1,property-name2:property-val2

Use this query-parameter to get the filtered set of policyexpression resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/policyexpression?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/policyexpression?**pagesize=#no&pageno=#no**

Use this query-parameter to get the policyexpression resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policyexpression": [ {
name:<String_value>,type:<String_value>      "value":<String_value>,
      "hits":<Double_value>,
      "pihits":<Double_value>,
      "type1":<String_value>,
      "clientsecuritymessage":<String_value>,
      "description":<String_value>,
      "comment":<String_value>,
      "isdefault":<Boolean_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policyexpression/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policyexpression/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/policyexpression/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policyexpression": [ {
name:<String_value>,type:<String_value>      "value":<String_value>,
      "hits":<Double_value>,
      "pihits":<Double_value>,
      "type1":<String_value>,
      "clientsecuritymessage":<String_value>,
      "description":<String_value>,
      "comment":<String_value>,
      "isdefault":<Boolean_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/policyexpression?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "policyexpression": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

policyhttpcallout

May 20, 2016

Configuration for HTTP callout resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the HTTP callout. Not case sensitive. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Must not begin with 're' or 'xp' or be a word reserved for use as a default syntax expression qualifier prefix (such as HTTP) or enumeration value (such as ASCII). Must not be the name of an existing named expression, pattern set, dataset, stringmap, or HTTP callout. Minimum length = 1
ipaddress	<String>	Read-write	IP Address of the server (callout agent) to which the callout is sent. Can be an IPv4 or IPv6 address. Mutually exclusive with the Virtual Server parameter. Therefore, you cannot set the <IP Address, Port> and the Virtual Server in the same HTTP callout.
port	<Integer>	Read-write	Server port to which the HTTP callout agent is mapped. Mutually exclusive with the Virtual Server parameter. Therefore, you cannot set the <IP Address, Port> and the Virtual Server in the same HTTP callout. Minimum value = 1
vserver	<String>	Read-write	Name of the load balancing, content switching, or cache redirection virtual server (the callout agent) to which the HTTP callout is sent. The service type of the virtual server must be HTTP. Mutually exclusive with the IP address and port parameters. Therefore, you cannot set the <IP Address, Port> and the Virtual Server in the same HTTP callout. Minimum length = 1
returntype	<String>	Read-write	Type of data that the target callout agent returns in response to the callout. Available settings function as follows: * TEXT - Treat the returned value as a text string. * NUM - Treat the returned value as a number. * BOOL - Treat the returned value as a Boolean value. Note: You cannot change the return type after it is set. Possible values = BOOL, NUM, TEXT
httpmethod	<String>	Read-write	Method used in the HTTP request that this callout sends. Mutually exclusive with the full HTTP request expression. Possible values = GET, POST
hostexpr	<String>	Read-write	Default Syntax string expression to configure the Host header. Can contain a literal value (for example, 10.101.10.11) or a derived value (for example, http.req.header("Host")). The literal value can be an IP address or a fully qualified

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
urlstemexpr	<String>	Read-write	Default Syntax string expression for generating the URL stem. Can contain a literal string (for example, "/mysite/index.html") or an expression that derives the value (for example, http.req.url). Mutually exclusive with the full HTTP request expression. Minimum length = 1
headers	<String[]>	Read-write	One or more headers to insert into the HTTP request. Each header is specified as "name(expr)", where expr is a default syntax expression that is evaluated at runtime to provide the value for the named header. You can configure a maximum of eight headers for an HTTP callout. Mutually exclusive with the full HTTP request expression.
parameters	<String[]>	Read-write	One or more query parameters to insert into the HTTP request URL (for a GET request) or into the request body (for a POST request). Each parameter is specified as "name(expr)", where expr is an default syntax expression that is evaluated at run time to provide the value for the named parameter (name=value). The parameter values are URL encoded. Mutually exclusive with the full HTTP request expression.
bodyexpr	<String>	Read-write	An advanced string expression for generating the body of the request. The expression can contain a literal string or an expression that derives the value (for example, client.ip.src). Mutually exclusive with -fullReqExpr. Minimum length = 1
fullreqexpr	<String>	Read-write	Exact HTTP request, in the form of a default syntax expression, which the NetScaler appliance sends to the callout agent. If you set this parameter, you must not include HTTP method, host expression, URL stem expression, headers, or parameters. The request expression is constrained by the feature for which the callout is used. For example, an HTTP.RES expression cannot be used in a request-time policy bank or in a TCP content switching policy bank. The NetScaler appliance does not check the validity of this request. You must manually validate the request. Minimum length = 1
scheme	<String>	Read-write	Type of scheme for the callout server. Possible values = http, https
resultexpr	<String>	Read-write	Expression that extracts the callout results from the response sent by the HTTP callout agent. Must be a response based expression, that is, it must begin with HTTP.RES. The operations in this expression must match the return type. For example, if you configure a return type of TEXT, the result expression must be a text based expression. If the return type is NUM, the result expression (resultExpr) must return a numeric value, as in the following example: http.res.body(10000).length. Minimum length = 1
cacheforsecs	<Double>	Read-write	Duration, in seconds, for which the callout response is cached. The cached responses are stored in an integrated caching content group named "calloutContentGroup". If no duration is configured, the callout responses will not be cached unless normal caching configuration is used to cache them. This parameter takes precedence over any normal caching configuration that would otherwise apply to these responses. Note that the calloutContentGroup definition may not be modified or removed nor may it be used with other cache policies. Minimum value = 1 Maximum value = 31536000

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
hits	<Double>	Read-only	Total hits.
undefhits	<Double>	Read-only	Total undefs.
svrstate	<String>	Read-only	The state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
effectivestate	<String>	Read-only	The effective state of the service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
undefreason	<String>	Read-only	Reason for last undef. Possible values = Failed to add service, Vserver not found, Not a HTTP or SSL vserver, Generated callout request is invalid, Content-Length header not found in callout request, Not enough space to put Content-Length value, Config incomplete, Server is DOWN, Creating callout connection failed, No memory to generate callout request packets, No memory to create callout task, No memory to create callout async, Callout request expression undef, No callout response expression, Skipped callout response eval, Callout response pixl init undef, Callout response expression undef
recursivecallout	<Double>	Read-only	Number of recursive callouts.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policyhttpcallout":{
    "name":<String_value>,
    "ipaddress":<String_value>,
    "port":<Integer_value>,
    "vserver":<String_value>,
    "returntype":<String_value>,
    "httpmethod":<String_value>,
    "hostexpr":<String_value>,
    "urlstemexpr":<String_value>,
    "headers":<String[]_value>,
    "parameters":<String[]_value>,
    "bodyexpr":<String_value>,
    "fullreqexpr":<String_value>,
    "scheme":<String_value>,
    "resultexpr":<String_value>,
    "cacheforsecs":<Double_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policyhttpcallout":{
    "name":<String_value>,
    "ipaddress":<String_value>,
    "port":<Integer_value>,
    "vserver":<String_value>,
    "returntype":<String_value>,
    "httpmethod":<String_value>,
    "hostexpr":<String_value>,
    "urlstemexpr":<String_value>,
    "headers":<String[]_value>,
    "parameters":<String[]_value>,
    "bodyexpr":<String_value>,
    "fullreqexpr":<String_value>,
    "scheme":<String_value>,
    "resultexpr":<String_value>,
    "cacheforsecs":<Double_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policyhttpcallout":{
    "name":<String_value>,
    "ipaddress":true,
    "port":true,
    "vserver":true,
    "httpmethod":true,
    "hostexpr":true,
    "urlstemexpr":true,
    "headers":true,
    "parameters":true,
    "bodyexpr":true,
    "fullreqexpr":true,
    "resultexpr":true,
    "cacheforsecs":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policyhttpcallout resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout?**pagesize=#no&pageno=#no**

Use this query-parameter to get the policyhttpcallout resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policyhttpcallout": [ {
    "name":<String_value>,
    "ipaddress":<String_value>,
    "port":<Integer_value>,
    "vserver":<String_value>,
    "returntype":<String_value>,
    "scheme":<String_value>,
    "httpmethod":<String_value>,
    "hostexpr":<String_value>,
    "urlstemexpr":<String_value>,
    "headers":<String[]_value>,
    "parameters":<String[]_value>,
    "fullreqexpr":<String_value>,
    "resultexpr":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "svrstate":<String_value>,
    "effectivestate":<String_value>,
    "undefreason":<String_value>,
    "recursivecallout":<Double_value>,
    "bodyexpr":<String_value>,
    "cacheforsecs":<Double_value>,
    "comment":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policyhttpcallout": [ {
      "name":<String_value>,
      "ipaddress":<String_value>,
      "port":<Integer_value>,
      "vserver":<String_value>,
      "returntype":<String_value>,
      "scheme":<String_value>,
      "httpmethod":<String_value>,
      "hostexpr":<String_value>,
      "urlstemexpr":<String_value>,
      "headers":<String[]_value>,
      "parameters":<String[]_value>,
      "fullreqexpr":<String_value>,
      "resultexpr":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "svrstate":<String_value>,
    "effectivestate":<String_value>,
    "undefreason":<String_value>,
    "recursivecallout":<Double_value>,
    "bodyexpr":<String_value>,
    "cacheforsecs":<Double_value>,
    "comment":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/policyhttpcallout?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policyhttpcallout": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

policymap

May 20, 2016

Configuration for map policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
mappolicyname	<String>	Read-write	Name for the map policy. Must begin with a letter, number, or the underscore (_) character and must consist only of letters, numbers, and the hash (#), period (.), colon (:), space ( ), at (@), equals (=), hyphen (-), and underscore (_) characters. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my map" or 'my map'). Minimum length = 1
sd	<String>	Read-write	Publicly known source domain name. This is the domain name with which a client request arrives at a reverse proxy virtual server for cache redirection. If you specify a source domain, you must specify a target domain. Minimum length = 1
su	<String>	Read-write	Source URL. Specify all or part of the source URL, in the following format: /[[prefix] [*]] [.suffix]. Minimum length = 1
td	<String>	Read-write	Target domain name sent to the server. The source domain name is replaced with this domain name. Minimum length = 1
tu	<String>	Read-write	Target URL. Specify the target URL in the following format: /[[prefix] [*]][.suffix]. Minimum length = 1
targetname	<String>	Read-only	The expression string.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/policymap

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policymap":{
    "mappolicyname":<String_value>,
    "sd":<String_value>,
    "su":<String_value>,
    "td":<String_value>,
    "tu":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/policymap/mappolicyname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/policymap

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policymap?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/policymap?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policymap resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/policymap?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/policymap?**pagesize=#no&pageno=#no**

Use this query-parameter to get the policymap resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policymap": [ {
      "mappolicyname":<String_value>,
      "sd":<String_value>,
      "su":<String_value>,
      "td":<String_value>,
      "tu":<String_value>,
      "targetname":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policymap/mappolicyname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policymap/mappolicyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/policymap/mappolicyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policymap": [ {
    "mappolicyname":<String_value>,
    "sd":<String_value>,
    "su":<String_value>,
    "td":<String_value>,
    "tu":<String_value>,
    "targetname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/policymap?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policymap": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

policypatset

May 20, 2016

Configuration for PAT set resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Unique name of the pattern set. Not case sensitive. Must begin with an ASCII letter or underscore (_) character and must contain only alphanumeric and underscore characters. Must not be the name of an existing named expression, pattern set, dataset, string map, or HTTP callout. Minimum length = 1
indextype	<String>	Read-write	Index type. Possible values = Auto-generated, User-defined
comment	<String>	Read-write	Any comments to preserve information about this patset.
description	<String>	Read-only	Description of the patset.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/policypatset

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policypatset":{
    "name":<String_value>,
    "indextype":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/policypatset/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/policypatset

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policypatset?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/policypatset?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policypatset resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/policypatset?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/policypatset?**pagesize=#no&pageno=#no**

Use this query-parameter to get the policypatset resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "policypatset": [ {
    "name":<String_value>,
    "description":<String_value>,
    "indextype":<String_value>,
    "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policypatset/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policypatset/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/policypatset/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policypatset": [ {
    "name":<String_value>,
    "description":<String_value>,
    "indextype":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "comment":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/policypatset?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policypatset": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

policypatset_binding

May 20, 2016

Binding object which returns the resources bound to policypatset.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the pattern set for which to display the detailed information. If a name is not provided, a list of all pattern sets configured on the appliance is shown. Minimum length = 1
policypatset_pattern_binding	<policypatset_pattern_binding[]>	Read-only	pattern that can be bound to policypatset.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policypatset_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "policypatset_binding": [ {
    "name":<String_value>,
    "policypatset_pattern_binding":<policypatset_pattern_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

policypatset_pattern_binding

Jul 07, 2017

Binding object showing the pattern that can be bound to policypatset.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
String	<String>	Read-write	String of characters that constitutes a pattern. For more information about the characters that can be used, refer to the character set parameter. Note: Minimum length for pattern sets used in rewrite actions of type REPLACE_ALL, DELETE_ALL, INSERT_AFTER_ALL, and INSERT_BEFORE_ALL, is three characters.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
name	<String>	Read-write	Name of the pattern set to which to bind the string. Minimum length = 1
charset	<String>	Read-write	Character set associated with the characters in the string. Note: UTF-8 characters can be entered directly (if the UI supports it) or can be encoded as a sequence of hexadecimal bytes '\xNN'. For example, the UTF-8 character '?' can be encoded as '\xC3\xBC'. Possible values = ASCII, UTF_8
index	<Double>	Read-write	The index of the string associated with the patset.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/policypatset_pattern_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"policypatset_pattern_binding":{
    "name":<String_value>,
    "String":<String_value>,
    "index":<Double_value>,
    "charset":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/policypatset_pattern_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policypatset_pattern_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/policypatset_pattern_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of policypatset_pattern_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/policypatset_pattern_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the policypatset_pattern_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policypatset_pattern_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "String":<String_value>,
    "builtin":<String[]_value>,
    "name":<String_value>,
    "charset":<String_value>,
    "index":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/policypatset_pattern_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"policypatset_pattern_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

policystringmap

May 20, 2016

Configuration for string map resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Unique name for the string map. Not case sensitive. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Must not begin with 're' or 'xp' or be a word reserved for use as a default syntax expression qualifier prefix (such as HTTP) or enumeration value (such as ASCII). Must not be the name of an existing named expression, pattern set, dataset, string map, or HTTP callout. Minimum length = 1
comment	<String>	Read-write	Comments associated with the string map.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policystringmap":{
    "name":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policystringmap":{
    "name":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"policystringmap":{
    "name":<String_value>,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policystringmap?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/policystringmap?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of policystringmap resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/policystringmap?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/policystringmap?**pagesize=#no&pageno=#no**

Use this query-parameter to get the policystringmap resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policystringmap": [ {
      "name":<String_value>,
      "comment":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/policystringmap/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/policystringmap/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policystringmap": [ {
      "name":<String_value>,
      "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "policystringmap": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

policystringmap_binding

May 20, 2016

Binding object which returns the resources bound to policystringmap.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the string map to display. If a name is not provided, a list of all the configured string maps is shown. Minimum length = 1
policystringmap_pattern_binding	<policystringmap_pattern_binding[]>	Read-only	pattern that can be bound to policystringmap.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "policystringmap_binding": [ {
      "name":<String_value>,
      "policystringmap_pattern_binding":<policystringmap_pattern_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

policystringmap_pattern_binding

May 20, 2016

Binding object showing the pattern that can be bound to policystringmap.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
key	<String>	Read-write	Character string constituting the key to be bound to the string map. The key is matched against the data processed by the operation that uses the string map. The default character set is ASCII. UTF-8 characters can be included if the character set is UTF-8. UTF-8 characters can be entered directly (if the UI supports it) or can be encoded as a sequence of hexadecimal bytes '\xNN'. For example, the UTF-8 character '?' can be encoded as '\xC3\xBC'. Minimum length = 1
value	<String>	Read-write	Character string constituting the value associated with the key. This value is returned when processed data matches the associated key. Refer to the key parameter for details of the value character set. Minimum length = 1
name	<String>	Read-write	Name of the string map to which to bind the key-value pair. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/policystringmap_pattern_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"policystringmap_pattern_binding":{
    "name":<String_value>,
    "key":<String_value>,
    "value":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of policystringmap_pattern_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the policystringmap_pattern_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "policystringmap_pattern_binding": [ {
    "key":<String_value>,
    "value":<String_value>,
    "name":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/policystringmap_pattern_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"policystringmap_pattern_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Priority Queuing

Jul 07, 2017

Priority Queuing configuration. Priority Queuing ensures that key customer web sites get the timely responses they need. System provides priority-based queuing, which provides different levels of service to customers.

Name	Description
pqbinding	Configuration for PQ bindings
pqpolicy	Configuration for PQ policy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pqbinding

May 20, 2016

Configuration for PQ bindings resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vservername	<String>	Read-write	Name of the load balancing virtual server for which to display the priority queuing policy. Minimum length = 1
policyname	<String>	Read-only	The name of the priority queuing policy.
rule	<String>	Read-only	The condition for applying the policy.
priority	<Double>	Read-only	The priority of queuing the request.
weight	<Double>	Read-only	Weight.
qdepth	<Double>	Read-only	Queue Depth.
polqdepth	<Double>	Read-only	Policy Queue Depth.
hits	<Double>	Read-only	Total number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/pqbinding

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/pqbinding?**args=vservername:<String_value>,**

Use this query-parameter to get pqbinding resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/pqbinding?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/pqbinding?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of pqbinding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/pqbinding?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/pqbinding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the pqbinding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "pqbinding": [ {
vservername:<String_value>,      "policyname":<String_value>,
      "rule":<String_value>,
      "priority":<Double_value>,
      "weight":<Double_value>,
      "qdepth":<Double_value>,
      "polqdepth":<Double_value>,
      "hits":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/pqbinding?**args=vservername:<String_value>,&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "pqbinding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

pqpolicy

May 20, 2016

Configuration for PQ policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name for the priority queuing policy. Must begin with a letter, number, or the underscore symbol (_). Other characters allowed, after the first character, are the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), and colon (:) characters. Minimum length = 1
rule	<String>	Read-write	Expression or name of a named expression, against which the request is evaluated. The priority queuing policy is applied if the rule evaluates to true. Note: * On the command line interface, if the expression includes blank spaces, the entire expression must be enclosed in double quotation marks. * If the expression itself includes double quotation marks, you must escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you will not have to escape the double quotation marks. * Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"'.
priority	<Double>	Read-write	Priority for queuing the request. If server resources are not available for a request that matches the configured rule, this option specifies a priority for queuing the request until the server resources are available again. Enter the value of positive_integer as 1, 2 or 3. The highest priority level is 1 and the lowest priority value is 3. Minimum value = 1 Maximum value = 3
weight	<Double>	Read-write	Weight of the priority. Each priority is assigned a weight according to which it is served when server resources are available. The weight for a higher priority request must be set higher than that of a lower priority request. To prevent delays for low-priority requests across multiple priority levels, you can configure weighted queuing for serving requests. The default weights for the priorities are: * Gold - Priority 1 - Weight 3 * Silver - Priority 2 - Weight 2 * Bronze - Priority 3 - Weight 1 Specify the weights as 0 through 101. A weight of 0 indicates that the particular priority level should be served only when there are no requests in any of the priority queues. A weight of 101 specifies a weight of infinity. This means that this priority level is served

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 0 Maximum value = 101
qdepth	<Double>	Read-write	Queue depth threshold value. When the queue size (number of requests in the queue) on the virtual server to which this policy is bound, increases to the specified qDepth value, subsequent requests are dropped to the lowest priority level. Default value: 0 Minimum value = 0 Maximum value = 4294967294
polqdepth	<Double>	Read-write	Policy queue depth threshold value. When the policy queue size (number of requests in all the queues belonging to this policy) increases to the specified polqDepth value, subsequent requests are dropped to the lowest priority level. Default value: 0 Minimum value = 0 Maximum value = 4294967294
hits	<Double>	Read-only	Total number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/pqpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"pqpolicy":{
    "policyname":<String_value>,
    "rule":<String_value>,
    "priority":<Double_value>,
    "weight":<Double_value>,
    "qdepth":<Double_value>,
    "polqdepth":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/pqpolicy/policyname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/pqpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"pqpolicy":{
    "policyname":<String_value>,
    "weight":<Double_value>,
    "qdepth":<Double_value>,
    "polqdepth":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/pqpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"pqpolicy":{
    "policyname":<String_value>,
    "weight":true,
    "qdepth":true,
    "polqdepth":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/pqpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/pqpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/pqpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of pqpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/pqpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/pqpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the pqpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "pqpolicy": [ {
    "policyname":<String_value>,
    "rule":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "weight":<Double_value>,
    "qdepth":<Double_value>,
    "polqdepth":<Double_value>,
    "hits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/pqpolicy/policyname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/pqpolicy/policyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/pqpolicy/policyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "pqpolicy": [ {
    "policyname":<String_value>,
    "rule":<String_value>,
    "priority":<Double_value>,
    "weight":<Double_value>,
    "qdepth":<Double_value>,
    "polqdepth":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "hits":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/pqpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "pqpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Protocol

Jul 07, 2017

Protocol Configuration.

Name	Description
protocolhttpband	Configuration for HTTP request/response band

© 1999-2017 Citrix Systems, Inc. All rights reserved.

protocolhttpband

May 20, 2016

Configuration for HTTP request/response band resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
reqbandsize	<Integer>	Read-write	Band size, in bytes, for HTTP request band statistics. For example, if you specify a band size of 100 bytes, statistics will be maintained and displayed for the following size ranges: 0 - 99 bytes 100 - 199 bytes 200 - 299 bytes and so on. Default value: 100 Minimum value = 50
respbandsize	<Integer>	Read-write	Band size, in bytes, for HTTP response band statistics. For example, if you specify a band size of 100 bytes, statistics will be maintained and displayed for the following size ranges: 0 - 99 bytes 100 - 199 bytes 200 - 299 bytes and so on. Default value: 1024 Minimum value = 50
type	<String>	Read-write	Type of statistics to display. Possible values = REQUEST, RESPONSE
bandrange	<Integer>	Read-only	The range of the HTTP request/response size, in bytes.
numberofbands	<Integer>	Read-only	The total number of http bands.
totalbandsize	<Double[]>	Read-only	The total size of all HTTP requests/responses in this size range.
avgbandsize	<Double[]>	Read-only	The average size of all HTTP requests/responses in this size range.
avgbandsizenew	<Double[]>	Read-only	The average size of all HTTP requests/responses in this size range.
banddata	<Double[]>	Read-only	The total size of all HTTP requests/responses in this size range, expressed as a percentage of the total size of all HTTP requests/responses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			percentage of the total size of all HTTP requests/responses.
accesscount	<Double[]>	Read-only	The number of HTTP requests/responses in this size range.
accessratio	<Double[]>	Read-only	The number of HTTP requests/responses in this size range, expressed as a percentage of the total number of HTTP requests/responses.
accessrationew	<Double[]>	Read-only	The number of HTTP requests/responses in this size range, expressed as a percentage of the total number of HTTP requests/responses.
totals	<Integer[]>	Read-only	The total of totalBandSize, avgBandSize, BandData, accessCount, accessRatio respectively.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/protocolhttpband

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"protocolhttpband":{
    "reqbandsize":<Integer_value>,
    "respbandsize":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/protocolhttpband?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"protocolhttpband":{
    "reqbandsize":true,
    "respbandsize":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/protocolhttpband

Query-parameters:

args

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/protocolhttpband?**args=type:<String_value>,**

Use this query-parameter to get protocolhttpband resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "protocolhttpband": [ {
type:<String_value>,      "bandrange":<Integer_value>,
    "numberofbands":<Integer_value>,
    "totalbandsize":<Double[]_value>,
    "avgbandsize":<Double[]_value>,
    "avgbandsizenew":<Double[]_value>,
    "banddata":<Double[]_value>,
    "banddatanew":<Double[]_value>,
    "accesscount":<Double[]_value>,
    "accessratio":<Double[]_value>,
    "accessrationew":<Double[]_value>,
    "totals":<Integer[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rdp

May 20, 2016

RDP configuration.

Name	Description
rdpclientprofile	Configuration for RDP clientprofile
rdpconnections	Configuration for active rdp connections
rdpserverprofile	Configuration for RDP serverprofile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rdpclientprofile

May 20, 2016

Configuration for RDP clientprofile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the rdp profile. Minimum length = 1
rdpurloverride	<String>	Read-write	This setting determines whether the RDP parameters supplied in the vpn url override those specified in the RDP profile. Default value: ENABLE Possible values = ENABLE, DISABLE
redirectclipboard	<String>	Read-write	This setting corresponds to the Clipboard check box on the Local Resources tab under Options in RDC. Default value: ENABLE Possible values = ENABLE, DISABLE
redirectdrives	<String>	Read-write	This setting corresponds to the selections for Drives under More on the Local Resources tab under Options in RDC. Default value: DISABLE Possible values = ENABLE, DISABLE
redirectprinters	<String>	Read-write	This setting corresponds to the selection in the Printers check box on the Local Resources tab under Options in RDC. Default value: ENABLE Possible values = ENABLE, DISABLE
redirectcomports	<String>	Read-write	This setting corresponds to the selections for comports under More on the Local Resources tab under Options in RDC. Default value: DISABLE Possible values = ENABLE, DISABLE
redirectpnpdevices	<String>	Read-write	This setting corresponds to the selections for pnpdevices under More on the Local Resources tab under Options in RDC. Default value: DISABLE Possible values = ENABLE, DISABLE
keyboardhook	<String>	Read-write	This setting corresponds to the selection in the Keyboard drop-down list on the Local Resources tab under Options in RDC. Default value: InFullScreenMode Possible values = OnLocal, OnRemote, InFullScreenMode
			This setting corresponds to the selections in the Remote audio area on the Local

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: DISABLE Possible values = ENABLE, DISABLE
videoplaybackmode	<String>	Read-write	This setting determines if Remote Desktop Connection (RDC) will use RDP efficient multimedia streaming for video playback. Default value: ENABLE Possible values = ENABLE, DISABLE
multimonitorsupport	<String>	Read-write	Enable/Disable Multiple Monitor Support for Remote Desktop Connection (RDC). Default value: ENABLE Possible values = ENABLE, DISABLE
rdpcookievalidity	<Double>	Read-write	RDP cookie validity period. Default value: 60 Minimum value = 60 Maximum value = 86400
addusernameinrdpfile	<String>	Read-write	Add username in rdp file. Default value: NO Possible values = YES, NO
rdpfilename	<String>	Read-write	RDP file name to be sent to End User. Minimum length = 1
rdphost	<String>	Read-write	Fully-qualified domain name (FQDN) of the RDP Listener. Maximum length = 252
rdpcustomparams	<String>	Read-write	Option for RDP custom parameters settings (if any). Custom params needs to be separated by '&'. Default value: 0 Minimum length = 1
psk	<String>	Read-write	Pre shared key value. Default value: 0
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rdpclientprofile":{
    "name":<String_value>,
    "rdpurloverride":<String_value>,
    "redirectclipboard":<String_value>,
    "redirectdrives":<String_value>,
    "redirectprinters":<String_value>,
    "redirectcomports":<String_value>,
    "redirectpnpdevices":<String_value>,
    "keyboardhook":<String_value>,
    "audiocapturemode":<String_value>,
    "videoplaybackmode":<String_value>,
    "multimonitorsupport":<String_value>,
    "rdpcookievalidity":<Double_value>,
    "addusernameinrdpfile":<String_value>,
    "rdpfilename":<String_value>,
    "rdphost":<String_value>,
    "rdpcustomparams":<String_value>,
    "psk":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rdpclientprofile":{
    "name":<String_value>,
    "rdpurloverride":<String_value>,
    "redirectclipboard":<String_value>,
    "redirectdrives":<String_value>,
    "redirectprinters":<String_value>,
    "redirectcomports":<String_value>,
    "redirectpnpdevices":<String_value>,
    "keyboardhook":<String_value>,
    "audiocapturemode":<String_value>,
    "videoplaybackmode":<String_value>,
    "multimonitorsupport":<String_value>,
    "rdpcookievalidity":<Double_value>,
    "addusernameinrdpfile":<String_value>,
    "rdpfilename":<String_value>,
    "rdphost":<String_value>,
    "rdpcustomparams":<String_value>,
    "psk":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile?**action=unset**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rdpclientprofile":{
    "name":<String_value>,
    "rdpurloverride":true,
    "redirectclipboard":true,
    "redirectdrives":true,
    "redirectprinters":true,
    "redirectcomports":true,
    "redirectpnpdevices":true,
    "keyboardhook":true,
    "audiocapturemode":true,
    "videoplaybackmode":true,
    "multimonitorsupport":true,
    "rdpcookievalidity":true,
    "addusernameinrdpfile":true,
    "rdpfilename":true,
    "rdphost":true,
    "rdpcustomparams":true,
    "psk":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rdpclientprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rdpclientprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rdpclientprofile": [ {
    "name":<String_value>,
    "rdpurloverride":<String_value>,
    "redirectclipboard":<String_value>,
    "redirectdrives":<String_value>,
    "redirectprinters":<String_value>,
    "redirectcomports":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "redirectpnpdevices":<String_value>,
    "keyboardhook":<String_value>,
    "audiocapturemode":<String_value>,
    "videoplaybackmode":<String_value>,
    "multimonitorsupport":<String_value>,
    "rdpcookievalidity":<Double_value>,
    "addusernameinrdpfile":<String_value>,
    "rdpfilename":<String_value>,
    "rdphost":<String_value>,
    "rdpcustomparams":<String_value>,
    "psk":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "rdpclientprofile": [ {
    "name":<String_value>,
    "rdpurloverride":<String_value>,
    "redirectclipboard":<String_value>,
    "redirectdrives":<String_value>,
    "redirectprinters":<String_value>,
    "redirectcomports":<String_value>,
    "redirectpnpdevices":<String_value>,
    "keyboardhook":<String_value>,
    "audiocapturemode":<String_value>,
    "videoplaybackmode":<String_value>,
    "multimonitorsupport":<String_value>,
    "rdpcookievalidity":<Double_value>,
    "addusernameinrdpfile":<String_value>,
    "rdpfilename":<String_value>,
    "rdphost":<String_value>,
    "rdpcustomparams":<String_value>,
    "psk":<String_value>,
    "builtin":<String[]_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpclientprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rdpclientprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rdpconnections

May 20, 2016

Configuration for active rdp connections resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	User name for which to display connections. Minimum length = 1
domain	<String>	Read-only	The domain name.
srcip	<String>	Read-only	The client IP address.
srcport	<Integer>	Read-only	The client port. Range 1 - 65535
destip	<String>	Read-only	The Server IP address.
destport	<Integer>	Read-only	The Client IP address. Range 1 - 65535
peid	<Double>	Read-only	Core id of the session owner.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpconnections

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/rdpconnections?**args=username:<String_value>**

Use this query-parameter to get rdpconnections resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/rdpconnections?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/rdpconnections?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rdpconnections resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/rdpconnections?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/rdpconnections?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rdpconnections resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rdpconnections": [ {
username:<String_value>      "domain":<String_value>,
      "srcip":<String_value>,
      "srcport":<Integer_value>,
      "destip":<String_value>,
      "destport":<Integer_value>,
      "peid":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpconnections?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "rdpconnections": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rdpserverprofile

May 20, 2016

Configuration for RDP serverprofile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the rdp server profile. Minimum length = 1 Maximum length = 32
rdpip	<String>	Read-write	IPv4 or IPv6 address of RDP listener. This terminates client RDP connections. Minimum length = 1
rdpport	<Double>	Read-write	TCP port on which the RDP connection is established. Default value: 3389 Minimum value = 1 Maximum value = 65535
psk	<String>	Read-write	Pre shared key value.
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rdpserverprofile":{
    "name":<String_value>,
    "rdpip":<String_value>,
    "rdpport":<Double_value>,
    "psk":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"rdpserverprofile":{
    "name":<String_value>,
    "rdpip":<String_value>,
    "rdpport":<Double_value>,
    "psk":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rdpserverprofile":{
    "name":<String_value>,
    "rdpip":true,
    "rdpport":true,
    "psk":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rdpserverprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rdpserverprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rdpserverprofile": [ {
      "name":<String_value>,
      "rdpip":<String_value>,
      "rdpport":<Double_value>,
      "psk":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "rdpserverprofile": [ {
    "name":<String_value>,
    "rdpip":<String_value>,
    "rdpport":<Double_value>,
    "psk":<String_value>,
    "builtin":<String[]_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rdpserverprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rdpserverprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Reputation

Jul 07, 2017

Reputation services configuration.

Name	Description
reputationsettings	Configuration for Reputation service settings

© 1999-2017 Citrix Systems, Inc. All rights reserved.

reputationsettings

May 20, 2016

Configuration for Reputation service settings resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
proxyserver	<String>	Read-write	Proxy server IP to get Reputation data. Minimum length = 1
proxyport	<Double>	Read-write	Proxy server port.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/reputationsettings

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"reputationsettings":{
    "proxyserver":<String_value>,
    "proxyport":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/reputationsettings?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"reputationsettings":{
    "proxyserver":true,
    "proxyport":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/reputationsettings

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "reputationsettings": [ {
      "proxyserver":<String_value>,
      "proxyport":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Responder

Jul 07, 2017

Responder configuration.

Name	Description
responderaction	Configuration for responder action
responderglobal_binding	Binding object showing the resources that can be bound to responderglobal
responderglobal_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to responderglobal
responderhtmlpage	Configuration for Responder HTML page
responderparam	Configuration for responser parameter
responderpolicy	Configuration for responder policy
responderpolicy_binding	Binding object showing the resources that can be bound to responderpolicy
responderpolicy_crvserver_binding	Binding object showing the crvserver that can be bound to responderpolicy
responderpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to responderpolicy
responderpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to responderpolicy
responderpolicy_responderglobal_binding	Binding object showing the responderglobal that can be bound to responderpolicy
responderpolicy_responderpolicylabel_binding	Binding object showing the responderpolicylabel that can be bound to responderpolicy
responderpolicylabel	Configuration for responder policy label
responderpolicylabel_binding	Binding object showing the resources that can be bound to responderpolicylabel
responderpolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to responderpolicylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicylabel_responderpolicy_binding	
Name	responderpolicylabel **Description**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderaction

May 20, 2016

Configuration for responder action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the responder action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the responder policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my responder action" or 'my responder action').
type	<String>	Read-write	Type of responder action. Available settings function as follows: * respondwith <target> - Respond to the request with the expression specified as the target. * respondwithhtmlpage - Respond to the request with the uploaded HTML page object specified as the target. * redirect - Redirect the request to the URL specified as the target. * sqlresponse_ok - Send an SQL OK response. * sqlresponse_error - Send an SQL ERROR response. Possible values = noop, respondwith, redirect, respondwithhtmlpage, sqlresponse_ok, sqlresponse_error
target	<String>	Read-write	Expression specifying what to respond with. Typically a URL for redirect policies or a default-syntax expression. In addition to NetScaler default-syntax expressions that refer to information in the request, a stringbuilder expression can contain text and HTML, and simple escape codes that define new lines and paragraphs. Enclose each stringbuilder expression element (either a NetScaler default-syntax expression or a string) in double quotation marks. Use the plus (+) character to join the elements. Examples: 1) Respondwith expression that sends an HTTP 1.1 200 OK response: "HTTP/1.1 200 OK\r\n\r\n" 2) Redirect expression that redirects user to the specified web host and appends the request URL to the redirect. "http://backupsite2.com" + HTTP.REQ.URL 3) Respondwith expression that sends an HTTP 1.1 404 Not Found response with the request URL included in the response: "HTTP/1.1 404 Not Found\r\n\r\n"+ "HTTP.REQ.URL.HTTP_URL_SAFE" + "does not

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			The following requirement applies only to the NetScaler CLI: Enclose the entire expression in single quotation marks. (NetScaler default expression elements should be included inside the single quotation marks for the entire expression, but do not need to be enclosed in double quotation marks.).
htmlpage	<String>	Read-write	For respondwithhtmlpage policies, name of the HTML page object to use as the response. You must first import the page object. Minimum length = 1
bypasssafetycheck	<String>	Read-write	Bypass the safety check, allowing potentially unsafe expressions. An unsafe expression in a response is one that contains references to request elements that might not be present in all requests. If a response refers to a missing request element, an empty string is used instead. Default value: NO Possible values = YES, NO
comment	<String>	Read-write	Comment. Any type of information about this responder action.
responsestatuscode	<Double>	Read-write	HTTP response status code, for example 200, 302, 404, etc. The default value for the redirect action type is 302 and for respondwithhtmlpage is 200. Minimum value = 100 Maximum value = 599
reasonphrase	<String>	Read-write	Expression specifying the reason phrase of the HTTP response. The reason phrase may be a string literal with quotes or a PI expression. For example: "Invalid URL: " + HTTP.REQ.URL.
newname	<String>	Read-write	New name for the responder action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my responder action" or my responder action'). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
referencecount	<Double>	Read-only	The number of references to the action.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
builtin	<String[]>	Read-only	Flag to determine whether responder action is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/responderaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderaction":{
    "name":<String_value>,
    "type":<String_value>,
    "target":<String_value>,
    "htmlpage":<String_value>,
    "bypasssafetycheck":<String_value>,
    "comment":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "responsestatuscode":<Double_value>,
    "reasonphrase":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/responderaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/responderaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderaction":{
    "name":<String_value>,
    "target":<String_value>,
    "bypasssafetycheck":<String_value>,
    "htmlpage":<String_value>,
    "responsestatuscode":<Double_value>,
    "reasonphrase":<String_value>,
    "comment":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/responderaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderaction":{
    "name":<String_value>,
    "responsestatuscode":true,
    "reasonphrase":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/responderaction?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderaction":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/responderaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/responderaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/responderaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of responderaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/responderaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "responderaction": [ {
      "name":<String_value>,
      "type":<String_value>,
      "target":<String_value>,
      "htmlpage":<String_value>,
      "bypasssafetycheck":<String_value>,
      "hits":<Double_value>,
      "referencecount":<Double_value>,
      "undefhits":<Double_value>,
      "comment":<String_value>,
      "builtin":<String[]_value>,
      "responsestatuscode":<Double_value>,
      "reasonphrase":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/responderaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/responderaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "responderaction": [ {
    "name":<String_value>,
    "type":<String_value>,
    "target":<String_value>,
    "htmlpage":<String_value>,
    "bypasssafetycheck":<String_value>,
    "hits":<Double_value>,
    "referencecount":<Double_value>,
    "undefhits":<Double_value>,
    "comment":<String_value>,
    "builtin":<String[]_value>,
    "responsestatuscode":<Double_value>,
    "reasonphrase":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderglobal_binding

May 20, 2016

Binding object which returns the resources bound to responderglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
responderglobal_responderpolicy_binding	<responderglobal_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to responderglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/responderglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "responderglobal_binding": [ {
      "responderglobal_responderpolicy_binding":<responderglobal_responderpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderglobal_responderpolicy_binding

May 20, 2016

Binding object showing the responderpolicy that can be bound to responderglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the responder policy.
labelname	<String>	Read-write	Name of the policy label to invoke. If the current policy evaluates to TRUE, the invoke parameter is set, and Label Type is policylabel.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
type	<String>	Read-write	Specifies the bind point whose policies you want to display. Available settings function as follows: * REQ_OVERRIDE - Request override. Binds the policy to the priority request queue. * REQ_DEFAULT - Binds the policy to the default request queue. * OTHERTCP_REQ_OVERRIDE - Binds the policy to the non-HTTP TCP priority request queue. * OTHERTCP_REQ_DEFAULT - Binds the policy to the non-HTTP TCP default request queue.. * SIPUDP_REQ_OVERRIDE - Binds the policy to the SIP UDP priority response queue.. * SIPUDP_REQ_DEFAULT - Binds the policy to the SIP UDP default response queue. * RADIUS_REQ_OVERRIDE - Binds the policy to the RADIUS priority response queue.. * RADIUS_REQ_DEFAULT - Binds the policy to the RADIUS default response queue. * MSSQL_REQ_OVERRIDE - Binds the policy to the Microsoft SQL priority response queue.. * MSSQL_REQ_DEFAULT - Binds the policy to the Microsoft SQL default response queue. * MYSQL_REQ_OVERRIDE - Binds the policy to the MySQL priority response queue. * MYSQL_REQ_DEFAULT - Binds the policy to the MySQL default response queue. Possible values = REQ_OVERRIDE, REQ_DEFAULT, OVERRIDE, DEFAULT, OTHERTCP_REQ_OVERRIDE, OTHERTCP_REQ_DEFAULT, SIPUDP_REQ_OVERRIDE, SIPUDP_REQ_DEFAULT, SIPTCP_REQ_OVERRIDE, SIPTCP_REQ_DEFAULT, MSSQL_REQ_OVERRIDE, MSSQL_REQ_DEFAULT,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			NAT_REQ_DEFAULT, DIAMETER_REQ_OVERRIDE, DIAMETER_REQ_DEFAULT, RADIUS_REQ_OVERRIDE, RADIUS_REQ_DEFAULT, DNS_REQ_OVERRIDE, DNS_REQ_DEFAULT
labeltype	<String>	Read-write	Type of invocation, Available settings function as follows: * vserver - Forward the request to the specified virtual server. * policylabel - Invoke the specified policy label. Possible values = vserver, policylabel
numpol	<Double>	Read-only	number of polices bound to label.
flowtype	<Double>	Read-only	flowtype of the bound responder policy.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address/nitro/v1/config/responderglobal_responderpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"responderglobal_responderpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/responderglobal_responderpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderglobal_responderpolicy_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/responderglobal_responderpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderglobal_responderpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderglobal_responderpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderglobal_responderpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderglobal_responderpolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "type":<String_value>,
    "labeltype":<String_value>,
    "numpol":<Double_value>,
    "flowtype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderglobal_responderpolicy_binding?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "responderglobal_responderpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderhtmlpage

May 20, 2016

Configuration for Responder HTML page resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
src	<String>	Read-write	Local path to and name of, or URL \(protocol, host, path, and file name\) for, the file in which to store the imported HTML page. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
name	<String>	Read-write	Name to assign to the HTML page object on the NetScaler appliance. Minimum length = 1 Maximum length = 31
comment	<String>	Read-write	Any comments to preserve information about the HTML page object. Maximum length = 128
overwrite	<Boolean>	Read-write	Overwrites the existing file.
response	<String>	Read-only	.

Operations (click to see Properties)

IMPORT | DELETE | CHANGE | GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/responderhtmlpage?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderhtmlpage":{
    "src":<String_value>,
    "name":<String_value>,
    "comment":<String_value>,
    "overwrite":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/responderhtmlpage/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

change

URL: http://<netscaler-ip-address>/nitro/v1/config/responderhtmlpage?**action=update**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderhtmlpage":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/responderhtmlpage

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "responderhtmlpage": [ {
    "name":<String_value>,
    "response":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderhtmlpage/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/responderhtmlpage/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/responderhtmlpage/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderhtmlpage": [ {
    "name":<String_value>,
    "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderparam

Jul 07, 2017

Configuration for responser parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
undefaction	<String>	Read-write	Action to perform when policy evaluation creates an UNDEF condition. Available settings function as follows: * NOOP - Send the request to the protected server. * RESET - Reset the request and notify the user's browser, so that the user can resend the request. * DROP - Drop the request without sending a response to the user. Default value: "NOOP"

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/responderparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderparam":{
    "undefaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/responderparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderparam":{
    "undefaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/responderparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderparam": [ {
      "undefaction":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicy

May 20, 2016

Configuration for responder policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the responder policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the responder policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my responder policy" or 'my responder policy').
rule	<String>	Read-write	Default syntax expression that the policy uses to determine whether to respond to the specified request.
action	<String>	Read-write	Name of the responder action to perform if the request matches this responder policy. There are also some built-in actions which can be used. These are: * NOOP - Send the request to the protected server instead of responding to it. * RESET - Reset the client connection by closing it. The client program, such as a browser, will handle this and may inform the user. The client may then resend the request if desired. * DROP - Drop the request without sending a response to the user.
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.
comment	<String>	Read-write	Any type of information about this responder policy.
logaction	<String>	Read-write	Name of the messagelog action to use for requests that match this policy.
appflowaction	<String>	Read-write	AppFlow action to invoke for requests that match this policy.
newname	<String>	Read-write	New name for the responder policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
builtin	<String[]>	Read-only	Flag to determine if responder policy is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "appflowaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"responderpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>,
    "appflowaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderpolicy":{
    "name":<String_value>,
    "undefaction":true,
    "comment":true,
    "logaction":true,
    "appflowaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy?**action=rename**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderpolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of responderpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderpolicy resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "undefaction":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "comment":<String_value>,
      "logaction":<String_value>,
      "appflowaction":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "undefaction":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "comment":<String_value>,
      "logaction":<String_value>,
      "appflowaction":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicylabel

Jul 07, 2017

Configuration for responder policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name for the responder policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the responder policy label is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my responder policy label" or my responder policy label').
policylabeltype	<String>	Read-write	Type of responses sent by the policies bound to this policy label. Types are: * HTTP - HTTP responses. * OTHERTCP - NON-HTTP TCP responses. * SIP_UDP - SIP responses. * RADIUS - RADIUS responses. * MYSQL - SQL responses in MySQL format. * MSSQL - SQL responses in Microsoft SQL format. * NAT - NAT response. Default value: HTTP Possible values = HTTP, OTHERTCP, SIP_UDP, SIP_TCP, MYSQL, MSSQL, NAT, DIAMETER, RADIUS, DNS
comment	<String>	Read-write	Any comments to preserve information about this responder policy label.
newname	<String>	Read-write	New name for the responder policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Minimum length = 1
numpol	<Double>	Read-only	number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label to invoke. Available settings function as follows: * vserver - Invoke an unnamed policy label associated with a virtual server. * policylabel - Invoke a user-defined policy label. Possible values = vserver, policylabel
invoke_labelname	<String>	Read-only	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderpolicylabel":{
    "labelname":<String_value>,
    "policylabeltype":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"responderpolicylabel":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "labelname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of responderpolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderpolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicylabel": [ {
      "labelname":<String_value>,
      "policylabeltype":<String_value>,
      "numpol":<Double_value>,
      "hits":<Double_value>,
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "labeltype":<String_value>,
      "invoke_labelname":<String_value>,
      "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicylabel": [ {
    "labelname":<String_value>,
    "policylabeltype":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicylabel": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicylabel_binding

May 20, 2016

Binding object which returns the resources bound to responderpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the responder policy label.
responderpolicylabel_policybinding_binding	<responderpolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to responderpolicylabel.
responderpolicylabel_responderpolicy_binding	<responderpolicylabel_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to responderpolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "responderpolicylabel_binding": [ {
    "labelname":<String_value>,
    "responderpolicylabel_responderpolicy_binding":<responderpolicylabel_responderpolicy_binding[]_value>,
    "responderpolicylabel_policybinding_binding":<responderpolicylabel_policybinding_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicylabel_policybinding_binding

Jul 07, 2017

Binding object showing the policybinding that can be bound to responderpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the responder policy.
labelname	<String>	Read-write	Name of the responder policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label and evaluate the specified policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Available settings function as follows: * vserver - Invoke an unnamed policy label associated with a virtual server. * policylabel - Invoke a user-defined policy label. Possible values = vserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that

© 1999-2017 Citrix Systems, Inc. All rights reserved.

session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicylabel_policybinding_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderpolicylabel_policybinding_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "responderpolicylabel_policybinding_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"responderpolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicylabel_responderpolicy_binding

May 20, 2016

Binding object showing the responderpolicy that can be bound to responderpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the responder policy.
labelname	<String>	Read-write	Name of the responder policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label and evaluate the specified policy label.
labeltype	<String>	Read-write	Type of policy label to invoke. Available settings function as follows: * vserver - Invoke an unnamed policy label associated with a virtual server. * policylabel - Invoke a user-defined policy label. Possible values = vserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/responderpolicylabel_responderpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"responderpolicylabel_responderpolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicylabel_responderpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderpolicylabel_responderpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicylabel_responderpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicylabel_responderpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"responderpolicylabel_responderpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicy_binding

May 20, 2016

Binding object which returns the resources bound to responderpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the responder policy for which to display settings.
responderpolicy_responderpolicylabel_binding	<responderpolicy_responderpolicylabel_binding[]>	Read-only	responderpolicylabel that can be bound to responderpolicy.
responderpolicy_csvserver_binding	<responderpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to responderpolicy.
responderpolicy_lbvserver_binding	<responderpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to responderpolicy.
responderpolicy_responderglobal_binding	<responderpolicy_responderglobal_binding[]>	Read-only	responderglobal that can be bound to responderpolicy.
responderpolicy_crvserver_binding	<responderpolicy_crvserver_binding[]>	Read-only	crvserver that can be bound to responderpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that

© 1999-2017 Citrix Systems, Inc. All rights reserved.

session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "responderpolicy_binding": [ {
    "name":<String_value>,
    "responderpolicy_crvserver_binding":<responderpolicy_crvserver_binding[]_value>,
    "responderpolicy_lbvserver_binding":<responderpolicy_lbvserver_binding[]_value>,
    "responderpolicy_responderpolicylabel_binding":<responderpolicy_responderpolicylabel_binding[]_value>,
    "responderpolicy_responderglobal_binding":<responderpolicy_responderglobal_binding[]_value>,
    "responderpolicy_csvserver_binding":<responderpolicy_csvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicy_crvserver_binding

May 20, 2016

Binding object showing the crvserver that can be bound to responderpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the responder policy for which to display settings.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_crvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_crvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicy_crvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_crvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderpolicy_crvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicy_crvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_crvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"responderpolicy_crvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicy_csvserver_binding

Jul 07, 2017

Binding object showing the csvserver that can be bound to responderpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the responder policy for which to display settings.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicy_csvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"responderpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to responderpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the responder policy for which to display settings.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicy_lbvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"responderpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicy_responderglobal_binding

May 20, 2016

Binding object showing the responderglobal that can be bound to responderpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the responder policy for which to display settings.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then

© 1999-2017 Citrix Systems, Inc. All rights reserved.

using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_responderglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_responderglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicy_responderglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_responderglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderpolicy_responderglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicy_responderglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_responderglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"responderpolicy_responderglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicy_responderpolicylabel_binding

May 20, 2016

Binding object showing the responderpolicylabel that can be bound to responderpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the responder policy for which to display settings.
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_responderpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_responderpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of responderpolicy_responderpolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_responderpolicylabel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the responderpolicy_responderpolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "responderpolicy_responderpolicylabel_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotoprioityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/responderpolicy_responderpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"responderpolicy_responderpolicylabel_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rewrite

May 20, 2016

Rewrite configuration.

Name	Description
rewriteaction	Configuration for rewrite action
rewriteglobal_binding	Binding object showing the resources that can be bound to rewriteglobal
rewriteglobal_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to rewriteglobal
rewriteparam	Configuration for rewrite parameter
rewritepolicy	Configuration for rewrite policy
rewritepolicy_binding	Binding object showing the resources that can be bound to rewritepolicy
rewritepolicy_csvserver_binding	Binding object showing the csvserver that can be bound to rewritepolicy
rewritepolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to rewritepolicy
rewritepolicy_rewriteglobal_binding	Binding object showing the rewriteglobal that can be bound to rewritepolicy
rewritepolicy_rewritepolicylabel_binding	Binding object showing the rewritepolicylabel that can be bound to rewritepolicy
rewritepolicylabel	Configuration for rewrite policy label
rewritepolicylabel_binding	Binding object showing the resources that can be bound to rewritepolicylabel
rewritepolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to rewritepolicylabel
rewritepolicylabel_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to rewritepolicylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewriteaction

May 20, 2016

Configuration for rewrite action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the user-defined rewrite action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the rewrite policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rewrite action" or ?my rewrite action?).
			Type of user-defined rewrite action. The information that you provide for, and the effect of, each type are as follows:: * REPLACE <target> <string_builder_expr>. Replaces the string with the string-builder expression. * REPLACE_ALL <target> <string_builder_expr1> -(pattern | search) <string_builder_expr2>. In the request or response specified by <target>, replaces all occurrences of the string defined by <string_builder_expr1> with the string defined by <string_builder_expr2>. You can use a PCRE-format pattern or the search facility to find the strings to be replaced. * REPLACE_HTTP_RES <string_builder_expr>. Replaces the complete HTTP response with the string defined by the string-builder expression. * REPLACE_SIP_RES <target> - Replaces the complete SIP response with the string specified by <target>. * INSERT_HTTP_HEADER <header_string_builder_expr> <contents_string_builder_expr>. Inserts the HTTP header specified by <header_string_builder_expr> and header contents specified by <contents_string_builder_expr>. * DELETE_HTTP_HEADER <target>. Deletes the HTTP header specified by <target>. * CORRUPT_HTTP_HEADER <target>. Replaces the header name of all occurrences of the HTTP header specified by <target> with a corrupted name, so that it will not be recognized by the receiver Example: MY_HEADER is changed to MHEY_ADER. * INSERT_BEFORE <string_builder_expr1> <string_builder_expr1>. Finds the string specified in <string_builder_expr1> and inserts the string in <string_builder_expr2> before it. * INSERT_BEFORE_ALL <target> <string_builder_expr1> -(pattern | search) <string_builder_expr2>. In the request or response specified by <target>, locates all occurrences of the string specified in <string_builder_expr1> and inserts the string specified in <string_builder_expr2> before each. You can use a PCRE-format pattern or the search facility to find the strings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			specified in <string_builder_expr1>, and inserts the string specified in <string_builder_expr2> after it. * INSERT_AFTER_ALL <target> <string_builder_expr1> -(pattern | search) <string_builder_expr>. In the request or response specified by <target>, locates all occurrences of the string specified by <string_builder_expr1> and inserts the string specified by <string_builder_expr2> after each. You can use a PCRE-format pattern or the search facility to find the strings. * DELETE <target>. Finds and deletes the specified target. * DELETE_ALL <target> -(pattern | search) <string_builder_expr>. In the request or response specified by <target>, locates and deletes all occurrences of the string specified by <string_builder_expr>. You can use a PCRE-format pattern or the search facility to find the strings. * REPLACE_DIAMETER_HEADER_FIELD <target> <field value>. In the request or response modify the header field specified by <target>. Use Diameter.req.flags.SET(<flag>) or Diameter.req.flags.UNSET<flag> as 'stringbuilderexpression' to set or unset flags. * REPLACE_DNS_HEADER_FIELD <target>. In the request or response modify the header field specified by <target>. * REPLACE_DNS_ANSWER_SECTION <target>. Replace the DNS answer section in the response. This is currently applicable for A and AAAA records only. Use DNS.NEW_RRSET_A & DNS.NEW_RRSET_AAAA expressions to configure the new answer section . Possible values = noop, delete, insert_http_header, delete_http_header, corrupt_http_header, insert_before, insert_after, replace, replace_http_res, delete_all, replace_all, insert_before_all, insert_after_all, clientless_vpn_encode, clientless_vpn_encode_all, clientless_vpn_decode, clientless_vpn_decode_all, insert_sip_header, delete_sip_header, corrupt_sip_header, replace_sip_res, replace_diameter_header_field, replace_dns_header_field, replace_dns_answer_section
target	<String>	Read-write	Default syntax expression that specifies which part of the request or response to rewrite. Minimum length = 1
stringbuilderexpr	<String>	Read-write	Default syntax expression that specifies the content to insert into the request or response at the specified location, or that replaces the specified string.
pattern	<String>	Read-write	Pattern that is used to match multiple strings in the request or response. The pattern may be a string literal (without quotes) or a PCRE-format regular expression with a delimiter that consists of any printable ASCII non-alphanumeric character except for the underscore (_) and space ( ) that is not otherwise used in the expression. Example: re~https?:// | HTTPS?://~ The preceding regular expression can use the tilde (~) as the delimiter because that character does not appear in the regular expression itself. Used in the INSERT_BEFORE_ALL, INSERT_AFTER_ALL, REPLACE_ALL, and DELETE_ALL action types.
			Search facility that is used to match multiple strings in the request or response. Used in the INSERT_BEFORE_ALL, INSERT_AFTER_ALL, REPLACE_ALL, and DELETE_ALL action types. The following search types are supported: * Text ("text(string)") - A literal string. Example: -search text("hello") * Regular expression (?regex(re<delimiter>regular exp<delimiter>)?) - Pattern that is used to match multiple strings in the request or response. The pattern may be a string literal (without quotes) or a PCRE-format regular expression with a delimiter that consists of any printable ASCII non-alphanumeric character except for the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
search	<String>	Read-write	-search regex(re~^hello~) The preceding regular expression can use the tilde (~) as the delimiter because that character does not appear in the regular expression itself. * XPath ("xpath(xp<delimiter>xpath expression<delimiter>)") - An XPath expression. Example: -search xpath(xp%/a/b%) * JSON ("xpath_json(xp<delimiter>xpath expression<delimiter>)") - An XPath JSON expression. Example: -search xpath_json(xp%/a/b%) NOTE: JSON searches use the same syntax as XPath searches, but operate on JSON files instead of standard XML files. * Patset ("patset(patset)") - A predefined pattern set. Example: -search patset("patset1"). * Datset ("dataset(dataset)") - A predefined dataset. Example: -search dataset("dataset1"). * AVP ("avp(avp number)") - AVP number that is used to match multiple AVPs in a Diameter/Radius Message. Example: -search avp(999).
bypasssafetycheck	<String>	Read-write	Bypass the safety check and allow unsafe expressions. An unsafe expression is one that contains references to message elements that might not be present in all messages. If an expression refers to a missing request element, an empty string is used instead. Default value: NO Possible values = YES, NO
refinesearch	<String>	Read-write	Specify additional criteria to refine the results of the search. Always starts with the "extend(m,n)" operation, where 'm' specifies number of bytes to the left of selected data and 'n' specifies number of bytes to the right of selected data. You can use refineSearch only on body expressions, and for the INSERT_BEFORE_ALL, INSERT_AFTER_ALL, REPLACE_ALL, and DELETE_ALL action types.
comment	<String>	Read-write	Comment. Can be used to preserve information about this rewrite action.
newname	<String>	Read-write	New name for the rewrite action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the rewrite policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rewrite action" or ?my rewrite action?). Minimum length = 1
hits	<Double>	Read-only	The number of times the action has been taken.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.
referencecount	<Double>	Read-only	The number of references to the action.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default rewriteaction.
builtin	<String[]>	Read-only	Flag to determine whether rewrite action is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"rewriteaction":{
    "name":<String_value>,
    "type":<String_value>,
    "target":<String_value>,
    "stringbuilderexpr":<String_value>,
    "pattern":<String_value>,
    "search":<String_value>,
    "bypasssafetycheck":<String_value>,
    "refinesearch":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"rewriteaction":{
    "name":<String_value>,
    "target":<String_value>,
    "stringbuilderexpr":<String_value>,
    "bypasssafetycheck":<String_value>,
    "pattern":<String_value>,
    "search":<String_value>,
    "bypasssafetycheck":<String_value>,
    "refinesearch":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rewriteaction":{
    "name":<String_value>,
    "stringbuilderexpr":true,
    "refinesearch":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteaction?**action=rename**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rewriteaction":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rewriteaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/rewriteaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rewriteaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/rewriteaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/rewriteaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rewriteaction resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewriteaction": [ {
      "name":<String_value>,
      "type":<String_value>,
      "target":<String_value>,
      "stringbuilderexpr":<String_value>,
      "pattern":<String_value>,
      "search":<String_value>,
      "bypasssafetycheck":<String_value>,
      "refinesearch":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "referencecount":<Double_value>,
      "description":<String_value>,
      "isdefault":<Boolean_value>,
      "comment":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rewriteaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/rewriteaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewriteaction": [ {
    "name":<String_value>,
    "type":<String_value>,
    "target":<String_value>,
    "stringbuilderexpr":<String_value>,
    "pattern":<String_value>,
    "search":<String_value>,
    "bypasssafetycheck":<String_value>,
    "refinesearch":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "referencecount":<Double_value>,
    "description":<String_value>,
    "isdefault":<Boolean_value>,
    "comment":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "rewriteaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewriteglobal_binding

May 20, 2016

Binding object which returns the resources bound to rewriteglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
rewriteglobal_rewritepolicy_binding	<rewriteglobal_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to rewriteglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "rewriteglobal_binding": [ {
      "rewriteglobal_rewritepolicy_binding":<rewriteglobal_rewritepolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewriteglobal_rewritepolicy_binding

May 20, 2016

Binding object showing the rewritepolicy that can be bound to rewriteglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the rewrite policy.
labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to which to forward the request of response.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
type	<String>	Read-write	The bindpoint to which to policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT, OTHERTCP_REQ_OVERRIDE, OTHERTCP_REQ_DEFAULT, OTHERTCP_RES_OVERRIDE, OTHERTCP_RES_DEFAULT, SIPUDP_REQ_OVERRIDE, SIPUDP_REQ_DEFAULT, SIPUDP_RES_OVERRIDE, SIPUDP_RES_DEFAULT, SIPTCP_REQ_OVERRIDE, SIPTCP_REQ_DEFAULT, SIPTCP_RES_OVERRIDE, SIPTCP_RES_DEFAULT, DIAMETER_REQ_OVERRIDE, DIAMETER_REQ_DEFAULT, DIAMETER_RES_OVERRIDE, DIAMETER_RES_DEFAULT, RADIUS_REQ_OVERRIDE, RADIUS_REQ_DEFAULT, RADIUS_RES_OVERRIDE, RADIUS_RES_DEFAULT, DNS_REQ_OVERRIDE, DNS_REQ_DEFAULT, DNS_RES_OVERRIDE, DNS_RES_DEFAULT
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/rewriteglobal_rewritepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{
"rewriteglobal_rewritepolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteglobal_rewritepolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteglobal_rewritepolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/rewriteglobal_rewritepolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewriteglobal_rewritepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

address>/nitro/v1/config/rewriteglobal_rewritepolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rewriteglobal_rewritepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewriteglobal_rewritepolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "type":<String_value>,
    "labeltype":<String_value>,
    "flowtype":<Double_value>,
    "numpol":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteglobal_rewritepolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "rewriteglobal_rewritepolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewriteparam

May 20, 2016

Configuration for rewrite parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Available settings function as follows: * NOOP - Send the request to the protected server instead of responding to it. * RESET - Reset the request and notify the user?s browser, so that the user can resend the request. * DROP - Drop the request without sending a response to the user. Default value: "NOREWRITE"

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rewriteparam":{
    "undefaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rewriteparam":{
    "undefaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rewriteparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewriteparam": [ {
      "undefaction":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicy

May 20, 2016

Configuration for rewrite policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the rewrite policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the rewrite policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rewrite policy" or ?my rewrite policy?).
rule	<String>	Read-write	Expression against which traffic is evaluated. Written in default syntax. Note: Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' (Classic expressions are not supported in the cluster build.) The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the rewrite action to perform if the request or response matches this rewrite policy. There are also some built-in actions which can be used. These are: * NOREWRITE - Send the request from the client to the server or response from the server to the client without making any changes in the message. * RESET - Resets the client connection by closing it. The client program, such as a browser, will handle this and may inform the user. The client may then resend the request if desired. * DROP - Drop the request without sending a response to the user.
undefaction	<String>	Read-write	Action to perform if the result of policy evaluation is undefined (UNDEF). An UNDEF event indicates an internal error condition. Only the above built-in actions can be used.
comment	<String>	Read-write	Any comments to preserve information about this rewrite policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
logaction	<String>	Read-write	Name of messagelog action to use when a request matches this policy.
newname	<String>	Read-write	New name for the rewrite policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rewrite policy" or ?my rewrite policy?). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
undefhits	<Double>	Read-only	Number of Undef hits.
description	<String>	Read-only	Description of the policy.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default rewritepolicy.
builtin	<String[]>	Read-only	Flag to determine if rewrite policy is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rewritepolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rewritepolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rewritepolicy":{
    "name":<String_value>,
    "undefaction":true,
    "comment":true,
    "logaction":true
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rewritepolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rewritepolicy resources configured on NetScaler.Filtering can

© 1999-2017 Citrix Systems, Inc. All rights reserved.

be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rewritepolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "undefaction":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "description":<String_value>,
      "comment":<String_value>,
      "logaction":<String_value>,
      "isdefault":<Boolean_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "undefaction":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "description":<String_value>,
      "comment":<String_value>,
      "logaction":<String_value>,
      "isdefault":<Boolean_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "rewritepolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicylabel

May 20, 2016

Configuration for rewrite policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name for the rewrite policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) hash (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the rewrite policy label is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my rewrite policy label" or ?my rewrite policy label?).
transform	<String>	Read-write	Types of transformations allowed by the policies bound to the label. For Rewrite, the following types are supported: * http_req - HTTP requests * http_res - HTTP responses * othertcp_req - Non-HTTP TCP requests * othertcp_res - Non-HTTP TCP responses * url - URLs * text - Text strings * clientless_vpn_req - NetScaler clientless VPN requests * clientless_vpn_res - NetScaler clientless VPN responses * sipudp_req - SIP requests * sipudp_res - SIP responses * diameter_req - DIAMETER requests * diameter_res - DIAMETER responses * radius_req - RADIUS requests * radius_res - RADIUS responses * dns_req - DNS requests * dns_res - DNS responses. Possible values = http_req, http_res, othertcp_req, othertcp_res, url, text, clientless_vpn_req, clientless_vpn_res, sipudp_req, sipudp_res, siptcp_req, siptcp_res, diameter_req, diameter_res, radius_req, radius_res, dns_req, dns_res
comment	<String>	Read-write	Any comments to preserve information about this rewrite policy label.
			New name for the rewrite policy label. Must begin with a letter, number, or the underscore character (_), and must

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
newname	<String>	Read-write	(@), equals (=), colon (:), and underscore characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy label" or ?my policy label?). Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
invoke_labelname	<String>	Read-only	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to which to forward the request or response.
flowtype	<Double>	Read-only	Flowtype of the bound rewrite policy.
description	<String>	Read-only	Description of the policylabel.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default rewritepolicylabel.
builtin	<String[]>	Read-only	Flag to determine if rewrite policy label is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rewritepolicylabel":{
    "labelname":<String_value>,
    "transform":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel/labelname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"rewritepolicylabel":{
    "labelname":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of rewritepolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rewritepolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicylabel": [ {
      "labelname":<String_value>,
      "transform":<String_value>,
      "numpol":<Double_value>,
      "hits":<Double_value>,
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "labeltype":<String_value>,
      "invoke_labelname":<String_value>,
      "flowtype":<Double_value>,
      "description":<String_value>,
      "isdefault":<Boolean_value>,
      "comment":<String_value>,
      "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicylabel": [ {
    "labelname":<String_value>,
    "transform":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "flowtype":<Double_value>,
    "description":<String_value>,
    "isdefault":<Boolean_value>,
    "comment":<String_value>,
    "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "rewritepolicylabel": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicylabel_binding

May 20, 2016

Binding object which returns the resources bound to rewritepolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the rewrite policy label.
rewritepolicylabel_policybinding_binding	<rewritepolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to rewritepolicylabel.
rewritepolicylabel_rewritepolicy_binding	<rewritepolicylabel_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to rewritepolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "rewritepolicylabel_binding": [ {
    "labelname":<String_value>,
    "rewritepolicylabel_policybinding_binding":<rewritepolicylabel_policybinding_binding[]_value>,
    "rewritepolicylabel_rewritepolicy_binding":<rewritepolicylabel_rewritepolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicylabel_policybinding_binding

Jul 07, 2017

Binding object showing the policybinding that can be bound to rewritepolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the rewrite policy to bind to the policy label.
labelname	<String>	Read-write	Name of the rewrite policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to which to forward the request or response.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Suspend evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform

© 1999-2017 Citrix Systems, Inc. All rights reserved.

individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicylabel_policybinding_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rewritepolicylabel_policybinding_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicylabel_policybinding_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"rewritepolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicylabel_rewritepolicy_binding

May 20, 2016

Binding object showing the rewritepolicy that can be bound to rewritepolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
policyname	<String>	Read-write	Name of the rewrite policy to bind to the policy label.
labelname	<String>	Read-write	Name of the rewrite policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	* If labelType is policylabel, name of the policy label to invoke. * If labelType is reqvserver or resvserver, name of the virtual server to which to forward the request or response.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Suspend evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * resvserver - Forward the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform

© 1999-2017 Citrix Systems, Inc. All rights reserved.

individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"rewritepolicylabel_rewritepolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicylabel_rewritepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rewritepolicylabel_rewritepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicylabel_rewritepolicy_binding": [ {
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "policyname":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicylabel_rewritepolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"rewritepolicylabel_rewritepolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicy_binding

May 20, 2016

Binding object which returns the resources bound to rewritepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the rewrite policy.
rewritepolicy_csvserver_binding	<rewritepolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to rewritepolicy.
rewritepolicy_lbvserver_binding	<rewritepolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to rewritepolicy.
rewritepolicy_rewriteglobal_binding	<rewritepolicy_rewriteglobal_binding[]>	Read-only	rewriteglobal that can be bound to rewritepolicy.
rewritepolicy_rewritepolicylabel_binding	<rewritepolicy_rewritepolicylabel_binding[]>	Read-only	rewritepolicylabel that can be bound to rewritepolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "rewritepolicy_binding": [ {
    "name":<String_value>,
    "rewritepolicy_lbvserver_binding":<rewritepolicy_lbvserver_binding[]_value>,
    "rewritepolicy_rewritepolicylabel_binding":<rewritepolicy_rewritepolicylabel_binding[]_value>,
    "rewritepolicy_rewriteglobal_binding":<rewritepolicy_rewriteglobal_binding[]_value>,
    "rewritepolicy_csvserver_binding":<rewritepolicy_csvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to rewritepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the rewrite policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rewritepolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicy_csvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"rewritepolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicy_lbvserver_binding

Jul 07, 2017

Binding object showing the lbvserver that can be bound to rewritepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the rewrite policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rewritepolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicy_lbvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"rewritepolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicy_rewriteglobal_binding

May 20, 2016

Binding object showing the rewriteglobal that can be bound to rewritepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the rewrite policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_rewriteglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_rewriteglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicy_rewriteglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_rewriteglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rewritepolicy_rewriteglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicy_rewriteglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_rewriteglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"rewritepolicy_rewriteglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicy_rewritepolicylabel_binding

May 20, 2016

Binding object showing the rewritepolicylabel that can be bound to rewritepolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the rewrite policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_rewritepolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_rewritepolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of rewritepolicy_rewritepolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_rewritepolicylabel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the rewritepolicy_rewritepolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicy_rewritepolicylabel_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/rewritepolicy_rewritepolicylabel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"rewritepolicy_rewritepolicylabel_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rise

May 20, 2016

RISE feature related configuration.

Name	Description
riseapbrsvc	Configuration for RISE Apbr services
riseparam	Configuration for RISE Parameter
riseprofile	Configuration for RISE profile
riseprofile_binding	Binding object showing the resources that can be bound to riseprofile
riseprofile_interface_binding	Binding object showing the interface that can be bound to riseprofile
riserhi	Configuration for RISE RHI rules

© 1999-2017 Citrix Systems, Inc. All rights reserved.

riseapbrsvc

May 20, 2016

Configuration for RISE Apbr services resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-only	Name for the APBR service.
risesvctype	<String>	Read-only	Service or Service Group. Possible values = Service, ServiceGroup
serverip	<String>	Read-only	Server IP for APBR service.
nexthopip	<String>	Read-only	Nexthop IP for APBR service.
vlan	<Double>	Read-only	Vlan id for APBR service.
protocol	<String>	Read-only	Protocol type for APBR service. Possible values = HTTP, FTP, TCP, UDP, SSL, SSL_BRIDGE, SSL_TCP, DTLS, NNTP, RPCSVR, DNS, ADNS, SNMP, RTSP, DHCPRA, ANY, SIP_UDP, SIP_TCP, SIP_SSL, DNS_TCP, ADNS_TCP, MYSQL, MSSQL, ORACLE, RADIUS, RADIUSListener, RDP, DIAMETER, SSL_DIAMETER, TFTP, SMPP, PPTP, GRE, SYSLOGTCP, SYSLOGUDP, FIX
serverport	<Integer>	Read-only	Server port for APBR service. Range 1 - 65535
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/riseapbrsvc

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/riseapbrsvc?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/riseapbrsvc?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of riseapbrsvc resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/riseapbrsvc?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/riseapbrsvc?**pagesize=#no&pageno=#no**

Use this query-parameter to get the riseapbrsvc resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "riseapbrsvc": [ {
      "name":<String_value>,
      "risesvctype":<String_value>,
      "serverip":<String_value>,
      "nexthopip":<String_value>,
      "vlan":<Double_value>,
      "protocol":<String_value>,
      "serverport":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/riseapbrsvc?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "riseapbrsvc": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

riseparam

May 20, 2016

Configuration for RISE Parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
directmode	<String>	Read-write	RISE Direct attach mode. Default value: ENABLED Possible values = ENABLED, DISABLED
indirectmode	<String>	Read-write	RISE Indirect attach mode. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/riseparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"riseparam":{
    "directmode":<String_value>,
    "indirectmode":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/riseparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"riseparam":{
    "directmode":true,
    "indirectmode":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/riseparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "riseparam": [ {
      "directmode":<String_value>,
      "indirectmode":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

riseprofile

May 20, 2016

Configuration for RISE profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
profilename	<String>	Read-write	Name of the RISE profile. Minimum length = 1 Maximum length = 83
servicename	<String>	Read-only	RISE Service name. Minimum length = 1 Maximum length = 31
deviceid	<String>	Read-only	Device id. Minimum length = 1 Maximum length = 31
slotid	<Double>	Read-only	Slot number of the RISE profile.
slotno	<Double>	Read-only	Slot number of the RISE profile.
vdcid	<Double>	Read-only	RISE vdc id.
vpcid	<Double>	Read-only	RISE vpc id.
ipaddress	<String>	Read-only	RISE ip address.
mode	<String>	Read-only	RISE attach mode. Possible values = Direct, Indirect, vPC-Direct
status	<String>	Read-only	RISE status. Possible values = Active, Inactive
vlan	<Double>	Read-only	RISE Vlan id.
vlangroupid	<Double>	Read-only	RISE Vlan Group id.
vlancfgstatus	<String>	Read-only	RISE config status.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ifnum	<String>	Read-only	RISE Interface number.
issu	<String>	Read-only	RISE issu status. Possible values = InProgress, None
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/riseprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/riseprofile?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/riseprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of riseprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/riseprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/riseprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the riseprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "riseprofile": [ {
    "profilename":<String_value>,
    "servicename":<String_value>,
    "deviceid":<String_value>,
    "slotid":<Double_value>,
    "slotno":<Double_value>,
    "vdcid":<Double_value>,
    "vpcid":<Double_value>,
    "ipaddress":<String_value>,
    "mode":<String_value>,
    "status":<String_value>,
    "vlan":<Double_value>,
    "vlangroupid":<Double_value>,
    "vlancfgstatus":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ifnum":<String_value>,
    "issu":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/riseprofile/profilename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/riseprofile/profilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/riseprofile/profilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "riseprofile": [ {
    "profilename":<String_value>,
    "servicename":<String_value>,
    "deviceid":<String_value>,
    "slotid":<Double_value>,
    "slotno":<Double_value>,
    "vdcid":<Double_value>,
    "vpcid":<Double_value>,
    "ipaddress":<String_value>,
    "mode":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "status":<String_value>,
    "vlan":<Double_value>,
    "vlangroupid":<Double_value>,
    "vlancfgstatus":<String_value>,
    "ifnum":<String_value>,
    "issu":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/riseprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "riseprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

riseprofile_binding

May 20, 2016

Binding object which returns the resources bound to riseprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
profilename	<String>	Read-write	Name of the RISE profile. Minimum length = 1 Maximum length = 83
riseprofile_interface_binding	<riseprofile_interface_binding[]>	Read-only	interface that can be bound to riseprofile.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/riseprofile_binding/profilename_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "riseprofile_binding": [ {
      "profilename":<String_value>,
      "riseprofile_interface_binding":<riseprofile_interface_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

riseprofile_interface_binding

May 20, 2016

Binding object showing the interface that can be bound to riseprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
memberinterface	<String>	Read-write	RISE profile member interfaces.
profilename	<String>	Read-write	Name of the RISE profile. Minimum length = 1 Maximum length = 83
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/riseprofile_interface_binding/profilename_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/riseprofile_interface_binding/profilename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of riseprofile_interface_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/riseprofile_interface_binding/profilename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the riseprofile_interface_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "riseprofile_interface_binding": [ {
      "memberinterface":<String_value>,
      "profilename":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/riseprofile_interface_binding/profilename_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"riseprofile_interface_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

riserhi

Jul 07, 2017

Configuration for RISE RHI rules resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipaddress	<String>	Read-only	(V)IP advertised.
prefixlen	<Double>	Read-only	Prefix Length.
hostrtgw	<String>	Read-only	Gateway for the advertised IP.
nexthopvlan	<Double>	Read-only	Vlan id on which the gateway is present.
weight	<Double>	Read-only	Cost of the route.
vserverrhilevel	<String>	Read-only	Advertisement level. Default value: ONE_VSERVER Possible values = ONE_VSERVER, ALL_VSERVERS, NONE, VSVR_CNTRLD
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/riserhi

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/riserhi?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/riserhi?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of riserhi resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/riserhi?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/riserhi?**pagesize=#no&pageno=#no**

Use this query-parameter to get the riserhi resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "riserhi": [ {
    "ipaddress":<String_value>,
    "prefixlen":<Double_value>,
    "hostrtgw":<String_value>,
    "nexthopvlan":<Double_value>,
    "weight":<Double_value>,
    "vserverrhilevel":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/riserhi?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "riserhi": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Router

May 20, 2016

Router configuration.

Name	Description
routerdynamicrouting	Configuration for dynamic routing config

© 1999-2017 Citrix Systems, Inc. All rights reserved.

routerdynamicrouting

Jul 07, 2017

Configuration for dynamic routing config resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
commandstring	<String>	Read-write	command to be executed.
output	<String>	Read-only	command output.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

APPLY | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

apply

URL: http://<netscaler-ip-address>/nitro/v1/config/routerdynamicrouting?**action=apply**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"routerdynamicrouting":{
    "commandstring":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/routerdynamicrouting

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/routerdynamicrouting?**args=commandstring:<String_value>**

Use this query-parameter to get routerdynamicrouting resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/routerdynamicrouting?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/routerdynamicrouting?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of routerdynamicrouting resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/routerdynamicrouting?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/routerdynamicrouting?**pagesize=#no&pageno=#no**

Use this query-parameter to get the routerdynamicrouting resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "routerdynamicrouting": [ {
commandstring:<String_value>      "output":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/routerdynamicrouting?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "routerdynamicrouting": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Sure Connect

May 20, 2016

SureConnect configuration. SureConnect allows the system to deliver alternate content to clients when the system is under heavy load for the original URL. The original content will be served to the client, automatically, after an estimated delay period.

Name	Description
scparameter	Configuration for SC parameter
scpolicy	Configuration for SureConnect policy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

scparameter

May 20, 2016

Configuration for SC parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sessionlife	<Double>	Read-write	Time, in seconds, between the first time and the next time the SureConnect alternative content window is displayed. The alternative content window is displayed only once during a session for the same browser accessing a configured URL, so this parameter determines the length of a session. Default value: 300 Minimum value = 1 Maximum value = 4294967294
vsr	<String>	Read-write	File containing the customized response to be displayed when the ACTION in the SureConnect policy is set to NS. Default value: "DEFAULT" Minimum length = 1

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/scparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"scparameter":{
    "sessionlife":<Double_value>,
    "vsr":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/scparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"scparameter":{
    "sessionlife":true,
    "vsr":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/scparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "scparameter": [ {
      "sessionlife":<Double_value>,
      "vsr":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

scpolicy

May 20, 2016

Configuration for SureConnect policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Minimum length = 1 Maximum length = 31
url	<String>	Read-write	URL against which to match incoming client request. Maximum length = 127
rule	<String>	Read-write	Expression against which the traffic is evaluated. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks. Maximum length = 1499
delay	<Double>	Read-write	Delay threshold, in microseconds, for requests that match the policy's URL or rule. If the delay statistics gathered for the matching request exceed the specified delay, SureConnect is triggered for that request. Minimum value = 1 Maximum value = 599999999
maxconn	<Double>	Read-write	Maximum number of concurrent connections that can be open for requests that match the policy's URL or rule. Minimum value = 1 Maximum value = 4294967294
			Action to be taken when the delay or maximum-connections threshold is reached. Available settings function as follows: ACS - Serve content from an alternative content service. NS - Serve alternative content from the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			collected for the configured URLs, and, if the Maximum Client Connections parameter is set, the number of connections is limited to the value specified by that parameter. (However, alternative content is not served even if the maxConn threshold is met). Possible values = ACS, NS, NOACTION
altcontentsvcname	<String>	Read-write	Name of the alternative content service to be used in the ACS action. Minimum length = 1 Maximum length = 127
altcontentpath	<String>	Read-write	Path to the alternative content service to be used in the ACS action. Minimum length = 1 Maximum length = 127
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/scpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"scpolicy":{
    "name":<String_value>,
    "url":<String_value>,
    "rule":<String_value>,
    "delay":<Double_value>,
    "maxconn":<Double_value>,
    "action":<String_value>,
    "altcontentsvcname":<String_value>,
    "altcontentpath":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/scpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/scpolicy

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"scpolicy":{
    "name":<String_value>,
    "url":<String_value>,
    "rule":<String_value>,
    "delay":<Double_value>,
    "maxconn":<Double_value>,
    "action":<String_value>,
    "altcontentsvcname":<String_value>,
    "altcontentpath":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/scpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"scpolicy":{
    "name":<String_value>,
    "delay":true,
    "maxconn":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/scpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/scpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/scpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of scpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/scpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/scpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the scpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "scpolicy": [ {
      "name":<String_value>,
      "url":<String_value>,
      "rule":<String_value>,
      "delay":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "maxconn":<Double_value>,
    "action":<String_value>,
    "altcontentsvcname":<String_value>,
    "altcontentpath":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/scpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/scpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/scpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "scpolicy": [ {
    "name":<String_value>,
    "url":<String_value>,
    "rule":<String_value>,
    "delay":<Double_value>,
    "maxconn":<Double_value>,
    "action":<String_value>,
    "altcontentsvcname":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "altcontentpath":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/scpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "scpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Smpp

May 20, 2016

All the commands associated with SMPP.

Name	Description
smppparam	Configuration for SMPP configuration parameters
smppuser	Configuration for SMPP user

© 1999-2017 Citrix Systems, Inc. All rights reserved.

smppparam

Jul 07, 2017

Configuration for SMPP configuration parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clientmode	<String>	Read-write	Mode in which the client binds to the ADC. Applicable settings function as follows: * TRANSCEIVER - Client can send and receive messages to and from the message center. * TRANSMITTERONLY - Client can only send messages. * RECEIVERONLY - Client can only receive messages. Default value: TRANSCEIVER Possible values = TRANSCEIVER, TRANSMITTERONLY, RECEIVERONLY
msgqueue	<String>	Read-write	Queue SMPP messages if a client that is capable of receiving the destination address messages is not available. Default value: OFF Possible values = ON, OFF
msgqueuesize	<Double>	Read-write	Maximum number of SMPP messages that can be queued. After the limit is reached, the NetScaler ADC sends a deliver_sm_resp PDU, with an appropriate error message, to the message center. Default value: 10000
addrton	<Double>	Read-write	Type of Number, such as an international number or a national number, used in the ESME address sent in the bind request. Default value: 0 Minimum value = 0 Maximum value = 256
addrnpi	<Double>	Read-write	Numbering Plan Indicator, such as landline, data, or WAP client, used in the ESME address sent in the bind request. Default value: 0 Minimum value = 0 Maximum value = 256
addrrange	<String>	Read-write	Set of SME addresses, sent in the bind request, serviced by the ESME. Default value: "\\d*"

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/smppparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"smppparam":{
    "clientmode":<String_value>,
    "msgqueue":<String_value>,
    "msgqueuesize":<Double_value>,
    "addrton":<Double_value>,
    "addrnpi":<Double_value>,
    "addrrange":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/smppparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"smppparam":{
    "clientmode":true,
    "msgqueue":true,
    "msgqueuesize":true,
    "addrton":true,
    "addrnpi":true,
    "addrrange":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/smppparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "smppparam": [ {
    "clientmode":<String_value>,
    "msgqueue":<String_value>,
    "msgqueuesize":<Double_value>,
    "addrton":<Double_value>,
    "addrnpi":<Double_value>,
    "addrrange":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

smppuser

May 20, 2016

Configuration for SMPP user resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	Name of the SMPP user. Must be the same as the user name specified in the SMPP server. Minimum length = 1
password	<String>	Read-write	Password for binding to the SMPP server. Must be the same as the password specified in the SMPP server. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/smppuser

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"smppuser":{
    "username":<String_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/smppuser/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/smppuser

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"smppuser":{
    "username":<String_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/smppuser

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/smppuser?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/smppuser?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of smppuser resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/smppuser?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/smppuser?**pagesize=#no&pageno=#no**

Use this query-parameter to get the smppuser resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "smppuser": [ {
    "username":<String_value>,

}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/smppuser/username_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/smppuser/username_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/smppuser/username_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "smppuser": [ {
    "username":<String_value>,

}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/smppuser?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "smppuser": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SNMP

May 20, 2016

SNMP(Simple Network Management Protocol) configuration.

Name	Description
snmpalarm	Configuration for alarm
snmpcommunity	Configuration for community
snmpengineid	Configuration for SNMP engine id
snmpgroup	Configuration for SNMP group
snmpmanager	Configuration for manager
snmpmib	Configuration for SNMP mib
snmpoid	Configuration for SNMP Object Identifier
snmpoption	Configuration for SNMP option
snmptrap	Configuration for snmp trap
snmptrap_binding	Binding object showing the resources that can be bound to snmptrap
snmptrap_snmpuser_binding	Binding object showing the snmpuser that can be bound to snmptrap
snmpuser	Configuration for SNMP user
snmpview	Configuration for view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmpalarm

May 20, 2016

Configuration for alarm resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
trapname	<String>	Read-write	Name of the SNMP alarm. This parameter is required for identifying the SNMP alarm and cannot be modified. Possible values = CPU-USAGE, AVERAGE-CPU, MEMORY, MGMT-CPU-USAGE, SYNFLOOD, VSERVER-REQRATE, SERVICE-REQRATE, ENTITY-RXRATE, ENTITY-TXRATE, ENTITY-SYNFLOOD, SERVICE-MAXCLIENTS, HA-STATE-CHANGE, ENTITY-STATE, CONFIG-CHANGE, CONFIG-SAVE, SERVICEGROUP-MEMBER-REQRATE, SERVICEGROUP-MEMBER-MAXCLIENTS, MONITOR-RTO-THRESHOLD, LOGIN-FAILURE, SSL-CERT-EXPIRY, FAN-SPEED-LOW, VOLTAGE-LOW, VOLTAGE-HIGH, TEMPERATURE-HIGH, CPU-TEMPERATURE-HIGH, POWER-SUPPLY-FAILURE, DISK-USAGE-HIGH, INTERFACE-THROUGHPUT-LOW, MON_PROBE_FAILED, HA-VERSION-MISMATCH, HA-SYNC-FAILURE, HA-NO-HEARTBEATS, HA-BAD-SECONDARY-STATE, INTERFACE-BW-USAGE, RATE-LIMIT-THRESHOLD-EXCEEDED, ENTITY-NAME-CHANGE, HA-PROP-FAILURE, IP-CONFLICT, PF-RL-RATE-THRESHOLD, PF-RL-PPS-THRESHOLD, PF-RL-RATE-PKTS-DROPPED, PF-RL-PPS-PKTS-DROPPED, APPFW-START-URL, APPFW-DENY-URL, APPFW-VIOLATIONS-TYPE, APPFW-REFERER-HEADER, APPFW-CSRF-TAG, APPFW-COOKIE, APPFW-FIELD-CONSISTENCY, APPFW-BUFFER-OVERFLOW, APPFW-FIELD-FORMAT, APPFW-SAFE-COMMERCE, APPFW-SAFE-OBJECT, APPFW-POLICY-HIT, APPFW-XSS, APPFW-XML-XSS, APPFW-SQL, APPFW-XML-SQL, APPFW-XML-ATTACHMENT, APPFW-XML-DOS, APPFW-XML-VALIDATION, APPFW-XML-WSI, APPFW-XML-SCHEMA-COMPILE, APPFW-XML-SOAP-FAULT, DNSKEY-EXPIRY, HA-LICENSE-MISMATCH, SSL-CARD-FAILED, SSL-CARD-NORMAL, WARM-RESTART-EVENT, HARD-DISK-DRIVE-ERRORS, COMPACT-FLASH-ERRORS, CALLHOME-UPLOAD-EVENT, 1024KEY-EXCHANGE-RATE, 2048KEY-EXCHANGE-RATE, 4096KEY-EXCHANGE-RATE, SSL-CUR-SESSION-INUSE, CLUSTER-NODE-HEALTH, CLUSTER-NODE-QUORUM, CLUSTER-VERSION-MISMATCH, CLUSTER-CCO-CHANGE, CLUSTER-OVS-CHANGE, CLUSTER-SYNC-FAILURE, CLUSTER-PROP-FAILURE, HA-STICKY-PRIMARY, INBAND-PROTOCOL-VERSION-MISMATCH, SSL-CHIP-REINIT, VRID-STATE-CHANGE, PORT-ALLOC-FAILED, LLDP-REMOTE-CHANGE, DUPLICATE-IPV6, PARTITION-CONFIG-EVENT, PARTITION-SWITCHED, LSN-PORTALLOC-FAILED, LSN-PORTQUOTA-EXCEED, LSN-SESSIONQUOTA-EXCEED, VSERVER-SPILLOVER, PARTITION-RATE-LIMIT, CLUSTER-BACKPLANE-HB-MISSING
thresholdvalue	<Double>	Read-write	Value for the high threshold. The NetScaler appliance generates an SNMP trap message when the value of the attribute associated with the alarm is greater than or equal to the specified high threshold value. Minimum value = 1
normalvalue	<Double>	Read-write	Value for the normal threshold. A trap message is generated if the value of the respective attribute falls to or below this value after exceeding the high threshold. Minimum value = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
time	<Double>	Read-write	when the conditions specified in the SNMP alarm are met.Can be specified for the following alarms: SYNFLOOD, HA-VERSION-MISMATCH, HA-SYNC-FAILURE, HA-NO-HEARTBEATS,HA-BAD-SECONDARY-STATE, CLUSTER-NODE-HEALTH, CLUSTER-NODE-QUORUM, CLUSTER-VERSION-MISMATCH, CLUSTER-BKHB-FAILED, PORT-ALLOC-FAILED, COMPACT-FLASH-ERRORS, HARD-DISK-DRIVE-ERRORS and APPFW traps. Default trap time intervals: SYNFLOOD and APPFW traps = 1sec, PORT-ALLOC-FAILED = 3600sec(1 hour), Other Traps = 86400sec(1 day). Default value: 1
state	<String>	Read-write	Current state of the SNMP alarm. The NetScaler appliance generates trap messages only for SNMP alarms that are enabled. Some alarms are enabled by default, but you can disable them. Default value: ENABLED Possible values = ENABLED, DISABLED
severity	<String>	Read-write	Severity level assigned to trap messages generated by this alarm. The severity levels are, in increasing order of severity, Informational, Warning, Minor, Major, and Critical. This parameter is useful when you want the NetScaler appliance to send trap messages to a trap listener on the basis of severity level. Trap messages with a severity level lower than the specified level (in the trap listener entry) are not sent. Default value: Unknown Possible values = Critical, Major, Minor, Warning, Informational
logging	<String>	Read-write	Logging status of the alarm. When logging is enabled, the NetScaler appliance logs every trap message that is generated for this alarm. Default value: ENABLED Possible values = ENABLED, DISABLED
timeout	<Double>	Read-only	If DB is enabled and clear config is fired, then to reset timeinterval of alarm, corresponding default time value is needed. This hidden argument holds the default time value for the corresponding alarm.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

UPDATE | UNSET | ENABLE | DISABLE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpalarm

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpalarm":{
    "trapname":<String_value>,
    "thresholdvalue":<Double_value>,
    "normalvalue":<Double_value>,
    "time":<Double_value>,
    "state":<String_value>,
    "severity":<String_value>,
    "logging":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpalarm?**action=unset**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpalarm":{
    "trapname":<String_value>,
    "thresholdvalue":true,
    "normalvalue":true,
    "time":true,
    "state":true,
    "severity":true,
    "logging":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpalarm?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpalarm":{
    "trapname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpalarm?**action=disable**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpalarm":{
    "trapname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpalarm

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpalarm?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/snmpalarm?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpalarm resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpalarm?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/snmpalarm?**pagesize=#no&pageno=#no**

Use this query-parameter to get the snmpalarm resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpalarm": [ {
      "trapname":<String_value>,
      "thresholdvalue":<Double_value>,
      "normalvalue":<Double_value>,
      "time":<Double_value>,
      "state":<String_value>,
      "severity":<String_value>,
      "logging":<String_value>,
      "timeout":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpalarm/trapname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpalarm/trapname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpalarm/trapname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpalarm": [ {
      "trapname":<String_value>,
      "thresholdvalue":<Double_value>,
      "normalvalue":<Double_value>,
      "time":<Double_value>,
      "state":<String_value>,
      "severity":<String_value>,
      "logging":<String_value>,
      "timeout":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpalarm?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpalarm": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmpcommunity

May 20, 2016

Configuration for community resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
communityname	<String>	Read-write	The SNMP community string. Can consist of 1 to 31 characters that include uppercase and lowercase letters,numbers and special characters. The following requirement applies only to the NetScaler CLI: If the string includes one or more spaces, enclose the name in double or single quotation marks (for example, "my string" or 'my string'). Minimum length = 1
permissions	<String>	Read-write	The SNMP V1 or V2 query-type privilege that you want to associate with this SNMP community. Possible values = GET, GET_NEXT, GET_BULK, SET, ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpcommunity":{
    "communityname":<String_value>,
    "permissions":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity/communityname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpcommunity resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity?**pagesize=#no&pageno=#no**

Use this query-parameter to get the snmpcommunity resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpcommunity": [ {
    "communityname":<String_value>,
    "permissions":<String_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity/communityname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity/communityname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity/communityname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpcommunity": [ {
    "communityname":<String_value>,
    "permissions":<String_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpcommunity?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "snmpcommunity": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmpengineid

May 20, 2016

Configuration for SNMP engine id resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
engineid	<String>	Read-write	A hexadecimal value of at least 10 characters, uniquely identifying the engineid. Minimum length = 10 Maximum length = 31
ownernode	<Double>	Read-write	ID of the cluster node for which you are setting the engineid. Default value: -1 Minimum value = 0 Maximum value = 31
defaultengineid	<String>	Read-only	Unique identifier to assign to the SNMPv3 engine. Should be a hexadecimal value with a minimum length of 10 hex characters.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpengineid

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpengineid":{
    "engineid":<String_value>,
    "ownernode":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpengineid?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpengineid":{
    "ownernode":<Double_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpengineid

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpengineid?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/snmpengineid?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpengineid resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpengineid?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/snmpengineid?**pagesize=#no&pageno=#no**

Use this query-parameter to get the snmpengineid resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpengineid": [ {
    "ownernode":<Double_value>,
    "engineid":<String_value>,
    "defaultengineid":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpengineid/ownernode_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpengineid/ownernode_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpengineid/ownernode_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpengineid": [ {
    "ownernode":<Double_value>,
    "engineid":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "defaultengineid":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpengineid?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpengineid": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmpgroup

May 20, 2016

Configuration for SNMP group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the SNMPv3 group. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the SNMPv3 group. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose it in double or single quotation marks (for example, "my name" or 'my name'). Minimum length = 1
securitylevel	<String>	Read-write	Security level required for communication between the NetScaler appliance and the SNMPv3 users who belong to the group. Specify one of the following options: noAuthNoPriv. Require neither authentication nor encryption. authNoPriv. Require authentication but no encryption. authPriv. Require authentication and encryption. Note: If you specify authentication, you must specify an encryption algorithm when you assign an SNMPv3 user to the group. If you also specify encryption, you must assign both an authentication and an encryption algorithm for each group member. Possible values = noAuthNoPriv, authNoPriv, authPriv
readviewname	<String>	Read-write	Name of the configured SNMPv3 view that you want to bind to this SNMPv3 group. An SNMPv3 user bound to this group can access the subtrees that are bound to this SNMPv3 view as type INCLUDED, but cannot access the ones that are type EXCLUDED. If the NetScaler appliance has multiple SNMPv3 view entries with the same name, all such entries are associated with the SNMPv3 group. Minimum length = 1
storagetype	<String>	Read-only	The storage type for this group. Possible values = volatile, nonVolatile
status	<String>	Read-only	The status of this group. Possible values = active
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD | DELETE | UPDATE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpgroup

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpgroup":{
    "name":<String_value>,
    "securitylevel":<String_value>,
    "readviewname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpgroup/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpgroup

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpgroup":{
    "name":<String_value>,
    "securitylevel":<String_value>,
    "readviewname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpgroup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpgroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/snmpgroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpgroup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpgroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/snmpgroup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the snmpgroup resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpgroup": [ {
      "name":<String_value>,
      "securitylevel":<String_value>,
      "readviewname":<String_value>,
      "storagetype":<String_value>,
      "status":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpgroup?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpgroup": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmpmanager

Jul 07, 2017

Configuration for manager resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ipaddress	<String>	Read-write	IP address of the SNMP manager. Can be an IPv4 or IPv6 address. You can instead specify an IPv4 network address or IPv6 network prefix if you want the NetScaler appliance to respond to SNMP queries from any device on the specified network. Alternatively, instead of an IPv4 address, you can specify a host name that has been assigned to an SNMP manager. If you do so, you must add a DNS name server that resolves the host name of the SNMP manager to its IP address. Note: The NetScaler appliance does not support host names for SNMP managers that have IPv6 addresses. Minimum length = 1 Maximum length = 255
netmask	<String>	Read-write	Subnet mask associated with an IPv4 network address. If the IP address specifies the address or host name of a specific host, accept the default value of 255.255.255.255.
domainresolveretry	<Integer>	Read-write	Amount of time, in seconds, for which the NetScaler appliance waits before sending another DNS query to resolve the host name of the SNMP manager if the last query failed. This parameter is valid for host-name based SNMP managers only. After a query succeeds, the TTL determines the wait time. Minimum value = 5 Maximum value = 20939
ip	<String>	Read-only	The resolved IP address of the hostname manager.
domain	<String>	Read-only	IP address of manager. It will be zero for hostname manager.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpmanager

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpmanager":{
    "ipaddress":<String_value>,
    "netmask":<String_value>,
    "domainresolveretry":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpmanager/ipaddress_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpmanager

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpmanager":{
    "ipaddress":<String_value>,
    "netmask":<String_value>,
    "domainresolveretry":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpmanager?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"snmpmanager":{
    "ipaddress":<String_value>,
    "netmask":<String_value>,
    "domainresolveretry":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpmanager

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpmanager?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/snmpmanager?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpmanager resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpmanager?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/snmpmanager?**pagesize=#no&pageno=#no**

Use this query-parameter to get the snmpmanager resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpmanager": [ {
      "ipaddress":<String_value>,
      "netmask":<String_value>,
      "ip":<String_value>,
      "domain":<String_value>,
      "domainresolveretry":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpmanager?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpmanager": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmpmib

Jul 07, 2017

Configuration for SNMP mib resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
contact	<String>	Read-write	Name of the administrator for this NetScaler appliance. Along with the name, you can include information on how to contact this person, such as a phone number or an email address. Can consist of 1 to 127 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. The following requirement applies only to the NetScaler CLI: If the information includes one or more spaces, enclose it in double or single quotation marks (for example, "my contact" or 'my contact'). Default value: "WebMaster (default)" Minimum length = 1
name	<String>	Read-write	Name for this NetScaler appliance. Can consist of 1 to 127 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the NetScaler appliance. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose it in double or single quotation marks (for example, "my name" or 'my name'). Default value: "NetScaler" Minimum length = 1
location	<String>	Read-write	Physical location of the NetScaler appliance. For example, you can specify building name, lab number, and rack number. Can consist of 1 to 127 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. The following requirement applies only to the NetScaler CLI: If the location includes one or more spaces, enclose it in double or single quotation marks (for example, "my location" or 'my location'). Default value: "POP (default)" Minimum length = 1
			Custom identification number for the NetScaler appliance. Can consist of 1 to 127 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a custom identification that helps identify the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			If the ID includes one or more spaces, enclose it in double or single quotation marks (for example, "my ID" or 'my ID'). Default value: "Default" Minimum length = 1
sysdesc	<String>	Read-only	The description of the system.
sysuptime	<Double>	Read-only	The UP time of the system in 100th of a second.
sysservices	<Double>	Read-only	The services offered by the system.
sysoid	<String>	Read-only	The OID of the system's management system.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpmib

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpmib":{
    "contact":<String_value>,
    "name":<String_value>,
    "location":<String_value>,
    "customid":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpmib?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpmib":{
    "contact":true,
    "name":true,
    "location":true,
    "customid":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpmib

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpmib": [ {
      "contact":<String_value>,
      "name":<String_value>,
      "location":<String_value>,
      "sysdesc":<String_value>,
      "sysuptime":<Double_value>,
      "sysservices":<Double_value>,
      "sysoid":<String_value>,
      "customid":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmpoid

Jul 07, 2017

Configuration for SNMP Object Identifier resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
entitytype	<String>	Read-write	The type of entity whose SNMP OIDs you want to displayType of entity whose SNMP OIDs you want the NetScaler appliance to display. Possible values = VSERVER, SERVICE, SERVICEGROUP
name	<String>	Read-write	Name of the entity whose SNMP OID you want the NetScaler appliance to display. Minimum length = 1
Snmpoid	<String>	Read-only	The snmp oid.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpoid

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/snmpoid?**args=entitytype:<String_value>,name:<String_value>**

Use this query-parameter to get snmpoid resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpoid?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/snmpoid?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpoid resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpoid?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/snmpoid?**pagesize=#no&pageno=#no**

Use this query-parameter to get the snmpoid resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpoid": [ {
entitytype:<String_value>,name:<String_value>      "Snmpoid":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpoid?**args=entitytype:<String_value>,name:<String_value>&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpoid": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmpoption

Jul 07, 2017

Configuration for SNMP option resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
snmpset	<String>	Read-write	Accept SNMP SET requests sent to the NetScaler appliance, and allow SNMP managers to write values to MIB objects that are configured for write access. Default value: DISABLED Possible values = ENABLED, DISABLED
snmptraplogging	<String>	Read-write	Log any SNMP trap events (for SNMP alarms in which logging is enabled) even if no trap listeners are configured. With the default setting, SNMP trap events are logged if at least one trap listener is configured on the appliance. Default value: DISABLED Possible values = ENABLED, DISABLED
partitionnameintrap	<String>	Read-write	Send partition name as a varbind in traps. By default the partition names are not sent as a varbind. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpoption

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpoption":{
    "snmpset":<String_value>,
    "snmptraplogging":<String_value>,
    "partitionnameintrap":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpoption?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpoption":{
    "snmpset":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "snmptraplogging":true,
    "partitionnameintrap":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpoption

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpoption": [ {
    "snmpset":<String_value>,
    "snmptraplogging":<String_value>,
    "partitionnameintrap":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmptrap

May 20, 2016

Configuration for snmp trap resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
trapclass	<String>	Read-write	Type of trap messages that the NetScaler appliance sends to the trap listener: Generic or the enterprise-specific messages defined in the MIB file. Possible values = generic, specific
trapdestination	<String>	Read-write	IPv4 or the IPv6 address of the trap listener to which the NetScaler appliance is to send SNMP trap messages. Minimum length = 1
version	<String>	Read-write	SNMP version, which determines the format of trap messages sent to the trap listener. This setting must match the setting on the trap listener. Otherwise, the listener drops the trap messages. Default value: V2 Possible values = V1, V2, V3
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
destport	<Integer>	Read-write	UDP port at which the trap listener listens for trap messages. This setting must match the setting on the trap listener. Otherwise, the listener drops the trap messages. Default value: 162 Minimum value = 1 Maximum value = 65534
communityname	<String>	Read-write	Password (string) sent with the trap messages, so that the trap listener can authenticate them. Can include 1 to 31 uppercase or lowercase letters, numbers, and hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) characters. You must specify the same community string on the trap listener device. Otherwise, the trap listener drops the trap messages. The following requirement applies only to the NetScaler CLI: If the string includes one or more spaces, enclose the name in double or single quotation marks (for example, "my string" or 'my string').
			IPv4 or IPv6 address that the NetScaler appliance inserts as the source IP address in all

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			NSIP or NSIP6 address, but you can specify an IPv4 MIP or SNIP address or a SNIP6 address. Minimum length = 1
severity	<String>	Read-write	Severity level at or above which the NetScaler appliance sends trap messages to this trap listener. The severity levels, in increasing order of severity, are Informational, Warning, Minor, Major, Critical. This parameter can be set for trap listeners of type SPECIFIC only. The default is to send all levels of trap messages. Important: Trap messages are not assigned severity levels unless you specify severity levels when configuring SNMP alarms. Default value: Unknown Possible values = Critical, Major, Minor, Warning, Informational
allpartitions	<String>	Read-write	Send traps of all partitions to this destination. Default value: DISABLED Possible values = ENABLED, DISABLED
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/snmptrap

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmptrap":{
    "trapclass":<String_value>,
    "trapdestination":<String_value>,
    "version":<String_value>,
    "td":<Double_value>,
    "destport":<Integer_value>,
    "communityname":<String_value>,
    "srcip":<String_value>,
    "severity":<String_value>,
    "allpartitions":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/snmptrap/trapclass_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/snmptrap

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmptrap":{
    "trapclass":<String_value>,
    "trapdestination":<String_value>,
    "version":<String_value>,
    "td":<Double_value>,
    "destport":<Integer_value>,
    "communityname":<String_value>,
    "srcip":<String_value>,
    "severity":<String_value>,
    "allpartitions":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/snmptrap?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmptrap":{
    "trapclass":<String_value>,
    "trapdestination":<String_value>,
    "version":<String_value>,
    "td":<Double_value>,
    "destport":true,
    "communityname":true,
    "srcip":true,
    "severity":true,
    "allpartitions":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmptrap

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmptrap?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/snmptrap?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmptrap resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/snmptrap?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/snmptrap?**pagesize=#no&pageno=#no**

Use this query-parameter to get the snmptrap resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "snmptrap": [ {
    "trapclass":<String_value>,
    "trapdestination":<String_value>,
    "version":<String_value>,
    "td":<Double_value>,
    "destport":<Integer_value>,
    "communityname":<String_value>,
    "srcip":<String_value>,
    "severity":<String_value>,
    "allpartitions":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/snmptrap?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmptrap": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmptrap_binding

May 20, 2016

Binding object which returns the resources bound to snmptrap.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
trapclass	<String>	Read-write	Trap type specified in the trap listener entry. Possible values = generic, specific
trapdestination	<String>	Read-write	IP address specified in the trap listener entry. Minimum length = 1
version	<String>	Read-write	The SNMP version of the trap specified in the trap listener entry. Possible values = V1, V2, V3
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
snmptrap_snmpuser_binding	<snmptrap_snmpuser_binding[]>	Read-only	snmpuser that can be bound to snmptrap.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/snmptrap_binding

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/snmptrap_binding?**args=trapdestination:<String_value>,version:<String_value>,td:<Double_value>,trapclass:<String_value>**

Use this query-parameter to get snmptrap_binding resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "snmptrap_binding": [ {
    "trapdestination":<String_value>,
    "version":<String_value>,
    "td":<Double_value>,
    "trapclass":<String_value>,
    "snmptrap_snmpuser_binding":<snmptrap_snmpuser_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmptrap_snmpuser_binding

May 20, 2016

Binding object showing the snmpuser that can be bound to snmptrap.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
trapclass	<String>	Read-write	Type of trap messages that the NetScaler appliance sends to the trap listener: Generic or the enterprise-specific messages defined in the MIB file. Possible values = generic, specific
securitylevel	<String>	Read-write	Security level of the SNMPv3 trap. Default value: authNoPriv, Possible values = noAuthNoPriv, authNoPriv, authPriv
username	<String>	Read-write	Name of the SNMP user that will send the SNMPv3 traps.
td	<Double>	Read-write	Integer value that uniquely identifies the traffic domain in which you want to configure the entity. If you do not specify an ID, the entity becomes part of the default traffic domain, which has an ID of 0. Minimum value = 0 Maximum value = 4094
trapdestination	<String>	Read-write	IPv4 or the IPv6 address of the trap listener to which the NetScaler appliance is to send SNMP trap messages. Minimum length = 1
version	<String>	Read-write	SNMP version, which determines the format of trap messages sent to the trap listener. This setting must match the setting on the trap listener. Otherwise, the listener drops the trap messages. Default value: V3 Possible values = V1, V2, V3
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/snmptrap_snmpuser_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"snmptrap_snmpuser_binding":{
    "trapclass":<String_value>,
    "trapdestination":<String_value>,
    "td":<Double_value>,
    "version":<String_value>,
    "username":<String_value>,
    "securitylevel":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/snmptrap_snmpuser_binding/trapclass_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/snmptrap_snmpuser_binding

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/snmptrap_snmpuser_binding?**args=trapdestination:<String_value>,version:<String_value>,td:<Double_value>,trapclass:<String_value>**

Use this query-parameter to get snmptrap_snmpuser_binding resources based on additional properties.

filter

http://<netscaler-ip-address>/nitro/v1/config/snmptrap_snmpuser_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of snmptrap_snmpuser_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/snmptrap_snmpuser_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the snmptrap_snmpuser_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmptrap_snmpuser_binding": [ {
    "trapclass":<String_value>,
    "securitylevel":<String_value>,
    "username":<String_value>,
    "td":<Double_value>,
    "trapdestination":<String_value>,
    "version":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/snmptrap_snmpuser_binding?**args=trapdestination:<String_value>,version:<String_value>,td:<Double_value>,trapclass:<String_value>&count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmptrap_snmpuser_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmpuser

May 20, 2016

Configuration for SNMP user resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the SNMPv3 user. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose it in double or single quotation marks (for example, "my user" or 'my user'). Minimum length = 1
group	<String>	Read-write	Name of the configured SNMPv3 group to which to bind this SNMPv3 user. The access rights (bound SNMPv3 views) and security level set for this group are assigned to this user. Minimum length = 1
authtype	<String>	Read-write	Authentication algorithm used by the NetScaler appliance and the SNMPv3 user for authenticating the communication between them. You must specify the same authentication algorithm when you configure the SNMPv3 user in the SNMP manager. Possible values = MD5, SHA
authpasswd	<String>	Read-write	Plain-text pass phrase to be used by the authentication algorithm specified by the authType (Authentication Type) parameter. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. The following requirement applies only to the NetScaler CLI: If the pass phrase includes one or more spaces, enclose it in double or single quotation marks (for example, "my phrase" or 'my phrase'). Minimum length = 8
privtype	<String>	Read-write	Encryption algorithm used by the NetScaler appliance and the SNMPv3 user for encrypting the communication between them. You must specify the same encryption algorithm when you configure the SNMPv3 user in the SNMP manager. Possible values = DES, AES
privpasswd	<String>	Read-write	Encryption key to be used by the encryption algorithm specified by the privType (Encryption Type) parameter. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. The following requirement applies only to the NetScaler CLI:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			example, "my key" or "my key"). Minimum.length = 8
engineid	<String>	Read-only	The context engine ID of the user.
storagetype	<String>	Read-only	The storage type for this user. Possible values = volatile, nonVolatile
status	<String>	Read-only	The status of this user. Possible values = active
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpuser

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpuser":{
    "name":<String_value>,
    "group":<String_value>,
    "authtype":<String_value>,
    "authpasswd":<String_value>,
    "privtype":<String_value>,
    "privpasswd":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpuser/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpuser

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"snmpuser":{
    "name":<String_value>,
    "group":<String_value>,
    "authtype":<String_value>,
    "authpasswd":<String_value>,
    "privtype":<String_value>,
    "privpasswd":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpuser?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpuser":{
    "name":<String_value>,
    "authtype":true,
    "privtype":true,
    "authpasswd":true,
    "privpasswd":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpuser

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpuser?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/snmpuser?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpuser resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpuser?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/snmpuser?**pagesize=#no&pageno=#no**

Use this query-parameter to get the snmpuser resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpuser": [ {
      "name":<String_value>,
      "group":<String_value>,
      "authtype":<String_value>,
      "privtype":<String_value>,
      "engineid":<String_value>,
      "storagetype":<String_value>,
      "status":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpuser/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpuser/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpuser/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpuser": [ {
      "name":<String_value>,
      "group":<String_value>,
      "authtype":<String_value>,
      "privtype":<String_value>,
      "engineid":<String_value>,
      "storagetype":<String_value>,
      "status":<String_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpuser?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "snmpuser": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmpview

May 20, 2016

Configuration for view resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the SNMPv3 view. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the SNMPv3 view. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose it in double or single quotation marks (for example, "my view" or 'my view'). Minimum length = 1
subtree	<String>	Read-write	A particular branch (subtree) of the MIB tree that you want to associate with this SNMPv3 view. You must specify the subtree as an SNMP OID. Minimum length = 1
type	<String>	Read-write	Include or exclude the subtree, specified by the subtree parameter, in or from this view. This setting can be useful when you have included a subtree, such as A, in an SNMPv3 view and you want to exclude a specific subtree of A, such as B, from the SNMPv3 view. Possible values = included, excluded
storagetype	<String>	Read-only	The storage type for this view. Possible values = volatile, nonVolatile
status	<String>	Read-only	The status of this view. Possible values = active
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpview

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpview":{
    "name":<String_value>,
    "subtree":<String_value>,
    "type":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpview/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpview

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"snmpview":{
    "name":<String_value>,
    "subtree":<String_value>,
    "type":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpview

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/snmpview?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/snmpview?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of snmpview resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/snmpview?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/snmpview?**pagesize=#no&pageno=#no**

Use this query-parameter to get the snmpview resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmpview": [ {
    "name":<String_value>,
    "subtree":<String_value>,
    "type":<String_value>,
    "storagetype":<String_value>,
    "status":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/snmpview?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "snmpview": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Spillover

May 20, 2016

Spillover policies and actions.

Name	Description
spilloveraction	Configuration for Spillover action
spilloverpolicy	Configuration for Spillover policy
spilloverpolicy_binding	Binding object showing the resources that can be bound to spilloverpolicy
spilloverpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to spilloverpolicy
spilloverpolicy_gslbvserver_binding	Binding object showing the gslbvserver that can be bound to spilloverpolicy
spilloverpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to spilloverpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

spilloveraction

May 20, 2016

Configuration for Spillover action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the spillover action.
action	<String>	Read-write	Spillover action. Currently only type SPILLOVER is supported. Possible values = SPILLOVER
newname	<String>	Read-write	New name for the spillover action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Choose a name that can be correlated with the function that the action performs. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
builtin	<String[]>	Read-only	Flag to determine whether compression is default or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloveraction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"spilloveraction":{
    "name":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloveraction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloveraction?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"spilloveraction":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloveraction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/spilloveraction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/spilloveraction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of spilloveraction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/spilloveraction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/spilloveraction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the spilloveraction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "spilloveraction": [ {
    "name":<String_value>,
    "action":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloveraction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/spilloveraction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/spilloveraction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "spilloveraction": [ {
      "name":<String_value>,
      "action":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloveraction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "spilloveraction": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

spilloverpolicy

May 20, 2016

Configuration for Spillover policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the spillover policy.
rule	<String>	Read-write	Expression to be used by the spillover policy.
action	<String>	Read-write	Action for the spillover policy. Action is created using add spillover action command.
comment	<String>	Read-write	Any comments that you might want to associate with the spillover policy.
newname	<String>	Read-write	New name for the spillover policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Choose a name that reflects the function that the policy performs. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
hits	<Double>	Read-only	The number of times the policy has been hit.
undefhits	<Double>	Read-only	Number of policy UNDEF hits.
builtin	<String[]>	Read-only	Flag to determine if compression policy is builtin or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"spilloverpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

update

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"spilloverpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy?**action=unset**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"spilloverpolicy":{
    "name":<String_value>,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"spilloverpolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of spilloverpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the spilloverpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "spilloverpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "comment":<String_value>,
    "builtin":<String[]_value>
}]}
```

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "spilloverpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "comment":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "spilloverpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

spilloverpolicy_binding

May 20, 2016

Binding object which returns the resources bound to spilloverpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the spillover policy.
spilloverpolicy_gslbvserver_binding	<spilloverpolicy_gslbvserver_binding[]>	Read-only	gslbvserver that can be bound to spilloverpolicy.
spilloverpolicy_csvserver_binding	<spilloverpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to spilloverpolicy.
spilloverpolicy_lbvserver_binding	<spilloverpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to spilloverpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "spilloverpolicy_binding": [ {
      "name":<String_value>,
      "spilloverpolicy_csvserver_binding":<spilloverpolicy_csvserver_binding[]_value>,
      "spilloverpolicy_lbvserver_binding":<spilloverpolicy_lbvserver_binding[]_value>,
      "spilloverpolicy_gslbvserver_binding":<spilloverpolicy_gslbvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

spilloverpolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to spilloverpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
name	<String>	Read-write	Name of the spillover policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of spilloverpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the spilloverpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "spilloverpolicy_csvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"spilloverpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

spilloverpolicy_gslbvserver_binding

May 20, 2016

Binding object showing the gslbvserver that can be bound to spilloverpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
name	<String>	Read-write	Name of the spillover policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_gslbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_gslbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of spilloverpolicy_gslbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_gslbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the spilloverpolicy_gslbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "spilloverpolicy_gslbvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_gslbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"spilloverpolicy_gslbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

spilloverpolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to spilloverpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The name of the entity to which the policy is bound.
name	<String>	Read-write	Name of the spillover policy.
priority	<Double>	Read-only	Specifies the priority of the policy.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, resvserver, policylabel
activepolicy	<Double>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of spilloverpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the spilloverpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "spilloverpolicy_lbvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/spilloverpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"spilloverpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL

May 20, 2016

SSL Configuration

Name	Description
sslaction	Configuration for SSL action
sslcert	Configuration for cerificate
sslcertchain	Configuration for CERT Chain
sslcertchain_binding	Binding object showing the resources that can be bound to sslcertchain
sslcertchain_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to sslcertchain
sslcertfile	Configuration for Imported Certfile
sslcertkey	Configuration for certificate key
sslcertkey_binding	Binding object showing the resources that can be bound to sslcertkey
sslcertkey_crldistribution_binding	Binding object showing the crldistribution that can be bound to sslcertkey
sslcertkey_service_binding	Binding object showing the service that can be bound to sslcertkey
sslcertkey_sslocspresponder_binding	Binding object showing the sslocspresponder that can be bound to sslcertkey
sslcertkey_sslvserver_binding	Binding object showing the sslvserver that can be bound to sslcertkey
sslcertlink	Configuration for linked certificate
sslcertreq	Configuration for certificate request
sslcipher	Configuration for cipher

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcipher_binding	
Name	**Description**
sslcipher_individualcipher_binding	Binding object showing the individualcipher that can be bound to sslcipher
sslcipher_service_binding	Binding object showing the service that can be bound to sslcipher
sslcipher_servicegroup_binding	Binding object showing the servicegroup that can be bound to sslcipher
sslcipher_sslciphersuite_binding	Binding object showing the sslciphersuite that can be bound to sslcipher
sslcipher_sslprofile_binding	Binding object showing the sslprofile that can be bound to sslcipher
sslcipher_sslvserver_binding	Binding object showing the sslvserver that can be bound to sslcipher
sslciphersuite	Configuration for ciphersuite
sslcrl	Configuration for Certificate Revocation List
sslcrl_binding	Binding object showing the resources that can be bound to sslcrl
sslcrl_serialnumber_binding	Binding object showing the serialnumber that can be bound to sslcrl
sslcrlfile	Configuration for Imported crl files
ssldhfile	Configuration for dh imported file
ssldhparam	Configuration for dh Parameter
ssldsakey	Configuration for dsa key
ssldtlsprofile	Configuration for DTLS profile
sslfips	Configuration for fips
sslfipskey	Configuration for FIPS key
sslfipssimsource	Configuration for FIPsSIM source

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslfipssimtarget	
Name	**Description**
sslglobal_binding	Binding object showing the resources that can be bound to sslglobal
sslglobal_sslpolicy_binding	Binding object showing the sslpolicy that can be bound to sslglobal
sslhsmkey	Configuration for HSM key
sslkeyfile	Configuration for Imported ssl key files
sslocspresponder	Configuration for OCSP responser
sslparameter	Configuration for SSL parameter
sslpkcs12	Configuration for pkcs12
sslpkcs8	Configuration for pkcs8
sslpolicy	Configuration for SSL policy
sslpolicy_binding	Binding object showing the resources that can be bound to sslpolicy
sslpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to sslpolicy
sslpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to sslpolicy
sslpolicy_sslglobal_binding	Binding object showing the sslglobal that can be bound to sslpolicy
sslpolicy_sslpolicylabel_binding	Binding object showing the sslpolicylabel that can be bound to sslpolicy
sslpolicy_sslservice_binding	Binding object showing the sslservice that can be bound to sslpolicy
sslpolicy_sslvserver_binding	Binding object showing the sslvserver that can be bound to sslpolicy
sslpolicylabel	Configuration for SSL policy label
sslpolicylabel_binding	Binding object showing the resources that can be bound to sslpolicylabel

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
sslpolicylabel_sslpolicy_binding	Binding object showing the sslpolicy that can be bound to sslpolicylabel
sslprofile	Configuration for SSL profile
sslprofile_binding	Binding object showing the resources that can be bound to sslprofile
sslprofile_ecccurve_binding	Binding object showing the ecccurve that can be bound to sslprofile
sslprofile_sslcipher_binding	Binding object showing the sslcipher that can be bound to sslprofile
sslprofile_sslciphersuite_binding	Binding object showing the sslciphersuite that can be bound to sslprofile
sslprofile_sslvserver_binding	Binding object showing the sslvserver that can be bound to sslprofile
sslrsakey	Configuration for RSA key
sslservice	Configuration for SSL service
sslservice_binding	Binding object showing the resources that can be bound to sslservice
sslservice_ecccurve_binding	Binding object showing the ecccurve that can be bound to sslservice
sslservice_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to sslservice
sslservice_sslcipher_binding	Binding object showing the sslcipher that can be bound to sslservice
sslservice_sslciphersuite_binding	Binding object showing the sslciphersuite that can be bound to sslservice
sslservice_sslpolicy_binding	Binding object showing the sslpolicy that can be bound to sslservice
sslservicegroup	Configuration for SSL service group
sslservicegroup_binding	Binding object showing the resources that can be bound to sslservicegroup
sslservicegroup_ecccurve_binding	Binding object showing the ecccurve that can be bound to sslservicegroup
sslservicegroup_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to sslservicegroup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
sslservicegroup_sslcipher_binding	Binding object showing the sslcipher that can be bound to sslservicegroup
sslservicegroup_sslciphersuite_binding	Binding object showing the sslciphersuite that can be bound to sslservicegroup
sslvserver	Configuration for SSL virtual server
sslvserver_binding	Binding object showing the resources that can be bound to sslvserver
sslvserver_ecccurve_binding	Binding object showing the ecccurve that can be bound to sslvserver
sslvserver_sslcertkey_binding	Binding object showing the sslcertkey that can be bound to sslvserver
sslvserver_sslcipher_binding	Binding object showing the sslcipher that can be bound to sslvserver
sslvserver_sslciphersuite_binding	Binding object showing the sslciphersuite that can be bound to sslvserver
sslvserver_sslpolicy_binding	Binding object showing the sslpolicy that can be bound to sslvserver
sslwrapkey	Configuration for WRAP key

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcert

May 20, 2016

Configuration for cerificate resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
certfile	<String>	Read-write	Name for and, optionally, path to the generated certificate file. /nsconfig/ssl/ is the default path. Maximum length = 63
reqfile	<String>	Read-write	Name for and, optionally, path to the certificate-signing request (CSR). /nsconfig/ssl/ is the default path. Maximum length = 63
certtype	<String>	Read-write	Type of certificate to generate. Specify one of the following: * ROOT_CERT - Self-signed Root-CA certificate. You must specify the key file name. The generated Root-CA certificate can be used for signing end-user client or server certificates or to create Intermediate-CA certificates. * INTM_CERT - Intermediate-CA certificate. * CLNT_CERT - End-user client certificate used for client authentication. * SRVR_CERT - SSL server certificate used on SSL servers for end-to-end encryption. Possible values = ROOT_CERT, INTM_CERT, CLNT_CERT, SRVR_CERT
keyfile	<String>	Read-write	Name for and, optionally, path to the private key. You can either use an existing RSA or DSA key that you own or create a new private key on the NetScaler appliance. This file is required only when creating a self-signed Root-CA certificate. The key file is stored in the /nsconfig/ssl directory by default. If the input key specified is an encrypted key, you are prompted to enter the PEM pass phrase that was used for encrypting the key. Maximum length = 63
keyform	<String>	Read-write	Format in which the key is stored on the appliance. Default value: PEM Possible values = DER, PEM
pempassphrase	<String>	Read-write	. Minimum length = 1 Maximum length = 31
days	<Double>	Read-write	Number of days for which the certificate will be valid, beginning with the time and day (system time) of creation. Default value: 365 Minimum value = 1 Maximum value = 3650

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: PEM Possible values = DER, PEM
cacert	<String>	Read-write	Name of the CA certificate file that issues and signs the Intermediate-CA certificate or the end-user client and server certificates. Maximum length = 63
cacertform	<String>	Read-write	Format of the CA certificate. Default value: PEM Possible values = DER, PEM
cakey	<String>	Read-write	Private key, associated with the CA certificate that is used to sign the Intermediate-CA certificate or the end-user client and server certificate. If the CA key file is password protected, the user is prompted to enter the pass phrase that was used to encrypt the key. Maximum length = 63
cakeyform	<String>	Read-write	Format for the CA certificate. Default value: PEM Possible values = DER, PEM
caserial	<String>	Read-write	Serial number file maintained for the CA certificate. This file contains the serial number of the next certificate to be issued or signed by the CA. If the specified file does not exist, a new file is created, with /nsconfig/ssl/ as the default path. If you do not specify a proper path for the existing serial file, a new serial file is created. This might change the certificate serial numbers assigned by the CA certificate to each of the certificates it signs. Maximum length = 63

Operations (click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

create

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcert?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcert":{
    "certfile":<String_value>,
    "reqfile":<String_value>,
    "certtype":<String_value>,
    "keyfile":<String_value>,
    "keyform":<String_value>,
    "pempassphrase":<String_value>,
    "days":<Double_value>,
    "certform":<String_value>,
    "cacert":<String_value>,
    "cacertform":<String_value>,
    "cakey":<String_value>,
    "cakeyform":<String_value>,
    "caserial":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertchain

May 20, 2016

Configuration for CERT Chain resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
certkeyname	<String>	Read-write	Name of the Certificate. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertchain

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcertchain?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcertchain?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcertchain resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcertchain?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcertchain?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcertchain resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcertchain": [ {
    "certkeyname":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertchain/certkeyname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcertchain/certkeyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcertchain/certkeyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcertchain": [ {
      "certkeyname":<String_value>,

}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertchain?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "sslcertchain": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertchain_binding

May 20, 2016

Binding object which returns the resources bound to sslcertchain.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
certkeyname	<String>	Read-write	Name of the Certificate. Minimum length = 1
sslcertchain_sslcertkey_binding	<sslcertchain_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to sslcertchain.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertchain_binding/certkeyname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "sslcertchain_binding": [ {
    "certkeyname":<String_value>,
    "sslcertchain_sslcertkey_binding":<sslcertchain_sslcertkey_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertchain_sslcertkey_binding

May 20, 2016

Binding object showing the sslcertkey that can be bound to sslcertchain.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
linkcertkeyname	<String>	Read-write	Name of the Linked Certificate. Minimum length = 1
certkeyname	<String>	Read-write	Name of the Certificate. Minimum length = 1
isca	<Boolean>	Read-only	Used to find if certificate is a CA.
islinked	<Boolean>	Read-only	Used to find if certificate is linked.
addsubject	<Boolean>	Read-only	Used to find if certificate is linked.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertchain_sslcertkey_binding/certkeyname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcertchain_sslcertkey_binding/certkeyname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertchain_sslcertkey_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcertchain_sslcertkey_binding/certkeyname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcertchain_sslcertkey_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "sslcertchain_sslcertkey_binding": [ {
      "linkcertkeyname":<String_value>,
      "certkeyname":<String_value>,
      "isca":<Boolean_value>,
      "islinked":<Boolean_value>,
      "addsubject":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertchain_sslcertkey_binding/certkeyname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslcertchain_sslcertkey_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertfile

May 20, 2016

Configuration for Imported Certfile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name to assign to the imported certificate file. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file'). Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL specifying the protocol, host, and path, including file name, to the certificate file to be imported. For example, http://www.example.com/cert_file. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

IMPORT | DELETE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertfile?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcertfile":{
    "name":<String_value>,
    "src":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertfile

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertfile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcertfile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcertfile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcertfile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcertfile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcertfile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcertfile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "sslcertfile": [ {
      "name":<String_value>,
      "src":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertfile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcertfile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertkey

May 20, 2016

Configuration for certificate key resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
certkey	<String>	Read-write	Name for the certificate and private-key pair. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the certificate-key pair is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cert" or 'my cert'). Minimum length = 1
cert	<String>	Read-write	Name of and, optionally, path to the X509 certificate file that is used to form the certificate-key pair. The certificate file should be present on the appliance's hard-disk drive or solid-state drive. Storing a certificate in any location other than the default might cause inconsistency in a high availability setup. /nsconfig/ssl/ is the default path. Minimum length = 1
key	<String>	Read-write	Name of and, optionally, path to the private-key file that is used to form the certificate-key pair. The certificate file should be present on the appliance's hard-disk drive or solid-state drive. Storing a certificate in any location other than the default might cause inconsistency in a high availability setup. /nsconfig/ssl/ is the default path. Minimum length = 1
password	<Boolean>	Read-write	Passphrase that was used to encrypt the private-key. Use this option to load encrypted private-keys in PEM format.
fipskey	<String>	Read-write	Name of the FIPS key that was created inside the Hardware Security Module (HSM) of a FIPS appliance, or a key that was imported into the HSM. Minimum length = 1
hsmkey	<String>	Read-write	Name of the HSM key that was created in the External Hardware Security Module (HSM) of a FIPS appliance. Minimum length = 1
			Input format of the certificate and the private-key files. The three formats supported by the appliance are: PEM - Privacy Enhanced Mail

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			PFX - Personal Information Exchange. Default value: PEM Possible values = DER, PEM, PFX
passplain	<String>	Read-write	Pass phrase used to encrypt the private-key. Required when adding an encrypted private-key in PEM format. Minimum length = 1
expirymonitor	<String>	Read-write	Issue an alert when the certificate is about to expire. Possible values = ENABLED, DISABLED
notificationperiod	<Double>	Read-write	Time, in number of days, before certificate expiration, at which to generate an alert that the certificate is about to expire. Minimum value = 10 Maximum value = 100
bundle	<String>	Read-write	Parse the certificate chain as a single file after linking the server certificate to its issuer's certificate within the file. Default value: NO Possible values = YES, NO
linkcertkeyname	<String>	Read-write	Name of the Certificate Authority certificate-key pair to which to link a certificate-key pair. Minimum length = 1
nodomaincheck	<Boolean>	Read-write	Override the check for matching domain names during a certificate update operation.
signaturealg	<String>	Read-only	Signature algorithm.
serial	<String>	Read-only	Serial number.
issuer	<String>	Read-only	Issuer name.
clientcertnotbefore	<String>	Read-only	Not-Before date.
clientcertnotafter	<String>	Read-only	Not-After date.
daystoexpiration	<Integer>	Read-only	Days remaining for the certificate to expire.
subject	<String>	Read-only	Subject name.
publickey	<String>	Read-only	Public key algorithm.
publickeysize	<Integer>	Read-only	Size of the public key.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
priority	<Double>	Read-only	ocsp priority.
status	<String>	Read-only	Status of the certificate. Possible values = Valid, Not yet valid, Expired
passcrypt	<String>	Read-only	Passcrypt. Minimum length = 1
data	<Double>	Read-only	Vserver Id.
servicename	<String>	Read-only	Service name to which the certificate key pair is bound.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | LINK | UNLINK | CHANGE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcertkey":{
    "certkey":<String_value>,
    "cert":<String_value>,
    "key":<String_value>,
    "password":<Boolean_value>,
    "fipskey":<String_value>,
    "hsmkey":<String_value>,
    "inform":<String_value>,
    "passplain":<String_value>,
    "expirymonitor":<String_value>,
    "notificationperiod":<Double_value>,
    "bundle":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey/certkey_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcertkey":{
    "certkey":<String_value>,
    "expirymonitor":<String_value>,
    "notificationperiod":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcertkey":{
    "certkey":<String_value>,
    "expirymonitor":true,
    "notificationperiod":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

link

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey?**action=link**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcertkey":{
    "certkey":<String_value>,
    "linkcertkeyname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unlink

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey?**action=unlink**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcertkey":{
    "certkey":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

change

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey?**action=update**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcertkey":{
    "certkey":<String_value>,
    "cert":<String_value>,
    "key":<String_value>,
    "password":<Boolean_value>,
    "fipskey":<String_value>,
    "inform":<String_value>,
    "passplain":<String_value>,
    "nodomaincheck":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcertkey resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey?**pagesize=#no&pageno=#no**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the sslcertkey resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcertkey": [ {
    "certkey":<String_value>,
    "cert":<String_value>,
    "key":<String_value>,
    "inform":<String_value>,
    "signaturealg":<String_value>,
    "serial":<String_value>,
    "issuer":<String_value>,
    "clientcertnotbefore":<String_value>,
    "clientcertnotafter":<String_value>,
    "daystoexpiration":<Integer_value>,
    "subject":<String_value>,
    "publickey":<String_value>,
    "publickeysize":<Integer_value>,
    "version":<Integer_value>,
    "priority":<Double_value>,
    "status":<String_value>,
    "fipskey":<String_value>,
    "hsmkey":<String_value>,
    "passcrypt":<String_value>,
    "data":<Double_value>,
    "servicename":<String_value>,
    "expirymonitor":<String_value>,
    "notificationperiod":<Double_value>,
    "linkcertkeyname":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey/certkey_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey/certkey_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey/certkey_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcertkey": [ {
      "certkey":<String_value>,
      "cert":<String_value>,
      "key":<String_value>,
      "inform":<String_value>,
      "signaturealg":<String_value>,
      "serial":<String_value>,
      "issuer":<String_value>,
      "clientcertnotbefore":<String_value>,
      "clientcertnotafter":<String_value>,
      "daystoexpiration":<Integer_value>,
      "subject":<String_value>,
      "publickey":<String_value>,
      "publickeysize":<Integer_value>,
      "version":<Integer_value>,
      "priority":<Double_value>,
      "status":<String_value>,
      "fipskey":<String_value>,
      "hsmkey":<String_value>,
      "passcrypt":<String_value>,
      "data":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "servicename":<String_value>,
    "expirymonitor":<String_value>,
    "notificationperiod":<Double_value>,
    "linkcertkeyname":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcertkey": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertkey_binding

May 20, 2016

Binding object which returns the resources bound to sslcertkey.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
certkey	<String>	Read-write	Name of the certificate-key pair for which to show detailed information. Minimum length = 1
sslcertkey_sslvserver_binding	<sslcertkey_sslvserver_binding[]>	Read-only	sslvserver that can be bound to sslcertkey.
sslcertkey_sslocspresponder_binding	<sslcertkey_sslocspresponder_binding[]>	Read-only	sslocspresponder that can be bound to sslcertkey.
sslcertkey_service_binding	<sslcertkey_service_binding[]>	Read-only	service that can be bound to sslcertkey.
sslcertkey_crldistribution_binding	<sslcertkey_crldistribution_binding[]>	Read-only	crldistribution that can be bound to sslcertkey.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_binding/certkey_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "sslcertkey_binding": [ {
    "certkey":<String_value>,
    "sslcertkey_crldistribution_binding":<sslcertkey_crldistribution_binding[]_value>,
    "sslcertkey_sslvserver_binding":<sslcertkey_sslvserver_binding[]_value>,
    "sslcertkey_sslocspresponder_binding":<sslcertkey_sslocspresponder_binding[]_value>,
    "sslcertkey_service_binding":<sslcertkey_service_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertkey_crldistribution_binding

May 20, 2016

Binding object showing the crldistribution that can be bound to sslcertkey.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
crlcheck	<String>	Read-write	The rule for use of CRL corresponding to this CA certificate during client authentication. If crlCheck is set to Mandatory, the system will deny all SSL clients if the CRL is missing, expired - NextUpdate date is in the past, or is incomplete with remote CRL refresh enabled. If crlCheck is set to optional, the system will allow SSL clients in the above error cases.However, in any case if the client certificate is revoked in the CRL, the SSL client will be denied access. Default value: CRLCHECK_OPTIONAL Possible values = Mandatory, Optional
ca	<Boolean>	Read-write	If this option is specified, it indicates that the certificate-key pair being bound to the SSL virtual server is a CA certificate. If this option is not specified, the certificate-key pair is bound as a normal server certificate. Note: In case of a normal server certificate, the certificate-key pair should consist of both the certificate and the private-key.
issuer	<String>	Read-write	Issuer name.
certkey	<String>	Read-write	Name of the certificate-key pair. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_crldistribution_binding/certkey_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_crldistribution_binding/certkey_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertkey_crldistribution_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_crldistribution_binding/certkey_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcertkey_crldistribution_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcertkey_crldistribution_binding": [ {
    "crlcheck":<String_value>,
    "ca":<Boolean_value>,
    "issuer":<String_value>,
    "certkey":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_crldistribution_binding/certkey_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslcertkey_crldistribution_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertkey_service_binding

May 20, 2016

Binding object showing the service that can be bound to sslcertkey.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	The name of the SSL service group to which the certificate-key pair needs to be bound. Use the "add servicegroup" command to create this service. Minimum length = 1
crlcheck	<String>	Read-write	The rule for use of CRL corresponding to this CA certificate during client authentication. If crlCheck is set to Mandatory, the system will deny all SSL clients if the CRL is missing, expired - NextUpdate date is in the past, or is incomplete with remote CRL refresh enabled. If crlCheck is set to optional, the system will allow SSL clients in the above error cases.However, in any case if the client certificate is revoked in the CRL, the SSL client will be denied access. Default value: CRLCHECK_OPTIONAL Possible values = Mandatory, Optional
ca	<Boolean>	Read-write	If this option is specified, it indicates that the certificate-key pair being bound to the SSL virtual server is a CA certificate. If this option is not specified, the certificate-key pair is bound as a normal server certificate. Note: In case of a normal server certificate, the certificate-key pair should consist of both the certificate and the private-key.
service	<Boolean>	Read-write	Bind the certificate to the named SSL service or service group.
servicename	<String>	Read-write	Service name to which the certificate key pair is bound.
certkey	<String>	Read-write	Name of the certificate-key pair. Minimum length = 1
version	<Integer>	Read-only	Version.
data	<Double>	Read-only	Vserver Id.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_service_binding/certkey_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_service_binding/certkey_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertkey_service_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_service_binding/certkey_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcertkey_service_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcertkey_service_binding": [ {
    "servicegroupname":<String_value>,
    "crlcheck":<String_value>,
    "ca":<Boolean_value>,
    "service":<Boolean_value>,
    "servicename":<String_value>,
    "certkey":<String_value>,
    "version":<Integer_value>,
    "data":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_service_binding/certkey_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"sslcertkey_service_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertkey_sslocspresponder_binding

May 20, 2016

Binding object showing the sslocspresponder that can be bound to sslcertkey.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	ocsp priority.
crlcheck	<String>	Read-write	The rule for use of CRL corresponding to this CA certificate during client authentication. If crlCheck is set to Mandatory, the system will deny all SSL clients if the CRL is missing, expired - NextUpdate date is in the past, or is incomplete with remote CRL refresh enabled. If crlCheck is set to optional, the system will allow SSL clients in the above error cases.However, in any case if the client certificate is revoked in the CRL, the SSL client will be denied access. Default value: CRLCHECK_OPTIONAL Possible values = Mandatory, Optional
ca	<Boolean>	Read-write	If this option is specified, it indicates that the certificate-key pair being bound to the SSL virtual server is a CA certificate. If this option is not specified, the certificate-key pair is bound as a normal server certificate. Note: In case of a normal server certificate, the certificate-key pair should consist of both the certificate and the private-key.
certkey	<String>	Read-write	Name of the certificate-key pair. Minimum length = 1
ocspresponder	<String>	Read-write	OCSP responders bound to this certkey.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslcertkey_sslocspresponder_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslcertkey_sslocspresponder_binding":{
    "certkey":<String_value>,
    "ocspresponder":<String_value>,
    "priority":<Double_value>,
    "ca":<Boolean_value>,
    "crlcheck":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertkey_sslocspresponder_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcertkey_sslocspresponder_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "sslcertkey_sslocspresponder_binding": [ {
    "priority":<Double_value>,
    "crlcheck":<String_value>,
    "ca":<Boolean_value>,
    "certkey":<String_value>,
    "ocspresponder":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_sslocspresponder_binding/certkey_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslcertkey_sslocspresponder_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertkey_sslvserver_binding

May 20, 2016

Binding object showing the sslvserver that can be bound to sslcertkey.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<Boolean>	Read-write	Specify this option to bind the certificate to an SSL virtual server. Note: The default option is -vServer.
crlcheck	<String>	Read-write	The rule for use of CRL corresponding to this CA certificate during client authentication. If crlCheck is set to Mandatory, the system will deny all SSL clients if the CRL is missing, expired - NextUpdate date is in the past, or is incomplete with remote CRL refresh enabled. If crlCheck is set to optional, the system will allow SSL clients in the above error cases.However, in any case if the client certificate is revoked in the CRL, the SSL client will be denied access. Default value: CRLCHECK_OPTIONAL Possible values = Mandatory, Optional
ca	<Boolean>	Read-write	If this option is specified, it indicates that the certificate-key pair being bound to the SSL virtual server is a CA certificate. If this option is not specified, the certificate-key pair is bound as a normal server certificate. Note: In case of a normal server certificate, the certificate-key pair should consist of both the certificate and the private-key.
vservername	<String>	Read-write	The name of the SSL virtual server name to which the certificate-key pair needs to be bound. Minimum length = 1
servername	<String>	Read-write	Vserver name to which the certificate key pair is bound.
certkey	<String>	Read-write	Name of the certificate-key pair. Minimum length = 1
version	<Integer>	Read-only	Version.
data	<Double>	Read-only	Vserver Id.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_sslvserver_binding/certkey_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_sslvserver_binding/certkey_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcertkey_sslvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_sslvserver_binding/certkey_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcertkey_sslvserver_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcertkey_sslvserver_binding": [ {
    "vserver":<Boolean_value>,
    "crlcheck":<String_value>,
    "ca":<Boolean_value>,
    "vservername":<String_value>,
    "servername":<String_value>,
    "certkey":<String_value>,
    "version":<Integer_value>,
    "data":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertkey_sslvserver_binding/certkey_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"sslcertkey_sslvserver_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertlink

May 20, 2016

Configuration for linked certificate resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
certkeyname	<String>	Read-only	Certificate key name.
linkcertkeyname	<String>	Read-only	Name of the Certificate-Authority.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertlink

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcertlink?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcertlink?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcertlink resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcertlink?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcertlink?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcertlink resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "sslcertlink": [ {
      "certkeyname":<String_value>,
      "linkcertkeyname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertlink?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcertlink": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslglobal_sslpolicy_binding

May 20, 2016

Binding object showing the sslpolicy that can be bound to sslglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy binding.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	The name for the SSL policy.
labelname	<String>	Read-write	Name of the virtual server or user-defined policy label to invoke if the policy evaluates to TRUE.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label. Default value: "END"
invoke	<Boolean>	Read-write	Invoke policies bound to a virtual server, service, or policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
type	<String>	Read-write	Possible values = CONTROL_OVERRIDE, CONTROL_DEFAULT, DATA_OVERRIDE, DATA_DEFAULT
labeltype	<String>	Read-write	Type of policy label to invoke. Specify virtual server for a policy label associated with a virtual server, or policy label for a user-defined policy label. Possible values = vserver, service, policylabel
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslglobal_sslpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{
"sslglobal_sslpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslglobal_sslpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslglobal_sslpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslglobal_sslpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslglobal_sslpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslglobal_sslpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslglobal_sslpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslglobal_sslpolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "type":<String_value>,
    "labeltype":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslglobal_sslpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "sslglobal_sslpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcertreq

May 20, 2016

Configuration for certificate request resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
reqfile	<String>	Read-write	Name for and, optionally, path to the certificate signing request (CSR). /nsconfig/ssl/ is the default path. Maximum length = 63
keyfile	<String>	Read-write	Name of and, optionally, path to the private key used to create the certificate signing request, which then becomes part of the certificate-key pair. The private key can be either an RSA or a DSA key. The key must be present in the appliance's local storage. /nsconfig/ssl is the default path. Maximum length = 63
fipskeyname	<String>	Read-write	Name of the FIPS key used to create the certificate signing request. FIPS keys are created inside the Hardware Security Module of the FIPS card. Minimum length = 1 Maximum length = 31
keyform	<String>	Read-write	Format in which the key is stored on the appliance. Default value: PEM Possible values = DER, PEM
pempassphrase	<String>	Read-write	. Minimum length = 1 Maximum length = 31
countryname	<String>	Read-write	Two letter ISO code for your country. For example, US for United States. Minimum length = 2 Maximum length = 2
statename	<String>	Read-write	Full name of the state or province where your organization is located. Do not abbreviate. Minimum length = 1
organizationname	<String>	Read-write	Name of the organization that will use this certificate. The organization name (corporation, limited partnership, university, or government agency) must be registered with some authority at the national, state, or city level. Use the legal name under which the organization is registered. Do not abbreviate the organization name and do not use the following characters in the name: Angle brackets (< >) tilde (~), exclamation mark, at (@), pound (#), zero (0), caret (^), asterisk (*), forward slash (/), square brackets ([ ]), question mark (?).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
organizationunitname	<String>	Read-write	Name of the division or section in the organization that will use the certificate. Minimum length = 1
localityname	<String>	Read-write	Name of the city or town in which your organization's head office is located. Minimum length = 1
commonname	<String>	Read-write	Fully qualified domain name for the company or web site. The common name must match the name used by DNS servers to do a DNS lookup of your server. Most browsers use this information for authenticating the server's certificate during the SSL handshake. If the server name in the URL does not match the common name as given in the server certificate, the browser terminates the SSL handshake or prompts the user with a warning message. Do not use wildcard characters, such as asterisk (*) or question mark (?), and do not use an IP address as the common name. The common name must not contain the protocol specifier <http://> or <https://>. Minimum length = 1
emailaddress	<String>	Read-write	Contact person's e-mail address. This address is publically displayed as part of the certificate. Provide an e-mail address that is monitored by an administrator who can be contacted about the certificate. Minimum length = 1
challengepassword	<String>	Read-write	Pass phrase, embedded in the certificate signing request that is shared only between the client or server requesting the certificate and the SSL certificate issuer (typically the certificate authority). This pass phrase can be used to authenticate a client or server that is requesting a certificate from the certificate authority. Minimum length = 4
companyname	<String>	Read-write	Additional name for the company or web site. Minimum length = 1

Operations (click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

create

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcertreq?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcertreq":{
    "reqfile":<String_value>,
    "keyfile":<String_value>,
    "fipskeyname":<String_value>,
    "keyform":<String_value>,
    "pempassphrase":<String_value>,
    "countryname":<String_value>,
    "statename":<String_value>,
    "organizationname":<String_value>,
    "organizationunitname":<String_value>,
    "localityname":<String_value>,
    "commonname":<String_value>,
    "emailaddress":<String_value>,
    "challengepassword":<String_value>,
    "companyname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcipher

May 20, 2016

Configuration for cipher resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphergroupname	<String>	Read-write	Name for the user-defined cipher group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the cipher group is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my ciphergroup" or 'my ciphergroup'). Minimum length = 1
ciphgrpalias	<String>	Read-write	The individual cipher name(s), a user-defined cipher group, or a system predefined cipher alias that will be added to the predefined cipher alias that will be added to the group cipherGroupName. If a cipher alias or a cipher group is specified, all the individual ciphers in the cipher alias or group will be added to the user-defined cipher group. Minimum length = 1
ciphername	<String>	Read-write	Cipher name.
cipherpriority	<Double>	Read-write	This indicates priority assigned to the particular cipher. Minimum value = 1
sslprofile	<String>	Read-write	Name of the profile to which cipher is attached.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcipher":{
    "ciphergroupname":<String_value>,
    "ciphgrpalias":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcipher":{
    "ciphergroupname":<String_value>,
    "ciphername":<String_value>,
    "cipherpriority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcipher":{
    "ciphergroupname":<String_value>,
    "ciphername":true,
    "cipherpriority":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher/ciphergroupname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/sslcipher?**args=ciphergroupname:<String_value>,sslprofile:<String_value>**

Use this query-parameter to get sslcipher resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcipher?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcipher?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcipher resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcipher?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcipher?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcipher resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcipher": [ {
ciphergroupname:<String_value>,sslprofile:<String_value>      "ciphername":<String_value>,
      "cipherpriority":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher/ciphergroupname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcipher/ciphergroupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcipher/ciphergroupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcipher": [ {
ciphergroupname:<String_value>,sslprofile:<String_value>      "ciphername":<String_value>,
      "cipherpriority":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcipher": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslciphersuite

May 20, 2016

Configuration for ciphersuite resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	Name of the cipher suite for which to show detailed information.
description	<String>	Read-only	Cipher suite description.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslciphersuite

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslciphersuite?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslciphersuite?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslciphersuite resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslciphersuite?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslciphersuite?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslciphersuite resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "sslciphersuite": [ {
    "ciphername":<String_value>,
    "description":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslciphersuite/ciphername_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslciphersuite/ciphername_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslciphersuite/ciphername_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslciphersuite": [ {
    "ciphername":<String_value>,
    "description":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslciphersuite?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslciphersuite": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcipher_binding

May 20, 2016

Binding object which returns the resources bound to sslcipher.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphergroupname	<String>	Read-write	Name of the cipher group for which to show detailed information. Minimum length = 1
sslcipher_individualcipher_binding	<sslcipher_individualcipher_binding[]>	Read-only	individualcipher that can be bound to sslcipher.
sslcipher_sslprofile_binding	<sslcipher_sslprofile_binding[]>	Read-only	sslprofile that can be bound to sslcipher.
sslcipher_sslciphersuite_binding	<sslcipher_sslciphersuite_binding[]>	Read-only	sslciphersuite that can be bound to sslcipher.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher_binding/ciphergroupname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "sslcipher_binding": [ {
      "ciphergroupname":<String_value>,
      "sslcipher_sslciphersuite_binding":<sslcipher_sslciphersuite_binding[]_value>,
      "sslcipher_individualcipher_binding":<sslcipher_individualcipher_binding[]_value>,
      "sslcipher_sslprofile_binding":<sslcipher_sslprofile_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcipher_individualcipher_binding

May 20, 2016

Binding object showing the individualcipher that can be bound to sslcipher.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	Cipher name.
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
cipherpriority	<Double>	Read-write	This indicates priority assigned to the particular cipher. Minimum value = 1
description	<String>	Read-write	Cipher suite description.
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform

© 1999-2017 Citrix Systems, Inc. All rights reserved.

individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher_individualcipher_binding/ciphergroupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcipher_individualcipher_binding/ciphergroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcipher_individualcipher_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcipher_individualcipher_binding/ciphergroupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcipher_individualcipher_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcipher_individualcipher_binding": [ {
    "ciphername":<String_value>,
    "ciphgrpals":<String_value>,
    "cipherpriority":<Double_value>,
    "description":<String_value>,
    "ciphergroupname":<String_value>,
    "cipheroperation":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher_individualcipher_binding/ciphergroupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslcipher_individualcipher_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcipher_servicegroup_binding

Jul 07, 2017

Binding object showing the servicegroup that can be bound to sslcipher.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cipherpriority	<Double>	Read-write	Priority of the cipher to be added. Minimum value = 1 Maximum value = 1000
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
servicegroupname	<String>	Read-write	The name of the SSL service name to which the cipher-suite is to be bound. Minimum length = 1
servicegroup	<Boolean>	Read-write	Indicates that the cipher operation is to be performed on the named SSL service or service group.
service	<Boolean>	Read-write	Indicates that the cipher operation is to be performed on the named SSL service or service group.
servicename	<String>	Read-write	The name of the SSL service name to which the cipher-suite is to be bound. Minimum length = 1
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD

Operations (click to see Properties)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcipher_service_binding

May 20, 2016

Binding object showing the service that can be bound to sslcipher.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cipherpriority	<Double>	Read-write	Priority of the cipher to be added. Minimum value = 1 Maximum value = 1000
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
servicegroupname	<String>	Read-write	The name of the SSL service name to which the cipher-suite is to be bound. Minimum length = 1
servicegroup	<Boolean>	Read-write	Indicates that the cipher operation is to be performed on the named SSL service or service group.
service	<Boolean>	Read-write	Indicates that the cipher operation is to be performed on the named SSL service or service group.
servicename	<String>	Read-write	The name of the SSL service name to which the cipher-suite is to be bound. Minimum length = 1
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD

Operations (click to see Properties)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcipher_sslciphersuite_binding

May 20, 2016

Binding object showing the sslciphersuite that can be bound to sslcipher.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	Cipher name.
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
cipherpriority	<Double>	Read-write	This indicates priority assigned to the particular cipher. Minimum value = 1
description	<String>	Read-write	Cipher suite description.
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslcipher_sslciphersuite_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslcipher_sslciphersuite_binding":{
    "ciphergroupname":<String_value>,
    "cipheroperation":<String_value>,
    "ciphgrpals":<String_value>,
    "ciphername":<String_value>,
    "cipherpriority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher_sslciphersuite_binding/ciphergroupname_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher_sslciphersuite_binding/ciphergroupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcipher_sslciphersuite_binding/ciphergroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcipher_sslciphersuite_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcipher_sslciphersuite_binding/ciphergroupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcipher_sslciphersuite_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcipher_sslciphersuite_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ciphername":<String_value>,
    "ciphgrpals":<String_value>,
    "cipherpriority":<Double_value>,
    "description":<String_value>,
    "ciphergroupname":<String_value>,
    "cipheroperation":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher_sslciphersuite_binding/ciphergroupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslcipher_sslciphersuite_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcipher_sslprofile_binding

May 20, 2016

Binding object showing the sslprofile that can be bound to sslcipher.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cipherpriority	<Double>	Read-write	Priority of the cipher to be added. Minimum value = 1 Maximum value = 1000
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
description	<String>	Read-write	Cipher suite description.
sslprofile	<String>	Read-write	Name of the profile to which cipher is attached.
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher_sslprofile_binding/ciphergroupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcipher_sslprofile_binding/ciphergroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcipher_sslprofile_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcipher_sslprofile_binding/ciphergroupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcipher_sslprofile_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcipher_sslprofile_binding": [ {
    "cipherpriority":<Double_value>,
    "ciphgrpals":<String_value>,
    "description":<String_value>,
    "sslprofile":<String_value>,
    "ciphergroupname":<String_value>,
    "cipheroperation":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcipher_sslprofile_binding/ciphergroupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslcipher_sslprofile_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcipher_sslvserver_binding

May 20, 2016

Binding object showing the sslvserver that can be bound to sslcipher.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cipherpriority	<Double>	Read-write	Priority of the cipher to be added. Minimum value = 1 Maximum value = 1000
ciphgrpals	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name. Minimum length = 1
vserver	<Boolean>	Read-write	Select the -vServer flag when the cipher operation is performed on an SSL virtual server. Note: By default the bind ssl cipher command internally assumes the flag of -vServer argument. Hence, while working with the SSL vserver, you need not specify this flag.
vservername	<String>	Read-write	The name of the SSL virtual server to which the cipher-suite is to be bound. Minimum length = 1
ciphergroupname	<String>	Read-write	Name of the user-defined cipher group. Minimum length = 1
cipheroperation	<String>	Read-write	The operation that is performed when adding the cipher-suite. Possible cipher operations are: ADD - Appends the given cipher-suite to the existing one configured for the virtual server. REM - Removes the given cipher-suite from the existing one configured for the virtual server. ORD - Overrides the current configured cipher-suite for the virtual server with the given cipher-suite. Default value: 0 Possible values = ADD, REM, ORD

Operations (click to see Properties)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcrl

May 20, 2016

Configuration for Certificate Revocation List resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
crlname	<String>	Read-write	Name for the Certificate Revocation List (CRL). Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the CRL is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my crl" or 'my crl'). Minimum length = 1
crlpath	<String>	Read-write	Path to the CRL file. /var/netscaler/ssl/ is the default path. Minimum length = 1
inform	<String>	Read-write	Input format of the CRL file. The two formats supported on the appliance are: PEM - Privacy Enhanced Mail. DER - Distinguished Encoding Rule. Default value: PEM Possible values = DER, PEM
refresh	<String>	Read-write	Set CRL auto refresh. Possible values = ENABLED, DISABLED
cacert	<String>	Read-write	CA certificate that has issued the CRL. Required if CRL Auto Refresh is selected. Install the CA certificate on the appliance before adding the CRL. Minimum length = 1
method	<String>	Read-write	Method for CRL refresh. If LDAP is selected, specify the method, CA certificate, base DN, port, and LDAP server name. If HTTP is selected, specify the CA certificate, method, URL, and port. Cannot be changed after a CRL is added. Possible values = HTTP, LDAP
server	<String>	Read-write	IP address of the LDAP server from which to fetch the CRLs. Minimum length = 1
url	<String>	Read-write	URL of the CRL distribution point.
port	<Integer>	Read-write	Port for the LDAP server. Minimum value = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
basedn	<String>	Read-write	Citrix recommends searching for the Base DN instead of the Issuer Name from the CA certificate, because the Issuer Name field might not exactly match the LDAP directory structure's DN. Minimum length = 1
scope	<String>	Read-write	Extent of the search operation on the LDAP server. Available settings function as follows: One - One level below Base DN. Base - Exactly the same level as Base DN. Default value: One Possible values = Base, One
interval	<String>	Read-write	CRL refresh interval. Use the NONE setting to unset this parameter. Possible values = MONTHLY, WEEKLY, DAILY, NONE
day	<Integer>	Read-write	Day on which to refresh the CRL, or, if the Interval parameter is not set, the number of days after which to refresh the CRL. If Interval is set to MONTHLY, specify the date. If Interval is set to WEEKLY, specify the day of the week (for example, Sun=0 and Sat=6). This parameter is not applicable if the Interval is set to DAILY. Minimum value = 0 Maximum value = 31
time	<String>	Read-write	Time, in hours (1-24) and minutes (1-60), at which to refresh the CRL.
binddn	<String>	Read-write	Bind distinguished name (DN) to be used to access the CRL object in the LDAP repository if access to the LDAP repository is restricted or anonymous access is not allowed. Minimum length = 1
password	<String>	Read-write	Password to access the CRL in the LDAP repository if access to the LDAP repository is restricted or anonymous access is not allowed. Minimum length = 1
binary	<String>	Read-write	Set the LDAP-based CRL retrieval mode to binary. Default value: NO Possible values = YES, NO
cacertfile	<String>	Read-write	Name of and, optionally, path to the CA certificate file. /nsconfig/ssl/ is the default path. Maximum length = 63
cakeyfile	<String>	Read-write	Name of and, optionally, path to the CA key file. /nsconfig/ssl/ is the default path. Maximum length = 63
indexfile	<String>	Read-write	Name of and, optionally, path to the file containing the serial numbers of all the certificates that are revoked. Revoked certificates are appended to the file. /nsconfig/ssl/ is the default path. Maximum length = 63
revoke	<String>	Read-write	Name of and, optionally, path to the certificate to be revoked. /nsconfig/ssl/ is the default path. Maximum length = 63

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
gencrl	<String>	Read-write	that have been revoked is obtained from the index file. /nsconfig/ssl/ is the default path. Maximum length = 63
flags	<Integer>	Read-only	CRL status flag.
lastupdatetime	<Integer>	Read-only	Last CRL refresh time.
version	<Integer>	Read-only	CRL version.
signaturealgo	<String>	Read-only	Signature algorithm.
issuer	<String>	Read-only	Issuer name.
lastupdate	<String>	Read-only	Last update time.
nextupdate	<String>	Read-only	Next update time.
daystoexpiration	<Integer>	Read-only	Number of days remaining for the CRL to expire.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | CREATE | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcrl":{
    "crlname":<String_value>,
    "crlpath":<String_value>,
    "inform":<String_value>,
    "refresh":<String_value>,
    "cacert":<String_value>,
    "method":<String_value>,
    "server":<String_value>,
    "url":<String_value>,
    "port":<Integer_value>,
    "basedn":<String_value>,
    "scope":<String_value>,
    "interval":<String_value>,
    "day":<Integer_value>,
    "time":<String_value>,
    "binddn":<String_value>,
    "password":<String_value>,
    "binary":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

create

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcrl":{
    "cacertfile":<String_value>,
    "cakeyfile":<String_value>,
    "indexfile":<String_value>,
    "revoke":<String_value>,
    "gencrl":<String_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl/crlname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl

HTTP Method: PUT

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcrl":{
    "crlname":<String_value>,
    "refresh":<String_value>,
    "cacert":<String_value>,
    "server":<String_value>,
    "method":<String_value>,
    "url":<String_value>,
    "port":<Integer_value>,
    "basedn":<String_value>,
    "scope":<String_value>,
    "interval":<String_value>,
    "day":<Integer_value>,
    "time":<String_value>,
    "binddn":<String_value>,
    "password":<String_value>,
    "binary":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcrl":{
    "crlname":<String_value>,
    "refresh":true,
    "cacert":true,
    "server":true,
    "method":true,
    "url":true,
    "port":true,
    "basedn":true,
    "scope":true,
    "interval":true,
    "day":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "time":true,
        "binddn":true,
        "password":true,
        "binary":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcrl?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcrl?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcrl resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcrl?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcrl?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcrl resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcrl": [ {
      "crlname":<String_value>,
      "crlpath":<String_value>,
      "inform":<String_value>,
      "cacert":<String_value>,
      "refresh":<String_value>,
      "scope":<String_value>,
      "server":<String_value>,
      "port":<Integer_value>,
      "url":<String_value>,
      "method":<String_value>,
      "basedn":<String_value>,
      "interval":<String_value>,
      "day":<Integer_value>,
      "time":<String_value>,
      "binddn":<String_value>,
      "password":<String_value>,
      "flags":<Integer_value>,
      "lastupdatetime":<Integer_value>,
      "version":<Integer_value>,
      "signaturealgo":<String_value>,
      "issuer":<String_value>,
      "lastupdate":<String_value>,
      "nextupdate":<String_value>,
      "binary":<String_value>,
      "daystoexpiration":<Integer_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl/crlname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcrl/crlname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcrl/crlname_value<String>?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcrl": [ {
    "crlname":<String_value>,
    "crlpath":<String_value>,
    "inform":<String_value>,
    "cacert":<String_value>,
    "refresh":<String_value>,
    "scope":<String_value>,
    "server":<String_value>,
    "port":<Integer_value>,
    "url":<String_value>,
    "method":<String_value>,
    "basedn":<String_value>,
    "interval":<String_value>,
    "day":<Integer_value>,
    "time":<String_value>,
    "binddn":<String_value>,
    "password":<String_value>,
    "flags":<Integer_value>,
    "lastupdatetime":<Integer_value>,
    "version":<Integer_value>,
    "signaturealgo":<String_value>,
    "issuer":<String_value>,
    "lastupdate":<String_value>,
    "nextupdate":<String_value>,
    "binary":<String_value>,
    "daystoexpiration":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "sslcrl": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcrlfile

May 20, 2016

Configuration for Imported crl files resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name to assign to the imported CRL file. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file'). Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL specifying the protocol, host, and path, including file name to the CRL file to be imported. For example, http://www.example.com/crl_file. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

IMPORT | DELETE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrlfile?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslcrlfile":{
    "name":<String_value>,
    "src":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrlfile

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrlfile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslcrlfile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcrlfile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslcrlfile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslcrlfile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcrlfile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcrlfile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "sslcrlfile": [ {
      "name":<String_value>,
      "src":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrlfile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcrlfile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcrl_binding

May 20, 2016

Binding object which returns the resources bound to sslcrl.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
crlname	<String>	Read-write	Name of the CRL for which to show detailed information. Minimum length = 1
sslcrl_serialnumber_binding	<sslcrl_serialnumber_binding[]>	Read-only	serialnumber that can be bound to sslcrl.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl_binding/crlname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "sslcrl_binding": [ {
      "crlname":<String_value>,
      "sslcrl_serialnumber_binding":<sslcrl_serialnumber_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslcrl_serialnumber_binding

May 20, 2016

Binding object showing the serialnumber that can be bound to sslcrl.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
number	<String>	Read-write	Certificate Serial number.
crlname	<String>	Read-write	Name of the CRL for which to show detailed information. Minimum length = 1
date	<String>	Read-only	Certificate Revocation date.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl_serialnumber_binding/crlname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslcrl_serialnumber_binding/crlname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslcrl_serialnumber_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslcrl_serialnumber_binding/crlname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslcrl_serialnumber_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslcrl_serialnumber_binding": [ {
    "number":<String_value>,
    "crlname":<String_value>,
    "date":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslcrl_serialnumber_binding/crlname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslcrl_serialnumber_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

ssldhfile

May 20, 2016

Configuration for dh imported file resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name to assign to the imported DH file. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file'). Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL specifying the protocol, host, and path, including file name, to the DH file to be imported. For example, http://www.example.com/dh_file. NOTE: The import fails if the file is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

IMPORT | DELETE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldhfile?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ssldhfile":{
    "name":<String_value>,
    "src":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldhfile

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldhfile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ssldhfile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/ssldhfile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of ssldhfile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/ssldhfile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/ssldhfile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the ssldhfile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "ssldhfile": [ {
      "name":<String_value>,
      "src":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldhfile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ssldhfile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ssldhparam

May 20, 2016

Configuration for dh Parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
dhfile	<String>	Read-write	Name of and, optionally, path to the DH key file. /nsconfig/ssl/ is the default path. Maximum length = 63
bits	<Double>	Read-write	Size, in bits, of the DH key being generated. Minimum value = 512 Maximum value = 2048
gen	<String>	Read-write	Random number required for generating the DH key. Required as part of the DH key generation algorithm. Default value: 2 Possible values = 2, 5

Operations (click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

create

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldhparam?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ssldhparam":{
    "dhfile":<String_value>,
    "bits":<Double_value>,
    "gen":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ssldsakey

May 20, 2016

Configuration for dsa key resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
keyfile	<String>	Read-write	Name for and, optionally, path to the DSA key file. /nsconfig/ssl/ is the default path. Maximum length = 63
bits	<Double>	Read-write	Size, in bits, of the DSA key. Minimum value = 512 Maximum value = 2048
keyform	<String>	Read-write	Format in which the DSA key file is stored on the appliance. Default value: PEM Possible values = DER, PEM
des	<Boolean>	Read-write	Encrypt the generated DSA key by using the DES algorithm. On the command line, you are prompted to enter the pass phrase (password) that will be used to encrypt the key.
des3	<Boolean>	Read-write	Encrypt the generated DSA key by using the Triple-DES algorithm. On the command line, you are prompted to enter the pass phrase (password) that will be used to encrypt the key.
password	<String>	Read-write	Pass phrase to use for encryption if DES or DES3 option is selected. Minimum length = 1 Maximum length = 31

Operations (click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

create

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldsakey?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ssldsakey":{
    "keyfile":<String_value>,
    "bits":<Double_value>,
    "keyform":<String_value>,
    "des":<Boolean_value>,
    "des3":<Boolean_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ssldtlsprofile

May 20, 2016

Configuration for DTLS profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the DTLS profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@),equals sign (=), and hyphen (-) characters. Cannot be changed after the profile is created. Minimum length = 1 Maximum length = 127
pmtudiscovery	<String>	Read-write	Source for the maximum record size value. If ENABLED, the value is taken from the PMTU table. If DISABLED, the value is taken from the profile. Default value: DISABLED Possible values = ENABLED, DISABLED
maxrecordsize	<Double>	Read-write	Maximum size of records that can be sent if PMTU is disabled. Default value: 1459 Minimum value = 250 Maximum value = 1459
maxretrytime	<Double>	Read-write	Wait for the specified time, in seconds, before resending the request. Default value: 3
helloverifyrequest	<String>	Read-write	Send a Hello Verify request to validate the client. Default value: DISABLED Possible values = ENABLED, DISABLED
terminatesession	<String>	Read-write	Terminate the session if the message authentication code (MAC) of the client and server do not match. Default value: DISABLED Possible values = ENABLED, DISABLED
maxpacketsize	<Double>	Read-write	Maximum number of packets to reassemble. This value helps protect against a fragmented packet attack. Default value: 120 Minimum value = 0 Maximum value = 86400
builtin	<String[]>	Read-only	Flag to determine whether dtls profile is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ssldtlsprofile":{
    "name":<String_value>,
    "pmtudiscovery":<String_value>,
    "maxrecordsize":<Double_value>,
    "maxretrytime":<Double_value>,
    "helloverifyrequest":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "terminatesession":<String_value>,
    "maxpacketsize":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ssldtlsprofile":{
    "name":<String_value>,
    "pmtudiscovery":<String_value>,
    "maxrecordsize":<Double_value>,
    "maxretrytime":<Double_value>,
    "helloverifyrequest":<String_value>,
    "terminatesession":<String_value>,
    "maxpacketsize":<Double_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"ssldtlsprofile":{
    "name":<String_value>,
    "pmtudiscovery":true,
    "maxrecordsize":true,
    "maxretrytime":true,
    "helloverifyrequest":true,
    "terminatesession":true,
    "maxpacketsize":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile?**filter=property-name1:property-val1,property-name2:property-val2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of ssldtlsprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the ssldtlsprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ssldtlsprofile": [ {
    "name":<String_value>,
    "pmtudiscovery":<String_value>,
    "maxrecordsize":<Double_value>,
    "maxretrytime":<Double_value>,
    "helloverifyrequest":<String_value>,
    "terminatesession":<String_value>,
    "maxpacketsize":<Double_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile/name_value<String>

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ssldtlsprofile": [ {
    "name":<String_value>,
    "pmtudiscovery":<String_value>,
    "maxrecordsize":<Double_value>,
    "maxretrytime":<Double_value>,
    "helloverifyrequest":<String_value>,
    "terminatesession":<String_value>,
    "maxpacketsize":<Double_value>,
    "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/ssldtlsprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "sslcdtlsprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslfips

May 20, 2016

Configuration for fips resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
inithsm	<String>	Read-write	FIPS initialization level. The appliance currently supports Level-2 (FIPS 140-2). Possible values = Level-2
sopassword	<String>	Read-write	Security officer password that will be in effect after you have configured the HSM. Minimum length = 1
oldsopassword	<String>	Read-write	Old password for the security officer. Minimum length = 1
userpassword	<String>	Read-write	The Hardware Security Module's (HSM) User password. Minimum length = 1
hsmlabel	<String>	Read-write	Label to identify the Hardware Security Module (HSM). Minimum length = 1
fipsfw	<String>	Read-write	Path to the FIPS firmware file. Minimum length = 1
erasedata	<String>	Read-only	Erase data. Default value: FIPS_ERASE Minimum length = 1
serial	<Integer>	Read-only	FIPS card serial number.
majorversion	<Integer>	Read-only	Firmware major version.
minorversion	<Integer>	Read-only	Firmware minor version.
fipshwmajorversion	<Integer>	Read-only	FIPS card hardware major version.
fipshwminorversion	<Integer>	Read-only	FIPS card hardware minor version.
fipshwversionstring	<String>	Read-only	FIPS card hardware extended version string.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
flashmemorytotal	<Integer>	Read-only	Total size of the flash memory on card.
flashmemoryfree	<Integer>	Read-only	Total size of free flash memory.
sramtotal	<Integer>	Read-only	Total size of the SRAM memory on card.
sramfree	<Integer>	Read-only	Total size of free SRAM memory.
status	<Integer>	Read-only	Status.
flag	<Integer>	Read-only	Internal Flags.
serialno	<String>	Read-only	FIPS card serial number.
model	<String>	Read-only	FIPS card model info.
state	<Integer>	Read-only	FIPS card state.
firmwarereleasedate	<String>	Read-only	FIPS card firmware revision date.
coresmax	<Integer>	Read-only	Maximum number of crypto cores present in the FIPS card.
coresenabled	<Integer>	Read-only	Number of crypto cores enabled in the FIPS card.

Operations (click to see Properties)

UPDATE | UNSET | RESET | CHANGE | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfips

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslfips":{
    "inithsm":<String_value>,
    "sopassword":<String_value>,
    "oldsopassword":<String_value>,
    "userpassword":<String_value>,
    "hsmlabel":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfips?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"sslfips":{
    "hsmlabel":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

reset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfips?**action=reset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslfips":{
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

change

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfips?**action=update**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"sslfips":{
    "fipsfw":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfips

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslfips": [ {
    "inithsm":<String_value>,
    "sopassword":<String_value>,
    "userpassword":<String_value>,
    "oldsopassword":<String_value>,
    "erasedata":<String_value>,
    "hsmlabel":<String_value>,
    "serial":<Integer_value>,
    "majorversion":<Integer_value>,
    "minorversion":<Integer_value>,
    "fipshwmajorversion":<Integer_value>,
    "fipshwminorversion":<Integer_value>,
    "fipshwversionstring":<String_value>,
    "flashmemorytotal":<Integer_value>,
    "flashmemoryfree":<Integer_value>,
    "sramtotal":<Integer_value>,
    "sramfree":<Integer_value>,
    "status":<Integer_value>,
    "flag":<Integer_value>,
    "serialno":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "model":<String_value>,
    "state":<Integer_value>,
    "firmwarereleasedate":<String_value>,
    "coresmax":<Integer_value>,
    "coresenabled":<Integer_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslfipskey

May 20, 2016

Configuration for FIPS key resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
fipskeyname	<String>	Read-write	Name for the FIPS key. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the FIPS key is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my fipskey" or 'my fipskey'). Minimum length = 1
modulus	<Double>	Read-write	Modulus, in multiples of 64, of the FIPS key to be created. Minimum value = 0 Maximum value = 4096
exponent	<String>	Read-write	Exponent value for the FIPS key to be created. Available values function as follows: 3=3 (hexadecimal) F4=10001 (hexadecimal). Default value: 3 Possible values = 3, F4
key	<String>	Read-write	Name of and, optionally, path to the key file to be imported. /nsconfig/ssl/ is the default path. Minimum length = 1
inform	<String>	Read-write	Input format of the key file. Available formats are: SIM - Secure Information Management; select when importing a FIPS key. If the external FIPS key is encrypted, first decrypt it, and then import it. PEM - Privacy Enhanced Mail; select when importing a non-FIPS key. Default value: SIM Possible values = SIM, DER, PEM
wrapkeyname	<String>	Read-write	Name of the wrap key to use for importing the key. Required for importing a non-FIPS key. Minimum length = 1
iv	<String>	Read-write	Initialization Vector (IV) to use for importing the key. Required for importing a non-FIPS key. Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

size	<Double>	Read-only	Size.
Name	**Data Type**	**Permissions**	**Description**
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

CREATE | DELETE | IMPORT | EXPORT | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

create

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipskey?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"sslfipskey":{
    "fipskeyname":<String_value>,
    "modulus":<Double_value>,
    "exponent":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipskey/fipskeyname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipskey?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslfipskey":{
    "fipskeyname":<String_value>,
    "key":<String_value>,
    "inform":<String_value>,
    "wrapkeyname":<String_value>,
    "iv":<String_value>,
    "exponent":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

export

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipskey?**action=export**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslfipskey":{
    "fipskeyname":<String_value>,
    "key":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipskey

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslfipskey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslfipskey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslfipskey resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/sslfipskey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslfipskey?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslfipskey resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslfipskey": [ {
    "fipskeyname":<String_value>,
    "modulus":<Double_value>,
    "exponent":<String_value>,
    "size":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipskey/fipskeyname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslfipskey/fipskeyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/sslfipskey/fipskeyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslfipskey": [ {
    "fipskeyname":<String_value>,
    "modulus":<Double_value>,
    "exponent":<String_value>,
    "size":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipskey?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "sslfipskey": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslhsmkey

May 20, 2016

Configuration for HSM key resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
hsmkeyname	<String>	Read-write	. Minimum length = 1
key	<String>	Read-write	Name of and, optionally, path to the HSM key file. /var/opt/nfast/kmdata/local/ is the default path. Maximum length = 63
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslhsmkey":{
    "hsmkeyname":<String_value>,
    "key":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey/hsmkeyname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslhsmkey resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslhsmkey resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslhsmkey": [ {
    "hsmkeyname":<String_value>,

}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey/hsmkeyname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey/hsmkeyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey/hsmkeyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslhsmkey": [ {
    "hsmkeyname":<String_value>,

}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslhsmkey?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "sslhsmkey": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslfipssimsource

May 20, 2016

Configuration for FIPsSIM source resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
targetsecret	<String>	Read-write	Name of and, optionally, path to the target FIPS appliance's secret data. /nsconfig/ssl/ is the default path. Minimum length = 1
sourcesecret	<String>	Read-write	Name for and, optionally, path to the source FIPS appliance's secret data. /nsconfig/ssl/ is the default path. Minimum length = 1
certfile	<String>	Read-write	Name for and, optionally, path to the source FIPS appliance's certificate file. /nsconfig/ssl/ is the default path. Minimum length = 1

Operations (click to see Properties)

ENABLE | INIT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipssimsource?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslfipssimsource":{
    "targetsecret":<String_value>,
    "sourcesecret":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

init

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipssimsource?**action=init**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslfipssimsource":{
    "certfile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslfipssimtarget

May 20, 2016

Configuration for FIPS SIM Target resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
keyvector	<String>	Read-write	Name of and, optionally, path to the target FIPS appliance's key vector. /nsconfig/ssl/ is the default path. Minimum length = 1
sourcesecret	<String>	Read-write	Name of and, optionally, path to the source FIPS appliance's secret data. /nsconfig/ssl/ is the default path. Minimum length = 1
certfile	<String>	Read-write	Name of and, optionally, path to the source FIPS appliance's certificate file. /nsconfig/ssl/ is the default path. Minimum length = 1
targetsecret	<String>	Read-write	Name for and, optionally, path to the target FIPS appliance's secret data. The default input path for the secret data is /nsconfig/ssl/. Minimum length = 1

Operations (click to see Properties)

ENABLE | INIT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipssimtarget?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslfipssimtarget":{
    "keyvector":<String_value>,
    "sourcesecret":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

init

URL: http://<netscaler-ip-address>/nitro/v1/config/sslfipssimtarget?**action=init**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslfipssimtarget":{
    "certfile":<String_value>,
    "keyvector":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "targetsecret":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslglobal_binding

May 20, 2016

Binding object which returns the resources bound to sslglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sslglobal_sslpolicy_binding	<sslglobal_sslpolicy_binding[]>	Read-only	sslpolicy that can be bound to sslglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/sslglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "sslglobal_binding": [ {
      "sslglobal_sslpolicy_binding":<sslglobal_sslpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslkeyfile

Jul 07, 2017

Configuration for Imported ssl key files resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name to assign to the imported key file. Must begin with an ASCII alphanumeric or underscore(_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@),equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my file" or 'my file'). Minimum length = 1 Maximum length = 31
src	<String>	Read-write	URL specifying the protocol, host, and path, including file name, to the key file to be imported. For example, http://www.example.com/key_file. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047
password	<String>	Read-write	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

IMPORT | DELETE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/sslkeyfile?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslkeyfile":{
    "name":<String_value>,
    "src":<String_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslkeyfile

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslkeyfile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslkeyfile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslkeyfile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslkeyfile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslkeyfile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslkeyfile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslkeyfile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "sslkeyfile": [ {
      "name":<String_value>,
      "src":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslkeyfile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslkeyfile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslocspresponder

May 20, 2016

Configuration for OCSP responser resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the OCSP responder. Cannot begin with a hash (#) or space character and must contain only ASCII alphanumeric, underscore (_), hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the responder is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my responder" or 'my responder'). Minimum length = 1
url	<String>	Read-write	URL of the OCSP responder. Minimum length = 1
cache	<String>	Read-write	Enable caching of responses. Caching of responses received from the OCSP responder enables faster responses to the clients and reduces the load on the OCSP responder. Possible values = ENABLED, DISABLED
cachetimeout	<Double>	Read-write	Timeout for caching the OCSP response. After the timeout, the NetScaler sends a fresh request to the OCSP responder for the certificate status. If a timeout is not specified, the timeout provided in the OCSP response applies. Default value: 1 Minimum value = 1 Maximum value = 1440
batchingdepth	<Double>	Read-write	Number of client certificates to batch together into one OCSP request. Batching avoids overloading the OCSP responder. A value of 1 signifies that each request is queried independently. For a value greater than 1, specify a timeout (batching delay) to avoid inordinately delaying the processing of a single certificate. Minimum value = 1 Maximum value = 8
batchingdelay	<Double>	Read-write	Maximum time, in milliseconds, to wait to accumulate OCSP requests to batch. Does not apply if the Batching Depth is 1. Minimum value = 0 Maximum value = 10000
			Time, in milliseconds, to wait for an OCSP response. When this time elapses, an error message appears or the transaction is forwarded, depending on the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 0 Maximum value = 120000
respondercert	<String>	Read-write	. Minimum length = 1
trustresponder	<Boolean>	Read-write	A certificate to use to validate OCSP responses. Alternatively, if -trustResponder is specified, no verification will be done on the reponse. If both are omitted, only the response times (producedAt, lastUpdate, nextUpdate) will be verified.
producedattimeskew	<Double>	Read-write	Time, in seconds, for which the NetScaler waits before considering the response as invalid. The response is considered invalid if the Produced At time stamp in the OCSP response exceeds or precedes the current NetScaler clock time by the amount of time specified. Default value: 300 Minimum value = 0 Maximum value = 86400
signingcert	<String>	Read-write	Certificate-key pair that is used to sign OCSP requests. If this parameter is not set, the requests are not signed. Minimum length = 1
usenonce	<String>	Read-write	Enable the OCSP nonce extension, which is designed to prevent replay attacks. Possible values = YES, NO
insertclientcert	<String>	Read-write	Include the complete client certificate in the OCSP request. Possible values = YES, NO
useaia	<Boolean>	Read-only	Only use the URL present in the certificate.
dns	<String>	Read-only	Was DNS resolution successful for a domain-based OCSP responder.
ipaddress	<String>	Read-only	The IPv6 address of the ocsp responder.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslocspresponder":{
    "name":<String_value>,
    "url":<String_value>,
    "cache":<String_value>,
    "cachetimeout":<Double_value>,
    "batchingdepth":<Double_value>,
    "batchingdelay":<Double_value>,
    "resptimeout":<Double_value>,
    "respondercert":<String_value>,
    "trustresponder":<Boolean_value>,
    "producedattimeskew":<Double_value>,
    "signingcert":<String_value>,
    "usenonce":<String_value>,
    "insertclientcert":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

update

URL: http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslocspresponder":{
    "name":<String_value>,
    "url":<String_value>,
    "cache":<String_value>,
    "cachetimeout":<Double_value>,
    "batchingdepth":<Double_value>,
    "batchingdelay":<Double_value>,
    "resptimeout":<Double_value>,
    "respondercert":<String_value>,
    "trustresponder":<Boolean_value>,
    "producedattimeskew":<Double_value>,
    "signingcert":<String_value>,
    "usenonce":<String_value>,
    "insertclientcert":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslocspresponder":{
    "name":<String_value>,
    "trustresponder":true,
    "insertclientcert":true,
    "cache":true,
    "cachetimeout":true,
    "batchingdepth":true,
    "batchingdelay":true,
    "resptimeout":true,
    "respondercert":true,
    "producedattimeskew":true,
    "signingcert":true,
    "usenonce":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslocspresponder resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslocspresponder resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslocspresponder": [ {
    "name":<String_value>,
    "url":<String_value>,
    "useaia":<Boolean_value>,
    "cache":<String_value>,
    "cachetimeout":<Double_value>,
    "batchingdepth":<Double_value>,
    "batchingdelay":<Double_value>,
    "resptimeout":<Double_value>,
    "producedattimeskew":<Double_value>,
    "respondercert":<String_value>,
    "trustresponder":<Boolean_value>,
    "signingcert":<String_value>,
    "usenonce":<String_value>,
    "dns":<String_value>,
    "insertclientcert":<String_value>,
    "ipaddress":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslocspresponder": [ {
      "name":<String_value>,
      "url":<String_value>,
      "useaia":<Boolean_value>,
      "cache":<String_value>,
      "cachetimeout":<Double_value>,
      "batchingdepth":<Double_value>,
      "batchingdelay":<Double_value>,
      "resptimeout":<Double_value>,
      "producedattimeskew":<Double_value>,
      "respondercert":<String_value>,
      "trustresponder":<Boolean_value>,
      "signingcert":<String_value>,
      "usenonce":<String_value>,
      "dns":<String_value>,
      "insertclientcert":<String_value>,
      "ipaddress":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslocspresponder?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslocspresponder": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslparameter

May 20, 2016

Configuration for SSL parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
quantumsize	<String>	Read-write	Amount of data to collect before the data is pushed to the crypto hardware for encryption. For large downloads, a larger quantum size better utilizes the crypto resources. Default value: 8192 Possible values = 4096, 8192, 16384
crlmemorysizemb	<Double>	Read-write	Maximum memory size to use for certificate revocation lists (CRLs). This parameter reserves memory for a CRL but sets a limit to the maximum memory that the CRLs loaded on the appliance can consume. Default value: 256 Minimum value = 10 Maximum value = 1024
strictcachecks	<String>	Read-write	Enable strict CA certificate checks on the appliance. Default value: NO Possible values = YES, NO
ssltriggertimeout	<Double>	Read-write	Time, in milliseconds, after which encryption is triggered for transactions that are not tracked on the NetScaler appliance because their length is not known. There can be a delay of up to 10ms from the specified timeout value before the packet is pushed into the queue. Default value: 100 Minimum value = 1 Maximum value = 200
sendclosenotify	<String>	Read-write	Send an SSL Close-Notify message to the client at the end of a transaction. Default value: YES Possible values = YES, NO
encrypttriggerpktcount	<Double>	Read-write	Maximum number of queued packets after which encryption is triggered. Use this setting for SSL transactions that send small packets from server to NetScaler. Default value: 45 Minimum value = 10 Maximum value = 50
			Deny renegotiation in specified circumstances. Available settings function as follows: * NO - Allow SSL renegotiation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
denysslreneg	<String>	Read-write	initiated by the client. * FRONTEND_CLIENTSERVER - Deny secure and nonsecure SSL renegotiation initiated by the client or the NetScaler during policy-based client authentication. * ALL - Deny all secure and nonsecure SSL renegotiation. * NONSECURE - Deny nonsecure SSL renegotiation. Allows only clients that support RFC 5746. Default value: ALL Possible values = NO, FRONTEND_CLIENT, FRONTEND_CLIENTSERVER, ALL, NONSECURE
insertionencoding	<String>	Read-write	Encoding method used to insert the subject or issuer's name in HTTP requests to servers. Default value: Unicode Possible values = Unicode, UTF-8
ocspcachesize	<Double>	Read-write	Size, per packet engine, in megabytes, of the OCSP cache. A maximum of 10% of the packet engine memory can be assigned. Because the maximum allowed packet engine memory is 4GB, the maximum value that can be assigned to the OCSP cache is approximately 410 MB. Default value: 10 Minimum value = 0 Maximum value = 512
pushflag	<Double>	Read-write	Insert PUSH flag into decrypted, encrypted, or all records. If the PUSH flag is set to a value other than 0, the buffered records are forwarded on the basis of the value of the PUSH flag. Available settings function as follows: 0 - Auto (PUSH flag is not set.) 1 - Insert PUSH flag into every decrypted record. 2 -Insert PUSH flag into every encrypted record. 3 - Insert PUSH flag into every decrypted and encrypted record. Minimum value = 0 Maximum value = 3
dropreqwithnohostheader	<String>	Read-write	Host header check for SNI enabled sessions. If this check is enabled and the HTTP request does not contain the host header for SNI enabled sessions, the request is dropped. Default value: NO Possible values = YES, NO
pushenctriggertimeout	<Double>	Read-write	PUSH encryption trigger timeout value. The timeout value is applied only if you set the Push Encryption Trigger parameter to Timer in the SSL virtual server settings. Default value: 1 Minimum value = 1 Maximum value = 200
cryptodevdisablelimit	<Double>	Read-write	Limit to the number of disabled SSL chips after which the ADC restarts. A value of zero implies that the ADC does not automatically restart. Default value: 0
undefactioncontrol	<String>	Read-write	Name of the undefined built-in control action: CLIENTAUTH, NOCLIENTAUTH, NOOP, RESET, or DROP. Default value: "CLIENTAUTH"

© 1999-2017 Citrix Systems, Inc. All rights reserved.

undefactiondata	<String>	Read-write	Name of the undefined built-in data action: NOOP, RESET or DROP. Default value: "NOOP"
Name	**Data Type**	**Permissions**	**Description**
defaultprofile	<String>	Read-write	Global parameter used to enable default profile feature. Default value: DISABLED Possible values = ENABLED, DISABLED
svctls1112disable	<String>	Read-only	Disable TLS 1.1 and 1.2 for dynamic and VPN created services. Default value: DISABLED Possible values = ENABLED, DISABLED
montls1112disable	<String>	Read-only	Disable TLS 1.1 and 1.2 for secure (https) monitors bound to SSL_BRIDGE services. Default value: DISABLED Possible values = ENABLED, DISABLED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/sslparameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslparameter":{
    "quantumsize":<String_value>,
    "crlmemorysizemb":<Double_value>,
    "strictcachecks":<String_value>,
    "ssltriggertimeout":<Double_value>,
    "sendclosenotify":<String_value>,
    "encrypttriggerpktcount":<Double_value>,
    "denysslreneg":<String_value>,
    "insertionencoding":<String_value>,
    "ocspcachesize":<Double_value>,
    "pushflag":<Double_value>,
    "dropreqwithnohostheader":<String_value>,
    "pushenctriggertimeout":<Double_value>,
    "cryptodevdisablelimit":<Double_value>,
    "undefactioncontrol":<String_value>,
    "undefactiondata":<String_value>,
    "defaultprofile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslparameter":{
    "quantumsize":true,
    "crlmemorysizemb":true,
    "strictcachecks":true,
    "ssltriggertimeout":true,
    "sendclosenotify":true,
    "encrypttriggerpktcount":true,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "denysslreneg":true,
    "insertionencoding":true,
    "ocspcachesize":true,
    "pushflag":true,
    "dropreqwithnohostheader":true,
    "pushenctriggertimeout":true,
    "cryptodevdisablelimit":true,
    "undefactioncontrol":true,
    "undefactiondata":true,
    "defaultprofile":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslparameter": [ {
    "quantumsize":<String_value>,
    "crlmemorysizemb":<Double_value>,
    "strictcachecks":<String_value>,
    "ssltriggertimeout":<Double_value>,
    "sendclosenotify":<String_value>,
    "encrypttriggerpktcount":<Double_value>,
    "denysslreneg":<String_value>,
    "insertionencoding":<String_value>,
    "ocspcachesize":<Double_value>,
    "pushflag":<Double_value>,
    "dropreqwithnohostheader":<String_value>,
    "pushenctriggertimeout":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cryptodevdisablelimit":<Double_value>,
    "undefactioncontrol":<String_value>,
    "undefactiondata":<String_value>,
    "defaultprofile":<String_value>,
    "svctls1112disable":<String_value>,
    "montls1112disable":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpkcs12

May 20, 2016

Configuration for pkcs12 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
outfile	<String>	Read-write	Name for and, optionally, path to, the output file that contains the certificate and the private key after converting from PKCS#12 to PEM format. /nsconfig/ssl/ is the default path. If importing, the certificate-key pair is stored in PEM format. If exporting, the certificate-key pair is stored in PKCS#12 format. Maximum length = 63
Import	<Boolean>	Read-write	Convert the certificate and private-key from PKCS#12 format to PEM format.
pkcs12file	<String>	Read-write	Name for and, optionally, path to, the PKCS#12 file. If importing, specify the input file name that contains the certificate and the private key in PKCS#12 format. If exporting, specify the output file name that contains the certificate and the private key after converting from PEM to PKCS#12 format. /nsconfig/ssl/ is the default path. During the import operation, if the key is encrypted, you are prompted to enter the pass phrase used for encrypting the key. Maximum length = 63
des	<Boolean>	Read-write	Encrypt the private key by using the DES algorithm in CBC mode during the import operation. On the command line, you are prompted to enter the pass phrase.
des3	<Boolean>	Read-write	Encrypt the private key by using the Triple-DES algorithm in EDE CBC mode (168-bit key) during the import operation. On the command line, you are prompted to enter the pass phrase.
export	<Boolean>	Read-write	Convert the certificate and private key from PEM format to PKCS#12 format. On the command line, you are prompted to enter the pass phrase.
certfile	<String>	Read-write	Certificate file to be converted from PEM to PKCS#12 format. Maximum length = 63
keyfile	<String>	Read-write	Name of the private key file to be converted from PEM to PKCS#12 format. If the key file is encrypted, you are prompted to enter the pass phrase used for encrypting the key. Maximum length = 63
password	<String>	Read-write	. Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
pempassphrase	<string>	Read-write	Minimum length = 1 Maximum length = 31

Operations (click to see Properties)

CONVERT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

convert

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpkcs12?**action=convert**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslpkcs12":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "outfile":<String_value>,
    "Import":<Boolean_value>,
    "pkcs12file":<String_value>,
    "des":<Boolean_value>,
    "des3":<Boolean_value>,
    "export":<Boolean_value>,
    "certfile":<String_value>,
    "keyfile":<String_value>,
    "password":<String_value>,
    "pempassphrase":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpkcs8

May 20, 2016

Configuration for pkcs8 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
pkcs8file	<String>	Read-write	Name for and, optionally, path to, the output file where the PKCS#8 format key file is stored. /nsconfig/ssl/ is the default path. Maximum length = 63
keyfile	<String>	Read-write	Name of and, optionally, path to the input key file to be converted from PEM or DER format to PKCS#8 format. /nsconfig/ssl/ is the default path. Maximum length = 63
keyform	<String>	Read-write	Format in which the key file is stored on the appliance. Default value: PEM Possible values = DER, PEM
password	<String>	Read-write	Password to assign to the file if the key is encrypted. Applies only for PEM format files. Minimum length = 1 Maximum length = 31

Operations (click to see Properties)

CONVERT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

convert

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpkcs8?**action=convert**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslpkcs8":{
    "pkcs8file":<String_value>,
    "keyfile":<String_value>,
    "keyform":<String_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicy

Jul 07, 2017

Configuration for SSL policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the new SSL policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression, against which traffic is evaluated. Written in the classic or default syntax. Note: Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' (Classic expressions are not supported in the cluster build.) The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
reqaction	<String>	Read-write	The name of the action to be performed on the request. Refer to 'add ssl action' command to add a new action. Builtin actions like NOOP, RESET, DROP, CLIENTAUTH and NOCLIENTAUTH are also allowed. Minimum length = 1
action	<String>	Read-write	Name of the built-in or user-defined action to perform on the request. Available built-in actions are NOOP, RESET, DROP, CLIENTAUTH, and NOCLIENTAUTH.
undefaction	<String>	Read-write	Name of the action to be performed when the result of rule evaluation is undefined. Possible values for control policies: CLIENTAUTH, NOCLIENTAUTH, NOOP, RESET, DROP. Possible values for data policies: NOOP, RESET or DROP.
comment	<String>	Read-write	Any comments associated with this policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
hits	<Double>	Read-only	Number of hits for this policy.
undefhits	<Double>	Read-only	Number of Undef hits.
description	<String>	Read-only	Description of the policy.
policytype	<String>	Read-only	. Possible values = Classic Policy, Advanced Policy
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "reqaction":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"sslpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslpolicy":{
    "name":<String_value>,
    "undefaction":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy

Query-parameters:

attrs

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "undefaction":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "description":<String_value>,
    "comment":<String_value>,
    "policytype":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "undefaction":<String_value>,
      "hits":<Double_value>,
      "undefhits":<Double_value>,
      "description":<String_value>,
      "comment":<String_value>,
      "policytype":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicylabel

May 20, 2016

Configuration for SSL policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name for the SSL policy label. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy label is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my label" or 'my label').
type	<String>	Read-write	Type of policies that the policy label can contain. Possible values = CONTROL, DATA
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
policyname	<String>	Read-only	Name of the SSL policy to bind to the policy label.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-only	Invoke flag.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
invoke_labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
flowtype	<Double>	Read-only	Flowtype of the bound SSL policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	**Data Type**	**Permissions**	**Description**
ccount	<Double>	Read-only	Count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslpolicylabel":{
    "labelname":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "type":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslpolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslpolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicylabel": [ {
    "labelname":<String_value>,
    "type":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "flowtype":<Double_value>,
    "description":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicylabel": [ {
      "labelname":<String_value>,
      "type":<String_value>,
      "numpol":<Double_value>,
      "hits":<Double_value>,
      "policyname":<String_value>,
      "priority":<Double_value>,
      "gotoprioritiyexpression":<String_value>,
      "invoke":<Boolean_value>,
      "labeltype":<String_value>,
      "invoke_labelname":<String_value>,
      "flowtype":<Double_value>,
      "description":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "sslpolicylabel": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservicegroup_sslciphersuite_binding

May 20, 2016

Binding object showing the sslciphersuite that can be bound to sslservicegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	The name of the cipher group/alias/name configured for the SSL service group.
servicegroupname	<String>	Read-write	The name of the SSL service to which the SSL policy needs to be bound. Minimum length = 1
description	<String>	Read-write	The description of the cipher.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslservicegroup_sslciphersuite_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslservicegroup_sslciphersuite_binding":{
    "servicegroupname":<String_value>,
    "ciphername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservicegroup_sslciphersuite_binding resources configured on

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservicegroup_sslciphersuite_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservicegroup_sslciphersuite_binding": [ {
    "ciphername":<String_value>,
    "servicegroupname":<String_value>,
    "description":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslciphersuite_binding/servicegroupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{"sslservicegroup_sslciphersuite_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicylabel_binding

Jul 07, 2017

Binding object which returns the resources bound to sslpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the SSL policy label for which to show detailed information.
sslpolicylabel_sslpolicy_binding	<sslpolicylabel_sslpolicy_binding[]>	Read-only	sslpolicy that can be bound to sslpolicylabel.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "sslpolicylabel_binding": [ {
      "labelname":<String_value>,
      "sslpolicylabel_sslpolicy_binding":<sslpolicylabel_sslpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicylabel_sslpolicy_binding

Jul 07, 2017

Binding object showing the sslpolicy that can be bound to sslpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the SSL policy to bind to the policy label.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = vserver, service, policylabel
labelname	<String>	Read-write	Name of the SSL policy label to which to bind policies.
invoke_labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke policies bound to a policy label. After the invoked policies are evaluated, the flow returns to the policy with the next priority.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslpolicylabel_sslpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslpolicylabel_sslpolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicylabel_sslpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslpolicylabel_sslpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "sslpolicylabel_sslpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicylabel_sslpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslpolicylabel_sslpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to sslpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
sslpolicy_sslservice_binding	<sslpolicy_sslservice_binding[]>	Read-only	sslservice that can be bound to sslpolicy.
sslpolicy_lbvserver_binding	<sslpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to sslpolicy.
sslpolicy_sslvserver_binding	<sslpolicy_sslvserver_binding[]>	Read-only	sslvserver that can be bound to sslpolicy.
sslpolicy_csvserver_binding	<sslpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to sslpolicy.
sslpolicy_sslpolicylabel_binding	<sslpolicy_sslpolicylabel_binding[]>	Read-only	sslpolicylabel that can be bound to sslpolicy.
sslpolicy_sslglobal_binding	<sslpolicy_sslglobal_binding[]>	Read-only	sslglobal that can be bound to sslpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "sslpolicy_binding": [ {
      "name":<String_value>,
      "sslpolicy_sslvserver_binding":<sslpolicy_sslvserver_binding[]_value>,
      "sslpolicy_csvserver_binding":<sslpolicy_csvserver_binding[]_value>,
      "sslpolicy_lbvserver_binding":<sslpolicy_lbvserver_binding[]_value>,
      "sslpolicy_sslservice_binding":<sslpolicy_sslservice_binding[]_value>,
      "sslpolicy_sslpolicylabel_binding":<sslpolicy_sslpolicylabel_binding[]_value>,
      "sslpolicy_sslglobal_binding":<sslpolicy_sslglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to sslpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicy_csvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to sslpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicy_lbvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicy_sslglobal_binding

May 20, 2016

Binding object showing the sslglobal that can be bound to sslpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_sslglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslpolicy_sslglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicy_sslglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslpolicy_sslglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicy_sslpolicylabel_binding

May 20, 2016

Binding object showing the sslpolicylabel that can be bound to sslpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_sslpolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslpolicylabel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslpolicy_sslpolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicy_sslpolicylabel_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslpolicy_sslpolicylabel_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicy_sslservice_binding

May 20, 2016

Binding object showing the sslservice that can be bound to sslpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslservice_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslservice_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_sslservice_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslservice_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslpolicy_sslservice_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicy_sslservice_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslservice_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslpolicy_sslservice_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslpolicy_sslvserver_binding

May 20, 2016

Binding object showing the sslvserver that can be bound to sslpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the SSL policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = vserver, service, policylabel
activepolicy	<Double>	Read-only	.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslpolicy_sslvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslpolicy_sslvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslpolicy_sslvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Double_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslpolicy_sslvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslpolicy_sslvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslprofile

May 20, 2016

Configuration for SSL profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the SSL profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the profile is created. Minimum length = 1 Maximum length = 127
sslprofiletype	<String>	Read-write	Type of profile. Front end profiles apply to the entity that receives requests from a client. Backend profiles apply to the entity that sends client requests to a server. Default value: FrontEnd Possible values = BackEnd, FrontEnd
dhcount	<Double>	Read-write	Number of interactions, between the client and the NetScaler appliance, after which the DH private-public pair is regenerated. A value of zero (0) specifies infinite use (no refresh). This parameter is not applicable when configuring a backend profile. Minimum value = 0 Maximum value = 65534
dh	<String>	Read-write	State of Diffie-Hellman (DH) key exchange. This parameter is not applicable when configuring a backend profile. Default value: DISABLED Possible values = ENABLED, DISABLED
dhfile	<String>	Read-write	The file name and path for the DH parameter. Minimum length = 1
ersa	<String>	Read-write	State of Ephemeral RSA (eRSA) key exchange. Ephemeral RSA allows clients that support only export ciphers to communicate with the secure server even if the server certificate does not support export clients. The ephemeral RSA key is automatically generated when you bind an export cipher to an SSL or TCP-based SSL virtual server or service. When you remove the export cipher, the eRSA key is not deleted. It is reused at a later date when another export cipher is bound to an SSL or TCP-based SSL virtual server or service. The eRSA key is deleted when the appliance restarts. This parameter is not applicable when configuring a backend profile. Default value: ENABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ersacount	<Double>	Read-write	The refresh count for the re-generation of RSA public-key and private-key pair. Minimum value = 0 Maximum value = 65534
sessreuse	<String>	Read-write	State of session reuse. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the ENABLED setting, session key exchange is avoided for session resumption requests received from the client. Default value: ENABLED Possible values = ENABLED, DISABLED
sesstimeout	<Double>	Read-write	The Session timeout value in seconds. Default value: 120 Minimum value = 0 Maximum value = 4294967294
cipherredirect	<String>	Read-write	State of Cipher Redirect. If this parameter is set to ENABLED, you can configure an SSL virtual server or service to display meaningful error messages if the SSL handshake fails because of a cipher mismatch between the virtual server or service and the client. This parameter is not applicable when configuring a backend profile. Default value: DISABLED Possible values = ENABLED, DISABLED
cipherurl	<String>	Read-write	The redirect URL to be used with the Cipher Redirect feature.
clientauth	<String>	Read-write	State of client authentication. In service-based SSL offload, the service terminates the SSL handshake if the SSL client does not provide a valid certificate. This parameter is not applicable when configuring a backend profile. Default value: DISABLED Possible values = ENABLED, DISABLED
clientcert	<String>	Read-write	The rule for client certificate requirement in client authentication. Possible values = Mandatory, Optional
dhkeyexpsizelimit	<String>	Read-write	This option enables the use of NIST recommended (NIST Special Publication 800-56A) bit size for private-key size. For example, for DH params of size 2048bit, the private-key size recommended is 224bits. This is rounded-up to 256bits. Default value: DISABLED Possible values = ENABLED, DISABLED
sslredirect	<String>	Read-write	State of HTTPS redirects for the SSL service. For an SSL session, if the client browser receives a redirect message, the browser tries to connect to the new location. However, the secure SSL session breaks if the object has moved from a secure site (https://) to an unsecure site (http://). Typically, a warning message appears on the screen, prompting the user to continue or disconnect. If SSL Redirect is ENABLED, the redirect message is automatically converted from http:// to https:// and the SSL session does not break.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: DISABLED Possible values = ENABLED, DISABLED
redirectportrewrite	<String>	Read-write	State of the port rewrite while performing HTTPS redirect. If this parameter is set to ENABLED, and the URL from the server does not contain the standard port, the port is rewritten to the standard. Default value: DISABLED Possible values = ENABLED, DISABLED
nonfipsciphers	<String>	Read-write	State of usage of ciphers that are not FIPS approved. Valid only for an SSL service bound with a FIPS key and certificate. Default value: DISABLED Possible values = ENABLED, DISABLED
ssl3	<String>	Read-write	State of SSLv3 protocol support for the SSL profile. Default value: DISABLED Possible values = ENABLED, DISABLED
tls1	<String>	Read-write	State of TLSv1.0 protocol support for the SSL profile. Default value: ENABLED Possible values = ENABLED, DISABLED
tls11	<String>	Read-write	State of TLSv1.1 protocol support for the SSL profile. Default value: ENABLED Possible values = ENABLED, DISABLED
tls12	<String>	Read-write	State of TLSv1.2 protocol support for the SSL profile. Default value: ENABLED Possible values = ENABLED, DISABLED
snienable	<String>	Read-write	State of the Server Name Indication (SNI) feature on the virtual server and service-based offload. SNI helps to enable SSL encryption on multiple domains on a single virtual server or service if the domains are controlled by the same organization and share the same second-level domain name. For example, *.sports.net can be used to secure domains such as login.sports.net and help.sports.net. Default value: DISABLED Possible values = ENABLED, DISABLED
serverauth	<String>	Read-write	State of server authentication support for the SSL Backend profile. Default value: DISABLED Possible values = ENABLED, DISABLED
commonname	<String>	Read-write	Name to be checked against the CommonName (CN) field in the server certificate bound to the SSL server. Minimum length = 1
pushenctrigger	<String>	Read-write	Trigger encryption on the basis of the PUSH flag value. Available settings function as follows: * ALWAYS - Any PUSH packet triggers encryption. * IGNORE - Ignore PUSH packet for triggering encryption. * MERGE - For a consecutive sequence of PUSH packets, the last PUSH packet triggers encryption.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			in the set ssl parameter command or in the Change Advanced SSL Settings dialog box. Possible values = Always, Merge, Ignore, Timer
sendclosenotify	<String>	Read-write	Enable sending SSL Close-Notify at the end of a transaction. Default value: YES Possible values = YES, NO
cleartextport	<Integer>	Read-write	Port on which clear-text data is sent by the appliance to the server. Do not specify this parameter for SSL offloading with end-to-end encryption. Range 1 - 65535
insertionencoding	<String>	Read-write	Encoding method used to insert the subject or issuer's name in HTTP requests to servers. Default value: Unicode Possible values = Unicode, UTF-8
denysslreneg	<String>	Read-write	Deny renegotiation in specified circumstances. Available settings function as follows: * NO - Allow SSL renegotiation. * FRONTEND_CLIENT - Deny secure and nonsecure SSL renegotiation initiated by the client. * FRONTEND_CLIENTSERVER - Deny secure and nonsecure SSL renegotiation initiated by the client or the NetScaler during policy-based client authentication. * ALL - Deny all secure and nonsecure SSL renegotiation. * NONSECURE - Deny nonsecure SSL renegotiation. Allows only clients that support RFC 5746. Default value: ALL Possible values = NO, FRONTEND_CLIENT, FRONTEND_CLIENTSERVER, ALL, NONSECURE
quantumsize	<String>	Read-write	Amount of data to collect before the data is pushed to the crypto hardware for encryption. For large downloads, a larger quantum size better utilizes the crypto resources. Default value: 8192 Possible values = 4096, 8192, 16384
strictcachecks	<String>	Read-write	Enable strict CA certificate checks on the appliance. Default value: NO Possible values = YES, NO
encrypttriggerpktcount	<Double>	Read-write	Maximum number of queued packets after which encryption is triggered. Use this setting for SSL transactions that send small packets from server to NetScaler. Default value: 45 Minimum value = 10 Maximum value = 50
pushflag	<Double>	Read-write	Insert PUSH flag into decrypted, encrypted, or all records. If the PUSH flag is set to a value other than 0, the buffered records are forwarded on the basis of the value of the PUSH flag. Available settings function as follows: 0 - Auto (PUSH flag is not set.) 1 - Insert PUSH flag into every decrypted record.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			3 - Insert PUSH flag into every decrypted and encrypted record. Minimum value = 0 Maximum value = 3
dropreqwithnohostheader	<String>	Read-write	Host header check for SNI enabled sessions. If this check is enabled and the HTTP request does not contain the host header for SNI enabled sessions, the request is dropped. Default value: NO Possible values = YES, NO
pushenctriggertimeout	<Double>	Read-write	PUSH encryption trigger timeout value. The timeout value is applied only if you set the Push Encryption Trigger parameter to Timer in the SSL virtual server settings. Default value: 1 Minimum value = 1 Maximum value = 200
ssltriggertimeout	<Double>	Read-write	Time, in milliseconds, after which encryption is triggered for transactions that are not tracked on the NetScaler appliance because their length is not known. There can be a delay of up to 10ms from the specified timeout value before the packet is pushed into the queue. Default value: 100 Minimum value = 1 Maximum value = 200
clientauthuseboundcachain	<String>	Read-write	Certficates bound on the VIP are used for validating the client cert. Certficates came along with client cert are not used for validating the client cert. Default value: DISABLED Possible values = ENABLED, DISABLED
ciphername	<String>	Read-write	The cipher group/alias/individual cipher configuration.
cipherpriority	<Double>	Read-write	cipher priority. Minimum value = 1
crlcheck	<String>	Read-only	The state of the CRL check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
ocspcheck	<String>	Read-only	The state of the OCSP check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
snicert	<Boolean>	Read-only	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
skipcaname	<Boolean>	Read-only	The flag is used to indicate whether this particular CA certificate's CA_Name needs to be sent to the SSL client while requesting for client certificate in a SSL handshake.
invoke	<Boolean>	Read-only	Invoke flag. This attribute is relevant only for ADVANCED policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Possible values = vserver, service, policylabel
Name	**Data Type**	**Permissions**	**Description**
service	<Double>	Read-only	Service.
builtin	<String[]>	Read-only	Flag to determine whether ssl profile is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslprofile":{
    "name":<String_value>,
    "sslprofiletype":<String_value>,
    "dhcount":<Double_value>,
    "dh":<String_value>,
    "dhfile":<String_value>,
    "ersa":<String_value>,
    "ersacount":<Double_value>,
    "sessreuse":<String_value>,
    "sesstimeout":<Double_value>,
    "cipherredirect":<String_value>,
    "cipherurl":<String_value>,
    "clientauth":<String_value>,
    "clientcert":<String_value>,
    "dhkeyexpsizelimit":<String_value>,
    "sslredirect":<String_value>,
    "redirectportrewrite":<String_value>,
    "nonfipsciphers":<String_value>,
    "ssl3":<String_value>,
    "tls1":<String_value>,
    "tls11":<String_value>,
    "tls12":<String_value>,
    "snienable":<String_value>,
    "serverauth":<String_value>,
    "commonname":<String_value>,
    "pushenctrigger":<String_value>,
    "sendclosenotify":<String_value>,
    "cleartextport":<Integer_value>,
    "insertionencoding":<String_value>,
    "denysslreneg":<String_value>,
    "quantumsize":<String_value>,
    "strictcachecks":<String_value>,
    "encrypttriggerpktcount":<Double_value>,
    "pushflag":<Double_value>,
    "dropreqwithnohostheader":<String_value>,
    "pushenctriggertimeout":<Double_value>,
    "ssltriggertimeout":<Double_value>,
    "clientauthuseboundcachain":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslprofile":{
    "name":<String_value>,
    "dh":<String_value>,
    "dhfile":<String_value>,
    "dhcount":<Double_value>,
    "dhkeyexpsizelimit":<String_value>,
    "ersa":<String_value>,
    "ersacount":<Double_value>,
    "sessreuse":<String_value>,
    "sesstimeout":<Double_value>,
    "cipherredirect":<String_value>,
    "cipherurl":<String_value>,
    "clientauth":<String_value>,
    "clientcert":<String_value>,
    "sslredirect":<String_value>,
    "redirectportrewrite":<String_value>,
    "nonfipsciphers":<String_value>,
    "ssl3":<String_value>,
    "tls1":<String_value>,
    "tls11":<String_value>,
    "tls12":<String_value>,
    "snienable":<String_value>,
    "serverauth":<String_value>,
    "commonname":<String_value>,
    "pushenctrigger":<String_value>,
    "sendclosenotify":<String_value>,
    "cleartextport":<Integer_value>,
    "insertionencoding":<String_value>,
    "denysslreneg":<String_value>,
    "quantumsize":<String_value>,
    "strictcachecks":<String_value>,
    "encrypttriggerpktcount":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "pushflag":<Double_value>,
    "dropreqwithnohostheader":<String_value>,
    "pushenctriggertimeout":<Double_value>,
    "ssltriggertimeout":<Double_value>,
    "clientauthuseboundcachain":<String_value>,
    "ciphername":<String_value>,
    "cipherpriority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslprofile":{
    "name":<String_value>,
    "dh":true,
    "dhfile":true,
    "dhcount":true,
    "dhkeyexpsizelimit":true,
    "ersa":true,
    "ersacount":true,
    "sessreuse":true,
    "sesstimeout":true,
    "cipherredirect":true,
    "cipherurl":true,
    "clientauth":true,
    "clientcert":true,
    "sslredirect":true,
    "redirectportrewrite":true,
    "nonfipsciphers":true,
    "ssl3":true,
    "tls1":true,
    "tls11":true,
    "tls12":true,
    "snienable":true,
    "serverauth":true,
    "commonname":true,
    "pushenctrigger":true,
    "sendclosenotify":true,
    "cleartextport":true,
    "insertionencoding":true,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "denysslreneg":true,
        "quantumsize":true,
        "strictcachecks":true,
        "encrypttriggerpktcount":true,
        "pushflag":true,
        "dropreqwithnohostheader":true,
        "pushenctriggertimeout":true,
        "ssltriggertimeout":true,
        "clientauthuseboundcachain":true,
        "ciphername":true,
        "cipherpriority":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslprofile resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "sslprofile": [ {
    "name":<String_value>,
    "dh":<String_value>,
    "dhfile":<String_value>,
    "dhcount":<Double_value>,
    "dhkeyexpsizelimit":<String_value>,
    "ersa":<String_value>,
    "ersacount":<Double_value>,
    "sessreuse":<String_value>,
    "sesstimeout":<Double_value>,
    "cipherredirect":<String_value>,
    "cipherurl":<String_value>,
    "clientauth":<String_value>,
    "clientcert":<String_value>,
    "sslredirect":<String_value>,
    "redirectportrewrite":<String_value>,
    "nonfipsciphers":<String_value>,
    "ssl3":<String_value>,
    "tls1":<String_value>,
    "tls11":<String_value>,
    "tls12":<String_value>,
    "snienable":<String_value>,
    "serverauth":<String_value>,
    "commonname":<String_value>,
    "pushenctrigger":<String_value>,
    "sendclosenotify":<String_value>,
    "cleartextport":<Integer_value>,
    "insertionencoding":<String_value>,
    "denysslreneg":<String_value>,
    "quantumsize":<String_value>,
    "strictcachecks":<String_value>,
    "encrypttriggerpktcount":<Double_value>,
    "pushflag":<Double_value>,
    "dropreqwithnohostheader":<String_value>,
    "pushenctriggertimeout":<Double_value>,
    "ssltriggertimeout":<Double_value>,
    "cipherpriority":<Double_value>,
    "ciphername":<String_value>,
    "crlcheck":<String_value>,
    "ocspcheck":<String_value>,
    "snicert":<Boolean_value>,
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "skipcaname":<Boolean_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "sslprofiletype":<String_value>,
    "clientauthuseboundcachain":<String_value>,
    "service":<Double_value>,
    "builtin":<String[]_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslprofile": [ {
    "name":<String_value>,
    "dh":<String_value>,
    "dhfile":<String_value>,
    "dhcount":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "dhkeyexpsizelimit":<String_value>,
    "ersa":<String_value>,
    "ersacount":<Double_value>,
    "sessreuse":<String_value>,
    "sesstimeout":<Double_value>,
    "cipherredirect":<String_value>,
    "cipherurl":<String_value>,
    "clientauth":<String_value>,
    "clientcert":<String_value>,
    "sslredirect":<String_value>,
    "redirectportrewrite":<String_value>,
    "nonfipsciphers":<String_value>,
    "ssl3":<String_value>,
    "tls1":<String_value>,
    "tls11":<String_value>,
    "tls12":<String_value>,
    "snienable":<String_value>,
    "serverauth":<String_value>,
    "commonname":<String_value>,
    "pushenctrigger":<String_value>,
    "sendclosenotify":<String_value>,
    "cleartextport":<Integer_value>,
    "insertionencoding":<String_value>,
    "denysslreneg":<String_value>,
    "quantumsize":<String_value>,
    "strictcachecks":<String_value>,
    "encrypttriggerpktcount":<Double_value>,
    "pushflag":<Double_value>,
    "dropreqwithnohostheader":<String_value>,
    "pushenctriggertimeout":<Double_value>,
    "ssltriggertimeout":<Double_value>,
    "cipherpriority":<Double_value>,
    "ciphername":<String_value>,
    "crlcheck":<String_value>,
    "ocspcheck":<String_value>,
    "snicert":<Boolean_value>,
    "skipcaname":<Boolean_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "sslprofiletype":<String_value>,
    "clientauthuseboundcachain":<String_value>,
    "service":<Double_value>,
    "builtin":<String[]_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "sslprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslprofile_binding

May 20, 2016

Binding object which returns the resources bound to sslprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the SSL profile for which to show detailed information. Minimum length = 1 Maximum length = 127
sslprofile_sslcipher_binding	<sslprofile_sslcipher_binding[]>	Read-only	sslcipher that can be bound to sslprofile.
sslprofile_ecccurve_binding	<sslprofile_ecccurve_binding[]>	Read-only	ecccurve that can be bound to sslprofile.
sslprofile_sslvserver_binding	<sslprofile_sslvserver_binding[]>	Read-only	sslvserver that can be bound to sslprofile.
sslprofile_sslciphersuite_binding	<sslprofile_sslciphersuite_binding[]>	Read-only	sslciphersuite that can be bound to sslprofile.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "sslprofile_binding": [ {
      "name":<String_value>,
      "sslprofile_ecccurve_binding":<sslprofile_ecccurve_binding[]_value>,
      "sslprofile_sslciphersuite_binding":<sslprofile_sslciphersuite_binding[]_value>,
      "sslprofile_sslvserver_binding":<sslprofile_sslvserver_binding[]_value>,
      "sslprofile_sslcipher_binding":<sslprofile_sslcipher_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslprofile_ecccurve_binding

Jul 07, 2017

Binding object showing the ecccurve that can be bound to sslprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cipherpriority	<Double>	Read-write	Priority of the cipher binding. Minimum value = 1 Maximum value = 1000
name	<String>	Read-write	Name of the SSL profile. Minimum length = 1 Maximum length = 127
ecccurvename	<String>	Read-write	Named ECC curve bound to vserver/service. Possible values = ALL, P_224, P_256, P_384, P_521
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslprofile_ecccurve_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslprofile_ecccurve_binding":{
    "name":<String_value>,
    "cipherpriority":<Double_value>,
    "ecccurvename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslprofile_ecccurve_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslprofile_ecccurve_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslprofile_ecccurve_binding": [ {
      "cipherpriority":<Double_value>,
      "name":<String_value>,
      "ecccurvename":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_ecccurve_binding/name_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslprofile_ecccurve_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslrsakey

May 20, 2016

Configuration for RSA key resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
keyfile	<String>	Read-write	Name for and, optionally, path to the RSA key file. /nsconfig/ssl/ is the default path. Maximum length = 63
bits	<Double>	Read-write	Size, in bits, of the RSA key. Minimum value = 512 Maximum value = 4096
exponent	<String>	Read-write	Public exponent for the RSA key. The exponent is part of the cipher algorithm and is required for creating the RSA key. Default value: F4 Possible values = 3, F4
keyform	<String>	Read-write	Format in which the RSA key file is stored on the appliance. Default value: PEM Possible values = DER, PEM
des	<Boolean>	Read-write	Encrypt the generated RSA key by using the DES algorithm. On the command line, you are prompted to enter the pass phrase (password) that is used to encrypt the key.
des3	<Boolean>	Read-write	Encrypt the generated RSA key by using the Triple-DES algorithm. On the command line, you are prompted to enter the pass phrase (password) that is used to encrypt the key.
password	<String>	Read-write	Pass phrase to use for encryption if DES or DES3 option is selected. Minimum length = 1 Maximum length = 31

Operations (click to see Properties)

CREATE

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

create

URL: http://<netscaler-ip-address>/nitro/v1/config/sslrsakey?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslrsakey":{
    "keyfile":<String_value>,
    "bits":<Double_value>,
    "exponent":<String_value>,
    "keyform":<String_value>,
    "des":<Boolean_value>,
    "des3":<Boolean_value>,
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslprofile_sslciphersuite_binding

May 20, 2016

Binding object showing the sslciphersuite that can be bound to sslprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	The cipher group/alias/individual cipher configuration.
cipherpriority	<Double>	Read-write	cipher priority. Minimum value = 1
name	<String>	Read-write	Name of the SSL profile. Minimum length = 1 Maximum length = 127
description	<String>	Read-write	The cipher suite description.
cipheraliasname	<String>	Read-only	The name of the cipher group/alias/individual cipheri bindings.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslprofile_sslciphersuite_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslprofile_sslciphersuite_binding":{
    "name":<String_value>,
    "ciphername":<String_value>,
    "cipherpriority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslprofile_sslciphersuite_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslprofile_sslciphersuite_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslprofile_sslciphersuite_binding": [ {
    "ciphername":<String_value>,
    "cipherpriority":<Double_value>,
    "name":<String_value>,
    "description":<String_value>,
    "cipheraliasname":<String_value>
}]}
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslciphersuite_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslprofile_sslciphersuite_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslprofile_sslcipher_binding

May 20, 2016

Binding object showing the sslcipher that can be bound to sslprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	The cipher group/alias/individual cipher configuration.
cipheraliasname	<String>	Read-write	The name of the cipher group/alias/individual cipheri bindings.
cipherpriority	<Double>	Read-write	cipher priority. Minimum value = 1
name	<String>	Read-write	Name of the SSL profile. Minimum length = 1 Maximum length = 127
description	<String>	Read-write	The cipher suite description.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslprofile_sslcipher_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslprofile_sslcipher_binding":{
    "name":<String_value>,
    "ciphername":<String_value>,
    "cipherpriority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslprofile_sslcipher_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslprofile_sslcipher_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslprofile_sslcipher_binding": [ {
      "ciphername":<String_value>,
      "cipheraliasname":<String_value>,
      "cipherpriority":<Double_value>,
      "name":<String_value>,
      "description":<String_value>
}]}
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslcipher_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslprofile_sslcipher_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslprofile_sslvserver_binding

Jul 07, 2017

Binding object showing the sslvserver that can be bound to sslprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
cipherpriority	<Double>	Read-write	Priority of the cipher binding. Minimum value = 1 Maximum value = 1000
name	<String>	Read-write	Name of the SSL profile. Minimum length = 1 Maximum length = 127
description	<String>	Read-write	The cipher suite description.
servicename	<String>	Read-write	Service name.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslprofile_sslvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslprofile_sslvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslprofile_sslvserver_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cipherpriority":<Double_value>,
    "name":<String_value>,
    "description":<String_value>,
    "servicename":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslprofile_sslvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslprofile_sslvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservice

May 20, 2016

Configuration for SSL service resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	Name of the SSL service. Minimum length = 1
dh	<String>	Read-write	State of Diffie-Hellman (DH) key exchange. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
dhfile	<String>	Read-write	Name for and, optionally, path to the PEM-format DH parameter file to be installed. /nsconfig/ssl/ is the default path. This parameter is not applicable when configuring a backend service. Minimum length = 1
dhcount	<Double>	Read-write	Number of interactions, between the client and the NetScaler appliance, after which the DH private-public pair is regenerated. A value of zero (0) specifies infinite use (no refresh). This parameter is not applicable when configuring a backend service. Minimum value = 0 Maximum value = 65534
dhkeyexpsizelimit	<String>	Read-write	This option enables the use of NIST recommended (NIST Special Publication 800-56A) bit size for private-key size. For example, for DH params of size 2048bit, the private-key size recommended is 224bits. This is rounded-up to 256bits. Default value: DISABLED Possible values = ENABLED, DISABLED
ersa	<String>	Read-write	State of Ephemeral RSA (eRSA) key exchange. Ephemeral RSA allows clients that support only export ciphers to communicate with the secure server even if the server certificate does not support export clients. The ephemeral RSA key is automatically generated when you bind an export cipher to an SSL or TCP-based SSL virtual server or service. When you remove the export cipher, the eRSA key is not deleted. It is reused at a later date when another export cipher is bound to an SSL or TCP-based SSL virtual server or service. The eRSA key is deleted when the appliance restarts. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
			Refresh count for regeneration of RSA public-key and private-key pair. Zero (0) specifies infinite usage (no refresh).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 0 Maximum value = 65534
sessreuse	<String>	Read-write	State of session reuse. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the ENABLED setting, session key exchange is avoided for session resumption requests received from the client. Default value: ENABLED Possible values = ENABLED, DISABLED
sesstimeout	<Double>	Read-write	Time, in seconds, for which to keep the session active. Any session resumption request received after the timeout period will require a fresh SSL handshake and establishment of a new SSL session. Default value: 300 Minimum value = 0 Maximum value = 4294967294
cipherredirect	<String>	Read-write	State of Cipher Redirect. If this parameter is set to ENABLED, you can configure an SSL virtual server or service to display meaningful error messages if the SSL handshake fails because of a cipher mismatch between the virtual server or service and the client. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
cipherurl	<String>	Read-write	URL of the page to which to redirect the client in case of a cipher mismatch. Typically, this page has a clear explanation of the error or an alternative location that the transaction can continue from. This parameter is not applicable when configuring a backend service.
sslv2redirect	<String>	Read-write	State of SSLv2 Redirect. If this parameter is set to ENABLED, you can configure an SSL virtual server or service to display meaningful error messages if the SSL handshake fails because of a protocol version mismatch between the virtual server or service and the client. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
sslv2url	<String>	Read-write	URL of the page to which to redirect the client in case of a protocol version mismatch. Typically, this page has a clear explanation of the error or an alternative location that the transaction can continue from. This parameter is not applicable when configuring a backend service.
clientauth	<String>	Read-write	State of client authentication. In service-based SSL offload, the service terminates the SSL handshake if the SSL client does not provide a valid certificate. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
clientcert	<String>	Read-write	Type of client authentication. If this parameter is set to MANDATORY, the appliance terminates the SSL handshake if the SSL client does not provide a valid certificate. With the OPTIONAL setting, the appliance requests a certificate from the SSL clients but proceeds with the SSL transaction even if the client presents an invalid certificate. This parameter is not applicable when configuring a backend SSL service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Optional. Possible values = Mandatory, Optional
sslredirect	<String>	Read-write	State of HTTPS redirects for the SSL service. For an SSL session, if the client browser receives a redirect message, the browser tries to connect to the new location. However, the secure SSL session breaks if the object has moved from a secure site (https://) to an unsecure site (http://). Typically, a warning message appears on the screen, prompting the user to continue or disconnect. If SSL Redirect is ENABLED, the redirect message is automatically converted from http:// to https:// and the SSL session does not break. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
redirectportrewrite	<String>	Read-write	State of the port rewrite while performing HTTPS redirect. If this parameter is set to ENABLED, and the URL from the server does not contain the standard port, the port is rewritten to the standard. Default value: DISABLED Possible values = ENABLED, DISABLED
nonfipsciphers	<String>	Read-write	State of usage of ciphers that are not FIPS approved. Valid only for an SSL service bound with a FIPS key and certificate. Default value: DISABLED Possible values = ENABLED, DISABLED
ssl2	<String>	Read-write	State of SSLv2 protocol support for the SSL service. This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
ssl3	<String>	Read-write	State of SSLv3 protocol support for the SSL service. Default value: ENABLED Possible values = ENABLED, DISABLED
tls1	<String>	Read-write	State of TLSv1.0 protocol support for the SSL service. Default value: ENABLED Possible values = ENABLED, DISABLED
tls11	<String>	Read-write	State of TLSv1.1 protocol support for the SSL service. Enabled for Front-end service on MPX-CVM platform only. Default value: ENABLED Possible values = ENABLED, DISABLED
tls12	<String>	Read-write	State of TLSv1.2 protocol support for the SSL service. Enabled for Front-end service on MPX-CVM platform only. Default value: ENABLED Possible values = ENABLED, DISABLED
			State of the Server Name Indication (SNI) feature on the virtual server and service-based offload. SNI helps to enable SSL encryption on multiple domains on a single

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
snienable	<String>	Read-write	share the same second-level domain name. For example, *.sports.net can be used to secure domains such as login.sports.net and help.sports.net.
			This parameter is not applicable when configuring a backend service. Default value: DISABLED Possible values = ENABLED, DISABLED
serverauth	<String>	Read-write	State of server authentication support for the SSL service. Default value: DISABLED Possible values = ENABLED, DISABLED
commonname	<String>	Read-write	Name to be checked against the CommonName (CN) field in the server certificate bound to the SSL server. Minimum length = 1
pushenctrigger	<String>	Read-write	Trigger encryption on the basis of the PUSH flag value. Available settings function as follows: * ALWAYS - Any PUSH packet triggers encryption. * IGNORE - Ignore PUSH packet for triggering encryption. * MERGE - For a consecutive sequence of PUSH packets, the last PUSH packet triggers encryption. * TIMER - PUSH packet triggering encryption is delayed by the time defined in the set ssl parameter command or in the Change Advanced SSL Settings dialog box. Possible values = Always, Merge, Ignore, Timer
sendclosenotify	<String>	Read-write	Enable sending SSL Close-Notify at the end of a transaction. Default value: YES Possible values = YES, NO
dtlsprofilename	<String>	Read-write	Name of the DTLS profile that contains DTLS settings for the service. Minimum length = 1 Maximum length = 127
sslprofile	<String>	Read-write	Name of the SSL profile that contains SSL settings for the service. Minimum length = 1 Maximum length = 127
cipherdetails	<Boolean>	Read-write	Display details of the individual ciphers bound to the SSL service.
service	<Double>	Read-only	.
skipcaname	<Boolean>	Read-only	The flag is used to indicate whether this particular CA certificate's CA_Name needs to be sent to the SSL client while requesting for client certificate in a SSL handshake.
dtlsflag	<Boolean>	Read-only	The flag is used to indicate whether DTLS is set or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslservice":{
    "servicename":<String_value>,
    "dh":<String_value>,
    "dhfile":<String_value>,
    "dhcount":<Double_value>,
    "dhkeyexpsizelimit":<String_value>,
    "ersa":<String_value>,
    "ersacount":<Double_value>,
    "sessreuse":<String_value>,
    "sesstimeout":<Double_value>,
    "cipherredirect":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cipherurl":<String_value>,
    "sslv2redirect":<String_value>,
    "sslv2url":<String_value>,
    "clientauth":<String_value>,
    "clientcert":<String_value>,
    "sslredirect":<String_value>,
    "redirectportrewrite":<String_value>,
    "nonfipsciphers":<String_value>,
    "ssl2":<String_value>,
    "ssl3":<String_value>,
    "tls1":<String_value>,
    "tls11":<String_value>,
    "tls12":<String_value>,
    "snienable":<String_value>,
    "serverauth":<String_value>,
    "commonname":<String_value>,
    "pushenctrigger":<String_value>,
    "sendclosenotify":<String_value>,
    "dtlsprofilename":<String_value>,
    "sslprofile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslservice":{
    "servicename":<String_value>,
    "dh":true,
    "dhfile":true,
    "dhcount":true,
    "dhkeyexpsizelimit":true,
    "ersa":true,
    "ersacount":true,
    "sessreuse":true,
    "sesstimeout":true,
    "cipherredirect":true,
    "cipherurl":true,
    "sslv2redirect":true,
    "sslv2url":true,
    "clientauth":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "clientcert":true,
        "sslredirect":true,
        "redirectportrewrite":true,
        "nonfipsciphers":true,
        "ssl2":true,
        "ssl3":true,
        "tls1":true,
        "tls11":true,
        "tls12":true,
        "snienable":true,
        "serverauth":true,
        "commonname":true,
        "sendclosenotify":true,
        "dtlsprofilename":true,
        "sslprofile":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/sslservice?**args=servicename:<String_value>,cipherdetails:<Boolean_value>**

Use this query-parameter to get sslservice resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslservice?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservice?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslservice resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslservice?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservice?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservice resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservice": [ {
servicename:<String_value>,cipherdetails:<Boolean_value>      "dh":<String_value>,
      "dhfile":<String_value>,
      "dhcount":<Double_value>,
      "dhkeyexpsizelimit":<String_value>,
      "ersa":<String_value>,
      "ersacount":<Double_value>,
      "sessreuse":<String_value>,
      "sesstimeout":<Double_value>,
      "cipherredirect":<String_value>,
      "cipherurl":<String_value>,
      "sslv2redirect":<String_value>,
      "sslv2url":<String_value>,
      "clientauth":<String_value>,
      "clientcert":<String_value>,
      "sslredirect":<String_value>,
      "redirectportrewrite":<String_value>,
      "nonfipsciphers":<String_value>,
      "ssl2":<String_value>,
      "ssl3":<String_value>,
      "tls1":<String_value>,
      "tls11":<String_value>,
      "tls12":<String_value>,
      "snienable":<String_value>,
      "serverauth":<String_value>,
      "commonname":<String_value>,
      "service":<Double_value>,
      "pushenctrigger":<String_value>,
      "skipcaname":<Boolean_value>,
      "sendclosenotify":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "dtlsprofilename":<String_value>,
    "dtlsflag":<Boolean_value>,
    "sslprofile":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice/servicename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslservice/servicename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslservice/servicename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservice": [ {
servicename:<String_value>,cipherdetails:<Boolean_value>      "dh":<String_value>,
    "dhfile":<String_value>,
    "dhcount":<Double_value>,
    "dhkeyexpsizelimit":<String_value>,
    "ersa":<String_value>,
    "ersacount":<Double_value>,
    "sessreuse":<String_value>,
    "sesstimeout":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cipherredirect":<String_value>,
    "cipherurl":<String_value>,
    "sslv2redirect":<String_value>,
    "sslv2url":<String_value>,
    "clientauth":<String_value>,
    "clientcert":<String_value>,
    "sslredirect":<String_value>,
    "redirectportrewrite":<String_value>,
    "nonfipsciphers":<String_value>,
    "ssl2":<String_value>,
    "ssl3":<String_value>,
    "tls1":<String_value>,
    "tls11":<String_value>,
    "tls12":<String_value>,
    "snienable":<String_value>,
    "serverauth":<String_value>,
    "commonname":<String_value>,
    "service":<Double_value>,
    "pushenctrigger":<String_value>,
    "skipcaname":<Boolean_value>,
    "sendclosenotify":<String_value>,
    "dtlsprofilename":<String_value>,
    "dtlsflag":<Boolean_value>,
    "sslprofile":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservice": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservicegroup

May 20, 2016

Configuration for SSL service group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Name of the SSL service group for which to set advanced configuration. Minimum length = 1
sslprofile	<String>	Read-write	Name of the SSL profile that contains SSL settings for the Service Group. Minimum length = 1 Maximum length = 127
sessreuse	<String>	Read-write	State of session reuse. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the ENABLED setting, session key exchange is avoided for session resumption requests received from the client. Default value: ENABLED Possible values = ENABLED, DISABLED
sesstimeout	<Double>	Read-write	Time, in seconds, for which to keep the session active. Any session resumption request received after the timeout period will require a fresh SSL handshake and establishment of a new SSL session. Default value: 300 Minimum value = 0 Maximum value = 4294967294
nonfipsciphers	<String>	Read-write	State of usage of ciphers that are not FIPS approved. Valid only for an SSL service bound with a FIPS key and certificate. Default value: DISABLED Possible values = ENABLED, DISABLED
ssl3	<String>	Read-write	State of SSLv3 protocol support for the SSL service group. Default value: ENABLED Possible values = ENABLED, DISABLED
tls1	<String>	Read-write	State of TLSv1.0 protocol support for the SSL service group. Default value: ENABLED Possible values = ENABLED, DISABLED
tls11	<String>	Read-write	State of TLSv1.1 protocol support for the SSL service group. Default value: ENABLED Possible values = ENABLED, DISABLED
tls12	<String>	Read-write	State of TLSv1.2 protocol support for the SSL service group. Default value: ENABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
serverauth	<String>	Read-write	State of server authentication support for the SSL service group. Default value: DISABLED Possible values = ENABLED, DISABLED
commonname	<String>	Read-write	Name to be checked against the CommonName (CN) field in the server certificate bound to the SSL server. Minimum length = 1
sendclosenotify	<String>	Read-write	Enable sending SSL Close-Notify at the end of a transaction. Default value: YES Possible values = YES, NO
cipherdetails	<Boolean>	Read-write	Display details of the individual ciphers bound to the SSL service group.
dh	<String>	Read-only	The state of DH key exchange support for the SSL service group. Possible values = ENABLED, DISABLED
dhfile	<String>	Read-only	The file name and path for the DH parameter.
dhcount	<Double>	Read-only	The refresh count for the re-generation of DH public-key and private-key from the DH parameter.
dhkeyexpsizelimit	<String>	Read-only	This option enables the use of NIST recommended (NIST Special Publication 800-56A) bit size for private-key size. For example, for DH params of size 2048bit, the private-key size recommended is 224bits. This is rounded-up to 256bits. Possible values = ENABLED, DISABLED
ersa	<String>	Read-only	The state of Ephemeral RSA key exchange support for the SSL service group.Ephemeral RSA is used for export ciphers. Possible values = ENABLED, DISABLED
ersacount	<Double>	Read-only	The refresh count for the re-generation of RSA public-key and private-key pair.
cipherredirect	<String>	Read-only	The state of Cipher Redirect feature. Cipher Redirect feature can be used to provide more readable information to SSL clients about mismatch in ciphers between the client and the SSL vserver. Possible values = ENABLED, DISABLED
cipherurl	<String>	Read-only	The redirect URL to be used with the Cipher Redirect feature.
sslv2redirect	<String>	Read-only	The state of SSLv2 Redirect feature.SSLv2 Redirect feature can be used to provide more readable information to SSL client about non-support of SSLv2 protocol on the SSL vserver. Possible values = ENABLED, DISABLED
sslv2url	<String>	Read-only	The redirect URL to be used with SSLv2 Redirect feature.
clientauth	<String>	Read-only	The state of Client-Authentication support for the SSL service group.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
clientcert	<String>	Read-only	The rule for client certificate requirement in client authentication. Possible values = Mandatory, Optional
sslredirect	<String>	Read-only	The state of HTTPS redirects for the SSL service group. This is required for the proper functioning of the redirect messages from the server. The redirect message from the server provides the new location for the moved object. This is contained in the HTTP header field: Location, e.g. Location: http://www.moved.org/here.html For the SSL session, if the client browser receives this message, the browser will try to connect to the new location. This will break the secure SSL session, as the object has moved from a secure site (https://) to an un-secure one (http://). Generally browsers flash a warning message on the screen and prompt the user, either to continue or disconnect. The above feature, when enabled will automatically convert all such http:// redirect message to https://. This will not break the client SSL session. Note: The set ssl service command can be used for configuring a front-end SSL service for service based SSL Off-Loading, or a backend SSL service for backend-encryption setup. Possible values = ENABLED, DISABLED
redirectportrewrite	<String>	Read-only	The state of port-rewrite feature. Possible values = ENABLED, DISABLED
ssl2	<String>	Read-only	The state of SSLv2 protocol support for the SSL service group. Possible values = ENABLED, DISABLED
ocspcheck	<String>	Read-only	The state of the OCSP check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
crlcheck	<String>	Read-only	The state of the CRL check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
cleartextport	<Integer>	Read-only	The port on the back-end web-servers where the clear-text data is sent by system. Use this setting for the wildcard IP based SSL Acceleration configuration (*:443). Range 1 - 65535
servicename	<String>	Read-only	The service name.
ca	<Boolean>	Read-only	CA certificate.
snicert	<Boolean>	Read-only	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslservicegroup":{
    "servicegroupname":<String_value>,
    "sslprofile":<String_value>,
    "sessreuse":<String_value>,
    "sesstimeout":<Double_value>,
    "nonfipsciphers":<String_value>,
    "ssl3":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "tls1":<String_value>,
    "tls11":<String_value>,
    "tls12":<String_value>,
    "serverauth":<String_value>,
    "commonname":<String_value>,
    "sendclosenotify":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslservicegroup":{
    "servicegroupname":<String_value>,
    "sslprofile":true,
    "sessreuse":true,
    "sesstimeout":true,
    "nonfipsciphers":true,
    "ssl3":true,
    "tls1":true,
    "tls11":true,
    "tls12":true,
    "serverauth":true,
    "commonname":true,
    "sendclosenotify":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup?**args=servicegroupname:<String_value>,cipherdetails:<Boolean_value>**

Use this query-parameter to get sslservicegroup resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslservicegroup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservicegroup resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "sslservicegroup": [ {
servicegroupname:<String_value>,cipherdetails:<Boolean_value>      "dh":<String_value>,
      "dhfile":<String_value>,
      "dhcount":<Double_value>,
      "dhkeyexpsizelimit":<String_value>,
      "ersa":<String_value>,
      "ersacount":<Double_value>,
      "sessreuse":<String_value>,
      "sesstimeout":<Double_value>,
      "cipherredirect":<String_value>,
      "cipherurl":<String_value>,
      "sslv2redirect":<String_value>,
      "sslv2url":<String_value>,
      "clientauth":<String_value>,
      "clientcert":<String_value>,
      "sslredirect":<String_value>,
      "redirectportrewrite":<String_value>,
      "nonfipsciphers":<String_value>,
      "ssl2":<String_value>,
      "ssl3":<String_value>,
      "tls1":<String_value>,
      "tls11":<String_value>,
      "tls12":<String_value>,
      "serverauth":<String_value>,
      "commonname":<String_value>,
      "ocspcheck":<String_value>,
      "crlcheck":<String_value>,
      "cleartextport":<Integer_value>,
      "servicename":<String_value>,
      "ca":<Boolean_value>,
      "snicert":<Boolean_value>,
      "sendclosenotify":<String_value>,
      "sslprofile":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup/servicegroupname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup/servicegroupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup/servicegroupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservicegroup": [ {
servicegroupname:<String_value>,cipherdetails:<Boolean_value>      "dh":<String_value>,
      "dhfile":<String_value>,
      "dhcount":<Double_value>,
      "dhkeyexpsizelimit":<String_value>,
      "ersa":<String_value>,
      "ersacount":<Double_value>,
      "sessreuse":<String_value>,
      "sesstimeout":<Double_value>,
      "cipherredirect":<String_value>,
      "cipherurl":<String_value>,
      "sslv2redirect":<String_value>,
      "sslv2url":<String_value>,
      "clientauth":<String_value>,
      "clientcert":<String_value>,
      "sslredirect":<String_value>,
      "redirectportrewrite":<String_value>,
      "nonfipsciphers":<String_value>,
      "ssl2":<String_value>,
      "ssl3":<String_value>,
      "tls1":<String_value>,
      "tls11":<String_value>,
      "tls12":<String_value>,
      "serverauth":<String_value>,
      "commonname":<String_value>,
      "ocspcheck":<String_value>,
      "crlcheck":<String_value>,
      "cleartextport":<Integer_value>,
      "servicename":<String_value>,
      "ca":<Boolean_value>,
      "snicert":<Boolean_value>,
      "sendclosenotify":<String_value>,
      "sslprofile":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "sslservicegroup": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservicegroup_binding

May 20, 2016

Binding object which returns the resources bound to sslservicegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Name of the SSL service group for which to show detailed information. Minimum length = 1
sslservicegroup_sslcipher_binding	<sslservicegroup_sslcipher_binding[]>	Read-only	sslcipher that can be bound to sslservicegroup.
sslservicegroup_ecccurve_binding	<sslservicegroup_ecccurve_binding[]>	Read-only	ecccurve that can be bound to sslservicegroup.
sslservicegroup_sslcertkey_binding	<sslservicegroup_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to sslservicegroup.
sslservicegroup_sslciphersuite_binding	<sslservicegroup_sslciphersuite_binding[]>	Read-only	sslciphersuite that can be bound to sslservicegroup.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_binding/servicegroupname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "sslservicegroup_binding": [ {
    "servicegroupname":<String_value>,
    "sslservicegroup_sslcipher_binding":<sslservicegroup_sslcipher_binding[]_value>,
    "sslservicegroup_sslciphersuite_binding":<sslservicegroup_sslciphersuite_binding[]_value>,
    "sslservicegroup_ecccurve_binding":<sslservicegroup_ecccurve_binding[]_value>,
    "sslservicegroup_sslcertkey_binding":<sslservicegroup_sslcertkey_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservicegroup_ecccurve_binding

May 20, 2016

Binding object showing the ecccurve that can be bound to sslservicegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	The name of the SSL service to which the SSL policy needs to be bound. Minimum length = 1
ecccurvename	<String>	Read-write	Named ECC curve bound to servicegroup. Possible values = ALL, P_224, P_256, P_384, P_521
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslservicegroup_ecccurve_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslservicegroup_ecccurve_binding":{
    "servicegroupname":<String_value>,
    "ecccurvename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of sslservicegroup_ecccurve_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservicegroup_ecccurve_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservicegroup_ecccurve_binding": [ {
    "servicegroupname":<String_value>,
    "ecccurvename":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_ecccurve_binding/servicegroupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{"sslservicegroup_ecccurve_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservice_binding

May 20, 2016

Binding object which returns the resources bound to sslservice.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	Name of the SSL service for which to show detailed information. Minimum length = 1
sslservice_sslcipher_binding	<sslservice_sslcipher_binding[]>	Read-only	sslcipher that can be bound to sslservice.
sslservice_ecccurve_binding	<sslservice_ecccurve_binding[]>	Read-only	ecccurve that can be bound to sslservice.
sslservice_sslpolicy_binding	<sslservice_sslpolicy_binding[]>	Read-only	sslpolicy that can be bound to sslservice.
sslservice_sslcertkey_binding	<sslservice_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to sslservice.
sslservice_sslciphersuite_binding	<sslservice_sslciphersuite_binding[]>	Read-only	sslciphersuite that can be bound to sslservice.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_binding/servicename_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "sslservice_binding": [ {
    "servicename":<String_value>,
    "sslservice_sslciphersuite_binding":<sslservice_sslciphersuite_binding[]_value>,
    "sslservice_sslcertkey_binding":<sslservice_sslcertkey_binding[]_value>,
    "sslservice_sslcipher_binding":<sslservice_sslcipher_binding[]_value>,
    "sslservice_ecccurve_binding":<sslservice_ecccurve_binding[]_value>,
    "sslservice_sslpolicy_binding":<sslservice_sslpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservicegroup_sslcertkey_binding

May 20, 2016

Binding object showing the sslcertkey that can be bound to sslservicegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	The name of the SSL service to which the SSL policy needs to be bound. Minimum length = 1
ca	<Boolean>	Read-write	CA certificate.
crlcheck	<String>	Read-write	The state of the CRL check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
certkeyname	<String>	Read-write	The name of the certificate bound to the SSL service group.
snicert	<Boolean>	Read-write	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
ocspcheck	<String>	Read-write	The state of the OCSP check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslservicegroup_sslcertkey_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslservicegroup_sslcertkey_binding":{
    "servicegroupname":<String_value>,
    "certkeyname":<String_value>,
    "ca":<Boolean_value>,
    "crlcheck":<String_value>,
    "snicert":<Boolean_value>,
    "ocspcheck":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservicegroup_sslcertkey_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservicegroup_sslcertkey_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservicegroup_sslcertkey_binding": [ {
    "servicegroupname":<String_value>,
    "ca":<Boolean_value>,
    "crlcheck":<String_value>,
    "certkeyname":<String_value>,
    "snicert":<Boolean_value>,
    "ocspcheck":<String_value>
}]}
```

count

URL: http://<netscaler-ip-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

address>/nitro/v1/config/sslservicegroup_sslcertkey_binding/servicegroupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslservicegroup_sslcertkey_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservicegroup_sslcipher_binding

May 20, 2016

Binding object showing the sslcipher that can be bound to sslservicegroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	A cipher-suite can consist of an individual cipher name, the system predefined cipher-alias name, or user defined cipher-group name.
cipheraliasname	<String>	Read-write	The name of the cipher group/alias/name configured for the SSL service group.
servicegroupname	<String>	Read-write	The name of the SSL service to which the SSL policy needs to be bound. Minimum length = 1
description	<String>	Read-write	The description of the cipher.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslservicegroup_sslcipher_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslservicegroup_sslcipher_binding":{
    "servicegroupname":<String_value>,
    "ciphername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservicegroup_sslcipher_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservicegroup_sslcipher_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservicegroup_sslcipher_binding": [ {
    "ciphername":<String_value>,
    "cipheraliasname":<String_value>,
    "servicegroupname":<String_value>,
    "description":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservicegroup_sslcipher_binding/servicegroupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslservicegroup_sslcipher_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservice_ecccurve_binding

May 20, 2016

Binding object showing the ecccurve that can be bound to sslservice.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	Name of the SSL service for which to set advanced configuration. Minimum length = 1
ecccurvename	<String>	Read-write	Named ECC curve bound to service/vserver. Possible values = ALL, P_224, P_256, P_384, P_521
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslservice_ecccurve_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslservice_ecccurve_binding":{
    "servicename":<String_value>,
    "ecccurvename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservice_ecccurve_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservice_ecccurve_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservice_ecccurve_binding": [ {
    "servicename":<String_value>,
    "ecccurvename":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_ecccurve_binding/servicename_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslservice_ecccurve_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservice_sslcertkey_binding

May 20, 2016

Binding object showing the sslcertkey that can be bound to sslservice.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	Name of the SSL service for which to set advanced configuration. Minimum length = 1
certkeyname	<String>	Read-write	The certificate key pair binding.
snicert	<Boolean>	Read-write	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
skipcaname	<Boolean>	Read-write	The flag is used to indicate whether this particular CA certificate's CA_Name needs to be sent to the SSL client while requesting for client certificate in a SSL handshake.
ca	<Boolean>	Read-write	CA certificate.
crlcheck	<String>	Read-write	The state of the CRL check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
ocspcheck	<String>	Read-write	Rule to use for the OCSP responder associated with the CA certificate during client authentication. If MANDATORY is specified, deny all SSL clients if the OCSP check fails because of connectivity issues with the remote OCSP server, or any other reason that prevents the OCSP check. With the OPTIONAL setting, allow SSL clients even if the OCSP check fails except when the client certificate is revoked. Possible values = Mandatory, Optional
cleartextport	<Integer>	Read-only	The clearTextPort settings. Range 1 - 65535
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslservice_sslcertkey_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslservice_sslcertkey_binding":{
    "servicename":<String_value>,
    "certkeyname":<String_value>,
    "ca":<Boolean_value>,
    "crlcheck":<String_value>,
    "skipcaname":<Boolean_value>,
    "snicert":<Boolean_value>,
    "ocspcheck":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservice_sslcertkey_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservice_sslcertkey_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservice_sslcertkey_binding": [ {
    "servicename":<String_value>,
    "certkeyname":<String_value>,
    "snicert":<Boolean_value>,
    "skipcaname":<Boolean_value>,
    "ca":<Boolean_value>,
    "crlcheck":<String_value>,
    "ocspcheck":<String_value>,
    "cleartextport":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslcertkey_binding/servicename_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslservice_sslcertkey_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservice_sslciphersuite_binding

May 20, 2016

Binding object showing the sslciphersuite that can be bound to sslservice.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	The cipher group/alias/individual cipher configuration.
servicename	<String>	Read-write	Name of the SSL service for which to set advanced configuration. Minimum length = 1
description	<String>	Read-write	The cipher suite description.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslservice_sslciphersuite_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslservice_sslciphersuite_binding":{
    "servicename":<String_value>,
    "ciphername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservice_sslciphersuite_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservice_sslciphersuite_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservice_sslciphersuite_binding": [ {
    "ciphername":<String_value>,
    "servicename":<String_value>,
    "description":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslciphersuite_binding/servicename_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslservice_sslciphersuite_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservice_sslcipher_binding

May 20, 2016

Binding object showing the sslcipher that can be bound to sslservice.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	Name of the individual cipher, user-defined cipher group, or predefined (built-in) cipher alias.
cipheraliasname	<String>	Read-write	The cipher group/alias/individual cipher configuration.
servicename	<String>	Read-write	Name of the SSL service for which to set advanced configuration. Minimum length = 1
description	<String>	Read-write	The cipher suite description.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslservice_sslcipher_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslservice_sslcipher_binding":{
    "servicename":<String_value>,
    "ciphername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservice_sslcipher_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservice_sslcipher_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservice_sslcipher_binding": [ {
    "ciphername":<String_value>,
    "cipheraliasname":<String_value>,
    "servicename":<String_value>,
    "description":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslcipher_binding/servicename_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslservice_sslcipher_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslservice_sslpolicy_binding

May 20, 2016

Binding object showing the sslpolicy that can be bound to sslservice.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policies bound to this SSL service.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	The SSL policy binding.
servicename	<String>	Read-write	Name of the SSL service for which to set advanced configuration. Minimum length = 1
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = vserver, service, policylabel
labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag. This attribute is relevant only for ADVANCED policies.
polinherit	<Double>	Read-only	Whether the bound policy is a inherited policy or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslservice_sslpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslservice_sslpolicy_binding":{
    "servicename":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslservice_sslpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslservice_sslpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslservice_sslpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "servicename":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke":<Boolean_value>,
    "polinherit":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslservice_sslpolicy_binding/servicename_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslservice_sslpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslvserver

May 20, 2016

Configuration for SSL virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vservername	<String>	Read-write	Name of the SSL virtual server for which to set advanced configuration. Minimum length = 1
cleartextport	<Integer>	Read-write	Port on which clear-text data is sent by the appliance to the server. Do not specify this parameter for SSL offloading with end-to-end encryption. Default value: 0
dh	<String>	Read-write	State of Diffie-Hellman (DH) key exchange. Default value: DISABLED Possible values = ENABLED, DISABLED
dhfile	<String>	Read-write	Name of and, optionally, path to the DH parameter file, in PEM format, to be installed. /nsconfig/ssl/ is the default path. Minimum length = 1
dhcount	<Double>	Read-write	Number of interactions, between the client and the NetScaler appliance, after which the DH private-public pair is regenerated. A value of zero (0) specifies infinite use (no refresh). Minimum value = 0 Maximum value = 65534
dhkeyexpsizelimit	<String>	Read-write	This option enables the use of NIST recommended (NIST Special Publication 800-56A) bit size for private-key size. For example, for DH params of size 2048bit, the private-key size recommended is 224bits. This is rounded-up to 256bits. Default value: DISABLED Possible values = ENABLED, DISABLED
ersa	<String>	Read-write	State of Ephemeral RSA (eRSA) key exchange. Ephemeral RSA allows clients that support only export ciphers to communicate with the secure server even if the server certificate does not support export clients. The ephemeral RSA key is automatically generated when you bind an export cipher to an SSL or TCP-based SSL virtual server or service. When you remove the export cipher, the eRSA key is not deleted. It is reused at a later date when another export cipher is bound to an SSL or TCP-based SSL virtual server or service. The eRSA key is deleted when the appliance restarts. Default value: ENABLED Possible values = ENABLED, DISABLED
			Refresh count for regeneration of the RSA public-key and private-key pair. Zero (0)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 0 Maximum value = 65534
sessreuse	<String>	Read-write	State of session reuse. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the ENABLED setting, session key exchange is avoided for session resumption requests received from the client. Default value: ENABLED Possible values = ENABLED, DISABLED
sesstimeout	<Double>	Read-write	Time, in seconds, for which to keep the session active. Any session resumption request received after the timeout period will require a fresh SSL handshake and establishment of a new SSL session. Default value: 120 Minimum value = 0 Maximum value = 4294967294
cipherredirect	<String>	Read-write	State of Cipher Redirect. If cipher redirect is enabled, you can configure an SSL virtual server or service to display meaningful error messages if the SSL handshake fails because of a cipher mismatch between the virtual server or service and the client. Default value: DISABLED Possible values = ENABLED, DISABLED
cipherurl	<String>	Read-write	The redirect URL to be used with the Cipher Redirect feature.
sslv2redirect	<String>	Read-write	State of SSLv2 Redirect. If SSLv2 redirect is enabled, you can configure an SSL virtual server or service to display meaningful error messages if the SSL handshake fails because of a protocol version mismatch between the virtual server or service and the client. Default value: DISABLED Possible values = ENABLED, DISABLED
sslv2url	<String>	Read-write	URL of the page to which to redirect the client in case of a protocol version mismatch. Typically, this page has a clear explanation of the error or an alternative location that the transaction can continue from.
clientauth	<String>	Read-write	State of client authentication. If client authentication is enabled, the virtual server terminates the SSL handshake if the SSL client does not provide a valid certificate. Default value: DISABLED Possible values = ENABLED, DISABLED
clientcert	<String>	Read-write	Type of client authentication. If this parameter is set to MANDATORY, the appliance terminates the SSL handshake if the SSL client does not provide a valid certificate. With the OPTIONAL setting, the appliance requests a certificate from the SSL clients but proceeds with the SSL transaction even if the client presents an invalid certificate. Caution: Define proper access control policies before changing this setting to Optional. Possible values = Mandatory, Optional
			State of HTTPS redirects for the SSL virtual server. For an SSL session, if the client browser receives a redirect message, the browser

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sslredirect	<String>	Read-write	object has moved from a secure site (https://) to an unsecure site (http://). Typically, a warning message appears on the screen, prompting the user to continue or disconnect. If SSL Redirect is ENABLED, the redirect message is automatically converted from http:// to https:// and the SSL session does not break. Default value: DISABLED Possible values = ENABLED, DISABLED
redirectportrewrite	<String>	Read-write	State of the port rewrite while performing HTTPS redirect. If this parameter is ENABLED and the URL from the server does not contain the standard port, the port is rewritten to the standard. Default value: DISABLED Possible values = ENABLED, DISABLED
nonfipsciphers	<String>	Read-write	State of usage of non-FIPS approved ciphers. Valid only for an SSL service bound with a FIPS key and certificate. Default value: DISABLED Possible values = ENABLED, DISABLED
ssl2	<String>	Read-write	State of SSLv2 protocol support for the SSL Virtual Server. Default value: DISABLED Possible values = ENABLED, DISABLED
ssl3	<String>	Read-write	State of SSLv3 protocol support for the SSL Virtual Server. Default value: ENABLED Possible values = ENABLED, DISABLED
tls1	<String>	Read-write	State of TLSv1.0 protocol support for the SSL Virtual Server. Default value: ENABLED Possible values = ENABLED, DISABLED
tls11	<String>	Read-write	State of TLSv1.1 protocol support for the SSL Virtual Server. TLSv1.1 protocol is supported only on the MPX appliance. Support is not available on a FIPS appliance or on a NetScaler VPX virtual appliance. On an SDX appliance, TLSv1.1 protocol is supported only if an SSL chip is assigned to the instance. Default value: ENABLED Possible values = ENABLED, DISABLED
tls12	<String>	Read-write	State of TLSv1.2 protocol support for the SSL Virtual Server. TLSv1.2 protocol is supported only on the MPX appliance. Support is not available on a FIPS appliance or on a NetScaler VPX virtual appliance. On an SDX appliance, TLSv1.2 protocol is supported only if an SSL chip is assigned to the instance. Default value: ENABLED Possible values = ENABLED, DISABLED
snienable	<String>	Read-write	State of the Server Name Indication (SNI) feature on the virtual server and service-based offload. SNI helps to enable SSL encryption on multiple domains on a single virtual server or service if the domains are controlled by the same organization and share the same second-level domain name. For example, *.sports.net can be used to secure domains such as login.sports.net and help.sports.net. Default value: DISABLED Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
pushenctrigger	<String>	Read-write	as follows: * ALWAYS - Any PUSH packet triggers encryption. * IGNORE - Ignore PUSH packet for triggering encryption. * MERGE - For a consecutive sequence of PUSH packets, the last PUSH packet triggers encryption. * TIMER - PUSH packet triggering encryption is delayed by the time defined in the set ssl parameter command or in the Change Advanced SSL Settings dialog box. Possible values = Always, Merge, Ignore, Timer
sendclosenotify	<String>	Read-write	Enable sending SSL Close-Notify at the end of a transaction. Default value: YES Possible values = YES, NO
dtlsprofilename	<String>	Read-write	Name of the DTLS profile whose settings are to be applied to the virtual server. Minimum length = 1 Maximum length = 127
sslprofile	<String>	Read-write	Name of the SSL profile that contains SSL settings for the virtual server. Minimum length = 1 Maximum length = 127
cipherdetails	<Boolean>	Read-write	Display details of the individual ciphers bound to the SSL virtual server.
crlcheck	<String>	Read-only	The state of the CRL check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
service	<Double>	Read-only	Service.
servicename	<String>	Read-only	Service name.
ocspcheck	<String>	Read-only	The state of the OCSP check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
ca	<Boolean>	Read-only	CA certificate.
snicert	<Boolean>	Read-only	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
skipcaname	<Boolean>	Read-only	The flag is used to indicate whether this particular CA certificate's CA_Name needs to be sent to the SSL client while requesting for client certificate in a SSL handshake.
dtlsflag	<Boolean>	Read-only	The flag is used to indicate whether DTLS is set or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslvserver":{
    "vservername":<String_value>,
    "cleartextport":<Integer_value>,
    "dh":<String_value>,
    "dhfile":<String_value>,
    "dhcount":<Double_value>,
    "dhkeyexpsizelimit":<String_value>,
    "ersa":<String_value>,
    "ersacount":<Double_value>,
    "sessreuse":<String_value>,
    "sesstimeout":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cipherredirect":<String_value>,
    "cipherurl":<String_value>,
    "sslv2redirect":<String_value>,
    "sslv2url":<String_value>,
    "clientauth":<String_value>,
    "clientcert":<String_value>,
    "sslredirect":<String_value>,
    "redirectportrewrite":<String_value>,
    "nonfipsciphers":<String_value>,
    "ssl2":<String_value>,
    "ssl3":<String_value>,
    "tls1":<String_value>,
    "tls11":<String_value>,
    "tls12":<String_value>,
    "snienable":<String_value>,
    "pushenctrigger":<String_value>,
    "sendclosenotify":<String_value>,
    "dtlsprofilename":<String_value>,
    "sslprofile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslvserver":{
    "vservername":<String_value>,
    "cleartextport":true,
    "dh":true,
    "dhfile":true,
    "dhcount":true,
    "dhkeyexpsizelimit":true,
    "ersa":true,
    "ersacount":true,
    "sessreuse":true,
    "sesstimeout":true,
    "cipherredirect":true,
    "cipherurl":true,
    "sslv2redirect":true,
    "sslv2url":true,
    "clientauth":true,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "clientcert":true,
        "sslredirect":true,
        "redirectportrewrite":true,
        "nonfipsciphers":true,
        "ssl2":true,
        "ssl3":true,
        "tls1":true,
        "tls11":true,
        "tls12":true,
        "snienable":true,
        "sendclosenotify":true,
        "dtlsprofilename":true,
        "sslprofile":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/sslvserver?**args=vservername:<String_value>,cipherdetails:<Boolean_value>**

Use this query-parameter to get sslvserver resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslvserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslvserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslvserver resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslvserver": [ {
vservername:<String_value>,cipherdetails:<Boolean_value>      "cleartextport":<Integer_value>,
      "dh":<String_value>,
      "dhfile":<String_value>,
      "dhcount":<Double_value>,
      "dhkeyexpsizelimit":<String_value>,
      "ersa":<String_value>,
      "ersacount":<Double_value>,
      "sessreuse":<String_value>,
      "sesstimeout":<Double_value>,
      "cipherredirect":<String_value>,
      "crlcheck":<String_value>,
      "cipherurl":<String_value>,
      "sslv2redirect":<String_value>,
      "sslv2url":<String_value>,
      "clientauth":<String_value>,
      "clientcert":<String_value>,
      "sslredirect":<String_value>,
      "redirectportrewrite":<String_value>,
      "nonfipsciphers":<String_value>,
      "ssl2":<String_value>,
      "ssl3":<String_value>,
      "tls1":<String_value>,
      "tls11":<String_value>,
      "tls12":<String_value>,
      "snienable":<String_value>,
      "service":<Double_value>,
      "servicename":<String_value>,
      "ocspcheck":<String_value>,
      "pushenctrigger":<String_value>,
      "ca":<Boolean_value>,
      "snicert":<Boolean_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "skipcaname":<Boolean_value>,
    "sendclosenotify":<String_value>,
    "dtlsprofilename":<String_value>,
    "dtlsflag":<Boolean_value>,
    "sslprofile":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver/vservername_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslvserver/vservername_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslvserver/vservername_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslvserver": [ {
vservername:<String_value>,cipherdetails:<Boolean_value>      "cleartextport":<Integer_value>,
    "dh":<String_value>,
    "dhfile":<String_value>,
    "dhcount":<Double_value>,
    "dhkeyexpsizelimit":<String_value>,
    "ersa":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ersacount":<Double_value>,
    "sessreuse":<String_value>,
    "sesstimeout":<Double_value>,
    "cipherredirect":<String_value>,
    "crlcheck":<String_value>,
    "cipherurl":<String_value>,
    "sslv2redirect":<String_value>,
    "sslv2url":<String_value>,
    "clientauth":<String_value>,
    "clientcert":<String_value>,
    "sslredirect":<String_value>,
    "redirectportrewrite":<String_value>,
    "nonfipsciphers":<String_value>,
    "ssl2":<String_value>,
    "ssl3":<String_value>,
    "tls1":<String_value>,
    "tls11":<String_value>,
    "tls12":<String_value>,
    "snienable":<String_value>,
    "service":<Double_value>,
    "servicename":<String_value>,
    "ocspcheck":<String_value>,
    "pushenctrigger":<String_value>,
    "ca":<Boolean_value>,
    "snicert":<Boolean_value>,
    "skipcaname":<Boolean_value>,
    "sendclosenotify":<String_value>,
    "dtlsprofilename":<String_value>,
    "dtlsflag":<Boolean_value>,
    "sslprofile":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "sslvserver": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslvserver_binding

May 20, 2016

Binding object which returns the resources bound to sslvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vservername	<String>	Read-write	Name of the SSL virtual server for which to show detailed information. Minimum length = 1
sslvserver_sslpolicy_binding	<sslvserver_sslpolicy_binding[]>	Read-only	sslpolicy that can be bound to sslvserver.
sslvserver_sslcipher_binding	<sslvserver_sslcipher_binding[]>	Read-only	sslcipher that can be bound to sslvserver.
sslvserver_ecccurve_binding	<sslvserver_ecccurve_binding[]>	Read-only	ecccurve that can be bound to sslvserver.
sslvserver_sslcertkey_binding	<sslvserver_sslcertkey_binding[]>	Read-only	sslcertkey that can be bound to sslvserver.
sslvserver_sslciphersuite_binding	<sslvserver_sslciphersuite_binding[]>	Read-only	sslciphersuite that can be bound to sslvserver.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_binding/vservername_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "sslvserver_binding": [ {
    "vservername":<String_value>,
    "sslvserver_sslpolicy_binding":<sslvserver_sslpolicy_binding[]_value>,
    "sslvserver_sslcertkey_binding":<sslvserver_sslcertkey_binding[]_value>,
    "sslvserver_sslciphersuite_binding":<sslvserver_sslciphersuite_binding[]_value>,
    "sslvserver_ecccurve_binding":<sslvserver_ecccurve_binding[]_value>,
    "sslvserver_sslcipher_binding":<sslvserver_sslcipher_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslvserver_ecccurve_binding

May 20, 2016

Binding object showing the ecccurve that can be bound to sslvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ecccurvename	<String>	Read-write	Named ECC curve bound to vserver/service. Possible values = ALL, P_224, P_256, P_384, P_521
vservername	<String>	Read-write	Name of the SSL virtual server. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslvserver_ecccurve_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslvserver_ecccurve_binding":{
    "vservername":<String_value>,
    "ecccurvename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslvserver_ecccurve_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslvserver_ecccurve_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslvserver_ecccurve_binding": [ {
    "ecccurvename":<String_value>,
    "vservername":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_ecccurve_binding/vservername_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslvserver_ecccurve_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslvserver_sslcertkey_binding

May 20, 2016

Binding object showing the sslcertkey that can be bound to sslvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ca	<Boolean>	Read-write	CA certificate.
crlcheck	<String>	Read-write	The state of the CRL check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
vservername	<String>	Read-write	Name of the SSL virtual server. Minimum length = 1
certkeyname	<String>	Read-write	The name of the certificate key pair binding.
skipcaname	<Boolean>	Read-write	The flag is used to indicate whether this particular CA certificate's CA_Name needs to be sent to the SSL client while requesting for client certificate in a SSL handshake.
snicert	<Boolean>	Read-write	The name of the CertKey. Use this option to bind Certkey(s) which will be used in SNI processing.
ocspcheck	<String>	Read-write	The state of the OCSP check parameter. (Mandatory/Optional). Possible values = Mandatory, Optional
cleartextport	<Integer>	Read-only	Port on which clear-text data is sent by the appliance to the server. Do not specify this parameter for SSL offloading with end-to-end encryption. Default value: 0
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslvserver_sslcertkey_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslvserver_sslcertkey_binding":{
    "vservername":<String_value>,
    "certkeyname":<String_value>,
    "ca":<Boolean_value>,
    "crlcheck":<String_value>,
    "skipcaname":<Boolean_value>,
    "snicert":<Boolean_value>,
    "ocspcheck":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslvserver_sslcertkey_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslvserver_sslcertkey_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslvserver_sslcertkey_binding": [ {
    "ca":<Boolean_value>,
    "crlcheck":<String_value>,
    "vservername":<String_value>,
    "certkeyname":<String_value>,
    "skipcaname":<Boolean_value>,
    "snicert":<Boolean_value>,
    "ocspcheck":<String_value>,
    "cleartextport":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslcertkey_binding/vservername_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslvserver_sslcertkey_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslvserver_sslciphersuite_binding

Jul 07, 2017

Binding object showing the sslciphersuite that can be bound to sslvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	The cipher group/alias/individual cipher configuration.
description	<String>	Read-write	The cipher suite description.
vservername	<String>	Read-write	Name of the SSL virtual server. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslvserver_sslciphersuite_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslvserver_sslciphersuite_binding":{
    "vservername":<String_value>,
    "ciphername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslvserver_sslciphersuite_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslvserver_sslciphersuite_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslvserver_sslciphersuite_binding": [ {
      "ciphername":<String_value>,
      "description":<String_value>,
      "vservername":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslciphersuite_binding/vservername_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslvserver_sslciphersuite_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslvserver_sslcipher_binding

Jul 07, 2017

Binding object showing the sslcipher that can be bound to sslvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ciphername	<String>	Read-write	Name of the individual cipher, user-defined cipher group, or predefined (built-in) cipher alias.
cipheraliasname	<String>	Read-write	The name of the cipher group/alias/individual cipheri bindings.
description	<String>	Read-write	The cipher suite description.
vservername	<String>	Read-write	Name of the SSL virtual server. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslvserver_sslcipher_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslvserver_sslcipher_binding":{
    "vservername":<String_value>,
    "ciphername":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslvserver_sslcipher_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslvserver_sslcipher_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslvserver_sslcipher_binding": [ {
    "ciphername":<String_value>,
    "cipheraliasname":<String_value>,
    "description":<String_value>,
    "vservername":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslcipher_binding/vservername_value<String>?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"sslvserver_sslcipher_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslvserver_sslpolicy_binding

May 20, 2016

Binding object showing the sslpolicy that can be bound to sslvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policies bound to this SSL service.
policyname	<String>	Read-write	The name of the SSL policy binding.
labelname	<String>	Read-write	Name of the label to invoke if the current policy rule evaluates to TRUE.
vservername	<String>	Read-write	Name of the SSL virtual server. Minimum length = 1
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	Invoke flag. This attribute is relevant only for ADVANCED policies.
labeltype	<String>	Read-write	Type of policy label invocation. Possible values = vserver, service, policylabel
polinherit	<Double>	Read-only	Whether the bound policy is a inherited policy or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/sslvserver_sslpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"sslvserver_sslpolicy_binding":{
    "vservername":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of sslvserver_sslpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslvserver_sslpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslvserver_sslpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "vservername":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "polinherit":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslvserver_sslpolicy_binding/vservername_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"sslvserver_sslpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslwrapkey

May 20, 2016

Configuration for WRAP key resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
wrapkeyname	<String>	Read-write	Name for the wrap key. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the wrap key is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my key" or 'my key'). Minimum length = 1
password	<String>	Read-write	Password string for the wrap key. Minimum length = 1
salt	<String>	Read-write	Salt string for the wrap key. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

CREATE | DELETE | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

create

URL: http://<netscaler-ip-address>/nitro/v1/config/sslwrapkey?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslwrapkey":{
    "wrapkeyname":<String_value>,
    "password":<String_value>,
    "salt":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslwrapkey/wrapkeyname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslwrapkey

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslwrapkey?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslwrapkey?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslwrapkey resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslwrapkey?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslwrapkey?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslwrapkey resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "sslwrapkey": [ {
      "wrapkeyname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslwrapkey?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslwrapkey": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslaction

May 20, 2016

Configuration for SSL action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the SSL action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
clientauth	<String>	Read-write	Perform client certificate authentication. Possible values = DOCLIENTAUTH, NOCLIENTAUTH
clientcert	<String>	Read-write	Insert the entire client certificate into the HTTP header of the request being sent to the web server. The certificate is inserted in ASCII (PEM) format. Possible values = ENABLED, DISABLED
certheader	<String>	Read-write	Name of the header into which to insert the client certificate.
clientcertserialnumber	<String>	Read-write	Insert the entire client serial number into the HTTP header of the request being sent to the web server. Possible values = ENABLED, DISABLED
certserialheader	<String>	Read-write	Name of the header into which to insert the client serial number.
clientcertsubject	<String>	Read-write	Insert the client certificate subject, also known as the distinguished name (DN), into the HTTP header of the request being sent to the web server. Possible values = ENABLED, DISABLED
certsubjectheader	<String>	Read-write	Name of the header into which to insert the client certificate subject.
clientcerthash	<String>	Read-write	Insert the certificate signature (hash) into the HTTP header of the request being sent to the web server. Possible values = ENABLED, DISABLED
certhashheader	<String>	Read-write	Name of the header into which to insert the client certificate signature (hash).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
clientcertissuer	<String>	Read-write	Insert the certificate issuer details into the HTTP header of the request being sent to the web server. Possible values = ENABLED, DISABLED
certissuerheader	<String>	Read-write	Name of the header into which to insert the client certificate issuer details.
sessionid	<String>	Read-write	Insert the SSL session ID into the HTTP header of the request being sent to the web server. Every SSL connection that the client and the NetScaler share has a unique ID that identifies the specific connection. Possible values = ENABLED, DISABLED
sessionidheader	<String>	Read-write	Name of the header into which to insert the Session ID.
cipher	<String>	Read-write	Insert the cipher suite that the client and the NetScaler appliance negotiated for the SSL session into the HTTP header of the request being sent to the web server. The appliance inserts the cipher-suite name, SSL protocol, export or non-export string, and cipher strength bit, depending on the type of browser connecting to the SSL virtual server or service (for example, Cipher-Suite: RC4-MD5 SSLv3 Non-Export 128-bit). Possible values = ENABLED, DISABLED
cipherheader	<String>	Read-write	Name of the header into which to insert the name of the cipher suite.
clientcertnotbefore	<String>	Read-write	Insert the date from which the certificate is valid into the HTTP header of the request being sent to the web server. Every certificate is configured with the date and time from which it is valid. Possible values = ENABLED, DISABLED
certnotbeforeheader	<String>	Read-write	Name of the header into which to insert the date and time from which the certificate is valid.
clientcertnotafter	<String>	Read-write	Insert the date of expiry of the certificate into the HTTP header of the request being sent to the web server. Every certificate is configured with the date and time at which the certificate expires. Possible values = ENABLED, DISABLED
certnotafterheader	<String>	Read-write	Name of the header into which to insert the certificate's expiry date.
owasupport	<String>	Read-write	If the appliance is in front of an Outlook Web Access (OWA) server, insert a special header field, FRONT-END-HTTPS: ON, into the HTTP requests going to the OWA server. This header communicates to the server that the transaction is HTTPS and not HTTP. Possible values = ENABLED, DISABLED
hits	<Double>	Read-only	The number of times the action has been taken.
undefhits	<Double>	Read-only	The number of times the action resulted in UNDEF.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
description	<String>	Read-only	Description of the action.
builtin	<String[]>	Read-only	Flag to determine whether ssl action is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/sslaction

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"sslaction":{
    "name":<String_value>,
    "clientauth":<String_value>,
    "clientcert":<String_value>,
    "certheader":<String_value>,
    "clientcertserialnumber":<String_value>,
    "certserialheader":<String_value>,
    "clientcertsubject":<String_value>,
    "certsubjectheader":<String_value>,
    "clientcerthash":<String_value>,
    "certhashheader":<String_value>,
    "clientcertissuer":<String_value>,
    "certissuerheader":<String_value>,
    "sessionid":<String_value>,
    "sessionidheader":<String_value>,
    "cipher":<String_value>,
    "cipherheader":<String_value>,
    "clientcertnotbefore":<String_value>,
    "certnotbeforeheader":<String_value>,
    "clientcertnotafter":<String_value>,
    "certnotafterheader":<String_value>,
    "owasupport":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/sslaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/sslaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/sslaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of sslaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/sslaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/sslaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the sslaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslaction": [ {
    "name":<String_value>,
    "clientauth":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "clientcert":<String_value>,
    "certheader":<String_value>,
    "clientcertserialnumber":<String_value>,
    "certserialheader":<String_value>,
    "clientcertsubject":<String_value>,
    "certsubjectheader":<String_value>,
    "clientcerthash":<String_value>,
    "certhashheader":<String_value>,
    "clientcertissuer":<String_value>,
    "certissuerheader":<String_value>,
    "sessionid":<String_value>,
    "sessionidheader":<String_value>,
    "cipher":<String_value>,
    "cipherheader":<String_value>,
    "owasupport":<String_value>,
    "clientcertnotbefore":<String_value>,
    "certnotbeforeheader":<String_value>,
    "clientcertnotafter":<String_value>,
    "certnotafterheader":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "referencecount":<Double_value>,
    "description":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/sslaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/sslaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/sslaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslaction": [ {
    "name":<String_value>,
    "clientauth":<String_value>,
    "clientcert":<String_value>,
    "certheader":<String_value>,
    "clientcertserialnumber":<String_value>,
    "certserialheader":<String_value>,
    "clientcertsubject":<String_value>,
    "certsubjectheader":<String_value>,
    "clientcerthash":<String_value>,
    "certhashheader":<String_value>,
    "clientcertissuer":<String_value>,
    "certissuerheader":<String_value>,
    "sessionid":<String_value>,
    "sessionidheader":<String_value>,
    "cipher":<String_value>,
    "cipherheader":<String_value>,
    "owasupport":<String_value>,
    "clientcertnotbefore":<String_value>,
    "certnotbeforeheader":<String_value>,
    "clientcertnotafter":<String_value>,
    "certnotafterheader":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "referencecount":<Double_value>,
    "description":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/sslaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "sslaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Stream

May 20, 2016

Stream configuration.

Name	Description
streamidentifier	Configuration for identifier
streamidentifier_binding	Binding object showing the resources that can be bound to streamidentifier
streamidentifier_streamsession_binding	Binding object showing the streamsession that can be bound to streamidentifier
streamselector	Configuration for selector
streamsession	Configuration for active connection

© 1999-2017 Citrix Systems, Inc. All rights reserved.

streamidentifier

May 20, 2016

Configuration for identifier resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of stream identifier.
selectorname	<String>	Read-write	Name of the selector to use with the stream identifier. Minimum length = 1
interval	<Double>	Read-write	Number of minutes of data to use when calculating session statistics (number of requests, bandwidth, and response times). The interval is a moving window that keeps the most recently collected data. Older data is discarded at regular intervals. Default value: 1 Minimum value = 1
samplecount	<Double>	Read-write	Size of the sample from which to select a request for evaluation. The smaller the sample count, the more accurate is the statistical data. To evaluate all requests, set the sample count to 1. However, such a low setting can result in excessive consumption of memory and processing resources. Default value: 1 Minimum value = 1 Maximum value = 65535
sort	<String>	Read-write	Sort stored records by the specified statistics column, in descending order. Performed during data collection, the sorting enables real-time data evaluation through NetScaler policies (for example, compression and caching policies) that use functions such as IS_TOP(n). Default value: REQUESTS Possible values = REQUESTS, CONNECTIONS, RESPTIME, BANDWIDTH, RESPTIME_BREACHES, NONE
snmptrap	<String>	Read-write	Enable/disable SNMP trap for stream identifier. Default value: DISABLED Possible values = ENABLED, DISABLED
appflowlog	<String>	Read-write	Enable/disable Appflow logging for stream identifier. Default value: DISABLED Possible values = ENABLED, DISABLED
			Track transactions exceeding configured threshold. Transaction tracking can be enabled for following metric: ResponseTime.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: NONE Possible values = RESPTIME, NONE
maxtransactionthreshold	<Double>	Read-write	Maximum per transcation value of metric. Metric to be tracked is specified by tracktransactions attribute. Default value: 0
mintransactionthreshold	<Double>	Read-write	Minimum per transcation value of metric. Metric to be tracked is specified by tracktransactions attribute. Default value: 0
acceptancethreshold	<String>	Read-write	Non-Breaching transactions to Total transactions threshold expressed in percent. Maximum of 6 decimal places is supported. Default value: 0.000000 Maximum length = 10
breachthreshold	<Double>	Read-write	Breaching transactions threshold calculated over interval. Default value: 0
rule	<String[]>	Read-only	Rule.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/streamidentifier

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"streamidentifier":{
    "name":<String_value>,
    "selectorname":<String_value>,
    "interval":<Double_value>,
    "samplecount":<Double_value>,
    "sort":<String_value>,
    "snmptrap":<String_value>,
    "appflowlog":<String_value>,
    "tracktransactions":<String_value>,
    "maxtransactionthreshold":<Double_value>,
    "mintransactionthreshold":<Double_value>,
    "acceptancethreshold":<String_value>,
    "breachthreshold":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/streamidentifier

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"streamidentifier":{
    "name":<String_value>,
    "selectorname":<String_value>,
    "interval":<Double_value>,
    "samplecount":<Double_value>,
    "sort":<String_value>,
    "snmptrap":<String_value>,
    "appflowlog":<String_value>,
    "tracktransactions":<String_value>,
    "maxtransactionthreshold":<Double_value>,
    "mintransactionthreshold":<Double_value>,
    "acceptancethreshold":<String_value>,
    "breachthreshold":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/streamidentifier?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"streamidentifier":{
    "name":<String_value>,
    "selectorname":true,
    "interval":true,
    "samplecount":true,
    "sort":true,
    "snmptrap":true,
    "appflowlog":true,
    "tracktransactions":true,
    "maxtransactionthreshold":true,
    "mintransactionthreshold":true,
    "acceptancethreshold":true,
    "breachthreshold":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/streamidentifier/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/streamidentifier

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/streamidentifier?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/streamidentifier?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of streamidentifier resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/streamidentifier?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/streamidentifier?**pagesize=#no&pageno=#no**

Use this query-parameter to get the streamidentifier resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "streamidentifier": [ {
      "name":<String_value>,
      "selectorname":<String_value>,
      "rule":<String[]_value>,
      "interval":<Double_value>,
      "samplecount":<Double_value>,
      "sort":<String_value>,
      "snmptrap":<String_value>,
      "appflowlog":<String_value>,
      "tracktransactions":<String_value>,
      "maxtransactionthreshold":<Double_value>,
      "mintransactionthreshold":<Double_value>,
      "acceptancethreshold":<String_value>,
      "breachthreshold":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/streamidentifier/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/streamidentifier/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/streamidentifier/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "streamidentifier": [ {
      "name":<String_value>,
      "selectorname":<String_value>,
      "rule":<String[]_value>,
      "interval":<Double_value>,
      "samplecount":<Double_value>,
      "sort":<String_value>,
      "snmptrap":<String_value>,
      "appflowlog":<String_value>,
      "tracktransactions":<String_value>,
      "maxtransactionthreshold":<Double_value>,
      "mintransactionthreshold":<Double_value>,
      "acceptancethreshold":<String_value>,
      "breachthreshold":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/streamidentifier?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "streamidentifier": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

streamidentifier_binding

May 20, 2016

Binding object which returns the resources bound to streamidentifier.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of stream identifier.
streamidentifier_streamsession_binding	<streamidentifier_streamsession_binding[]>	Read-only	streamsession that can be bound to streamidentifier.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/streamidentifier_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "streamidentifier_binding": [ {
      "name":<String_value>,
      "streamidentifier_streamsession_binding":<streamidentifier_streamsession_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

streamidentifier_streamsession_binding

May 20, 2016

Binding object showing the streamsession that can be bound to streamidentifier.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of stream identifier.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/streamidentifier_streamsession_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/streamidentifier_streamsession_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of streamidentifier_streamsession_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/streamidentifier_streamsession_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the streamidentifier_streamsession_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "streamidentifier_streamsession_binding": [ {
    "name":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/streamidentifier_streamsession_binding?**args=&count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "streamidentifier_streamsession_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

streamselector

May 20, 2016

Configuration for selector resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the selector. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. If the name includes one or more spaces, and you are using the NetScaler CLI, enclose the name in double or single quotation marks (for example, "my selector" or 'my selector').
rule	<String[]>	Read-write	Set of up to five individual (not compound) default syntax expressions. Maximum length: 7499 characters. Each expression must identify a specific request characteristic, such as the client's IP address (with CLIENT.IP.SRC) or requested server resource (with HTTP.REQ.URL). Note: If two or more selectors contain the same expressions in different order, a separate set of records is created for each selector. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/streamselector

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"streamselector":{
    "name":<String_value>,
    "rule":<String[]_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/streamselector

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"streamselector":{
    "name":<String_value>,
    "rule":<String[]_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/streamselector/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/streamselector

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/streamselector?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/streamselector?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of streamselector resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/streamselector?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/streamselector?**pagesize=#no&pageno=#no**

Use this query-parameter to get the streamselector resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "streamselector": [ {
      "name":<String_value>,
      "rule":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/streamselector/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/streamselector/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/streamselector/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "streamselector": [ {
      "name":<String_value>,
      "rule":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/streamselector?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "streamselector": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

streamsession

May 20, 2016

Configuration for active connection resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the stream identifier. Minimum length = 1

Operations (click to see Properties)

CLEAR

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

clear

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/streamsession?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"streamsession":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Subscriber

May 20, 2016

Subscriber configuration

Name	Description
subscribergxinterface	Configuration for Gx interface Parameters
subscriberparam	Configuration for Subscriber Params
subscriberprofile	Configuration for Subscriber Profile
subscriberradiusinterface	Configuration for RADIUS interface Parameters
subscribersessions	Configuration for subscriber sesions

© 1999-2017 Citrix Systems, Inc. All rights reserved.

subscribergxinterface

Jul 07, 2017

Configuration for Gx interface Parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vserver	<String>	Read-write	Name of the load balancing, or content switching vserver to which the Gx connections are established. The service type of the virtual server must be DIAMETER/SSL_DIAMETER. Mutually exclusive with the service parameter. Therefore, you cannot set both service and the Virtual Server in the Gx Interface. Minimum length = 1
service	<String>	Read-write	Name of DIAMETER/SSL_DIAMETER service corresponding to PCRF to which the Gx connection is established. The service type of the service must be DIAMETER/SSL_DIAMETER. Mutually exclusive with vserver parameter. Therefore, you cannot set both Service and the Virtual Server in the Gx Interface. Minimum length = 1
pcrfrealm	<String>	Read-write	PCRF realm is of type DiameterIdentity and contains the realm of PCRF to which the message is to be routed. This is the realm used in Destination-Realm AVP by Netscaler Gx client (as a Diameter node). Minimum length = 1
holdonsubscriberabsence	<String>	Read-write	Set this setting to yes if Netscaler needs to Hold pakcets till subscriber session is fetched from PCRF. Else set to NO. By default set to yes. If this setting is set to NO, then till NetScaler fetches subscriber from PCRF, default subscriber profile will be applied to this subscriber if configured. If default subscriber profile is also not configured an undef would be raised to expressions which use Subscriber attributes. . Default value: NO Possible values = YES, NO
requesttimeout	<Double>	Read-write	q!Time, in seconds, within which the Gx CCR request must complete. If the request does not complete within this time, the request is retransmitted for requestRetryAttempts time. If still reuqest is not complete then default subscriber profile will be applied to this subscriber if configured. If default subscriber profile is also not configured an undef would be raised to expressions which use Subscriber attributes. Zero disables the timeout. !. Default value: 10 Minimum value = 0 Maximum value = 86400

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
requestretryattempts	<Double>	Read-write	retransmitted for requestRetryAttempts time. Default value: 3
idlettl	<Double>	Read-write	q!Idle Time, in seconds, after which the Gx CCR-U request will be sent after any PCRF activity on a session. Any RAR or CCA message resets the timer. Zero value disables the idle timeout. !. Default value: 900 Minimum value = 0 Maximum value = 86400
revalidationtimeout	<Double>	Read-write	q!Revalidation Timeout, in seconds, after which the Gx CCR-U request will be sent after any PCRF activity on a session. Any RAR or CCA message resets the timer. Zero value disables the idle timeout. !. Default value: 0 Minimum value = 0 Maximum value = 86400
negativettl	<Double>	Read-write	q!Negative TTL, in seconds, after which the Gx CCR-I request will be resent for sessions that have not been resolved by PCRF due to server being down or no response or failed response. Instead of polling the PCRF server constantly, negative-TTL makes NS stick to un-resolved session. Meanwhile Netscaler installs a negative session to avoid going to PCRF. For Negative Sessions, Netcaler inherits the attributes from default subscriber profile if default subscriber is configured. A default subscriber could be configured as 'add subscriber profile *'. Or these attributes can be inherited from Radius as well if Radius is configued. Zero value disables the Negative Sessions. And Netscaler does not install Negative sessions even if subscriber session could not be fetched. !. Default value: 600 Minimum value = 0 Maximum value = 86400
servicepathavp	<Double[]>	Read-write	The AVP code in which PCRF sends service path applicable for subscriber. Minimum value = 1
servicepathvendorid	<Double>	Read-write	The vendorid of the AVP in which PCRF sends service path for subscriber.
svrstate	<String>	Read-only	The state of the gx service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
identity	<String>	Read-only	DiameterIdentity to be used by NS. DiameterIdentity is used to identify a Diameter node uniquely. Before setting up diameter configuration, Netscaler (as a Diameter node) MUST be assigned a unique DiameterIdentity. example => set ns diameter -identity netscaler.com Now whenever Netscaler system needs to use identity in diameter messages. It will use 'netscaler.com' as Origin-Host AVP as defined in RFC3588

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
realm	<String>	Read-only	Diameter Realm to be used by NS. example => set ns diameter -realm com Now whenever Netscaler system needs to use realm in diameter messages. It will use 'com' as Origin-Realm AVP as defined in RFC3588 . Minimum length = 1
status	<String>	Read-only	NetScaler PCRF connection Status. (Gx Protocol State).
servicepathinfomode	<String>	Read-only	The type of info in which service path is passed from PCRF, SERVICE_FUNCTION: Denotes the service chain is passed as servicefunction names in-order in AVPS with code servicepathAVP SERVICE_PATH: Denotes the service path name is passed in AVPS with code servicepathAVP.q. Default value: SERVICEPATH Possible values = SERVICEFUNCTIONS, SERVICEPATH

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/subscribergxinterface

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"subscribergxinterface":{
    "vserver":<String_value>,
    "service":<String_value>,
    "pcrfrealm":<String_value>,
    "holdonsubscriberabsence":<String_value>,
    "requesttimeout":<Double_value>,
    "requestretryattempts":<Double_value>,
    "idlettl":<Double_value>,
    "revalidationtimeout":<Double_value>,
    "negativettl":<Double_value>,
    "servicepathavp":<Double[]_value>,
    "servicepathvendorid":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/subscribergxinterface?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"subscribergxinterface":{
    "vserver":true,
    "service":true,
    "holdonsubscriberabsence":true,
    "requesttimeout":true,
    "requestretryattempts":true,
    "idlettl":true,
    "revalidationtimeout":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "negativettl":true,
    "servicepathavp":true,
    "servicepathvendorid":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/subscribergxinterface

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "subscribergxinterface": [ {
    "vserver":<String_value>,
    "service":<String_value>,
    "svrstate":<String_value>,
    "pcrfrealm":<String_value>,
    "holdonsubscriberabsence":<String_value>,
    "requesttimeout":<Double_value>,
    "requestretryattempts":<Double_value>,
    "idlettl":<Double_value>,
    "revalidationtimeout":<Double_value>,
    "negativettl":<Double_value>,
    "identity":<String_value>,
    "realm":<String_value>,
    "status":<String_value>,
    "servicepathavp":<Double[]_value>,
    "servicepathvendorid":<Double_value>,
    "servicepathinfomode":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

subscriberparam

May 20, 2016

Configuration for Subscriber Params resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
keytype	<String>	Read-write	Type of subscriber key type IP or IPANDVLAN. Default value: IP Possible values = IP
interfacetype	<String>	Read-write	Subscriber Interface refers to Netscaler interaction with control plane protocols, RADIUS and GX. Types of subscriber interface: NONE, RadiusOnly, RadiusAndGx, GxOnly. NONE: Only static subscribers can be configured. RadiusOnly: GX interface is absent. Subscriber information is obtained through RADIUS Accounting messages. RadiusAndGx: Subscriber ID obtained through RADIUS Accounting is used to query PCRF. Subscriber information is obtained from both RADIUS and PCRF. GxOnly: RADIUS interface is absent. Subscriber information is queried using Subscriber IP. Default value: None Possible values = None, RadiusOnly, RadiusAndGx, GxOnly
idlettl	<Double>	Read-write	q!Idle Timeout, in seconds, after which Netscaler will take an idleAction on a subscriber session (refer to 'idleAction' arguement in 'set subscriber param' for more details on idleAction). Any data-plane or control plane activity updates the idleTimeout on subscriber session. idleAction could be to 'just delete the session' or 'delete and CCR-T' (if PCRF is configured) or 'do not delete but send a CCR-U'. Zero value disables the idle timeout. !. Default value: 0 Minimum value = 0 Maximum value = 172800
idleaction	<String>	Read-write	q!Once idleTTL exprires on a subscriber session, Netscaler will take an idle action on that session. idleAction could be chosen from one of these ==> 1. ccrTerminate: (default) send CCR-T to inform PCRF about session termination and delete the session. 2. delete: Just delete the subscriber session without informing PCRF. 3. ccrUpdate: Do not delete the session and instead send a CCR-U to PCRF requesting for an updated session. !. Default value: ccrTerminate Possible values = ccrTerminate, delete, ccrUpdate
			The ipv6PrefixLookupList should consist of all the ipv6 prefix lengths assigned to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum value = 1 Maximum value = 128
builtin	<String[]>	Read-only	. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"subscriberparam":{
    "keytype":<String_value>,
    "interfacetype":<String_value>,
    "idlettl":<Double_value>,
    "idleaction":<String_value>,
    "ipv6prefixlookuplist":<Double[]_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"subscriberparam":{
    "keytype":true,
    "interfacetype":true,
    "idlettl":true,
    "idleaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "subscriberparam": [ {
      "keytype":<String_value>,
      "interfacetype":<String_value>,
      "idlettl":<Double_value>,
      "idleaction":<String_value>,
      "ipv6prefixlookuplist":<Double[]_value>,
      "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

subscriberprofile

Jul 07, 2017

Configuration for Subscriber Profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ip	<String>	Read-write	Subscriber ip address.
subscriberrules	<String[]>	Read-write	Rules configured for this subscriber. This is similar to rules received from PCRF for dynamic subscriber sessions.
subscriptionidtype	<String>	Read-write	Subscription-Id type. Possible values = E164, IMSI, SIP_URI, NAI, PRIVATE
subscriptionidvalue	<String>	Read-write	Subscription-Id value.
servicepath	<String>	Read-write	Name of the servicepath to be taken for this subscriber.
flags	<Double>	Read-only	Subscriber Session flags.
ttl	<Double>	Read-only	Subscriber Session TTL.
avpdisplaybuffer	<String>	Read-only	Subscriber Attributes Display.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | UPDATE | UNSET | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"subscriberprofile":{
    "ip":<String_value>,
    "subscriberrules":<String[]_value>,
    "subscriptionidtype":<String_value>,
    "subscriptionidvalue":<String_value>,
    "servicepath":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"subscriberprofile":{
    "ip":<String_value>,
    "subscriberrules":<String[]_value>,
    "subscriptionidtype":<String_value>,
    "subscriptionidvalue":<String_value>,
    "servicepath":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"subscriberprofile":{
    "ip":<String_value>,
    "servicepath":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile/ip_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of subscriberprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the subscriberprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "subscriberprofile": [ {
    "ip":<String_value>,
    "subscriptionidtype":<String_value>,
    "subscriptionidvalue":<String_value>,
    "subscriberrules":<String[]_value>,
    "flags":<Double_value>,
    "ttl":<Double_value>,
    "avpdisplaybuffer":<String_value>,
    "servicepath":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile/ip_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile/ip_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile/ip_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "subscriberprofile": [ {
    "ip":<String_value>,
    "subscriptionidtype":<String_value>,
    "subscriptionidvalue":<String_value>,
    "subscriberrules":<String[]_value>,
    "flags":<Double_value>,
    "ttl":<Double_value>,
    "avpdisplaybuffer":<String_value>,
    "servicepath":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "subscriberprofile": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

subscriberradiusinterface

Jul 07, 2017

Configuration for RADIUS interface Parameters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
listeningservice	<String>	Read-write	Name of RADIUS LISTENING service that will process RADIUS accounting requests. Minimum length = 1
svrstate	<String>	Read-only	The state of the radius service. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED

Operations (click to see Properties)

UPDATE | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberradiusinterface

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"subscriberradiusinterface":{
    "listeningservice":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/subscriberradiusinterface

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "subscriberradiusinterface": [ {
    "listeningservice":<String_value>,
    "svrstate":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

}]}

© 1999-2017 Citrix Systems, Inc. All rights reserved.

subscribersessions

May 20, 2016

Configuration for subscriber sesions resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ip	<String>	Read-write	Subscriber IP Address.
subscriptionidtype	<String>	Read-only	Subscription-Id type. Possible values = E164, IMSI, SIP_URI, NAI, PRIVATE
subscriptionidvalue	<String>	Read-only	Subscription-Id value.
subscriberrules	<String[]>	Read-only	Rules stored in this session for this subscriber. When PCRF sends Charging-Rule-Name or Charging-Rule-Base-Name AVP for a subscriber, Netscaler stores these AVPs in the subscriber session. These Rules can be retreived using 'Subscriber.Rule_Active(<rule name>)' expression. For static subscriber profiles, these rules are configured using '-subscriberRules <list of rules>'.
flags	<Double>	Read-only	Subscriber Session flags.
ttl	<Double>	Read-only	Subscriber Session revalidation Timeout remaining. This TTL gets refreshed when a radius or CCA or RAR message is received for this subscriber session. Netscaler will send a CCR-U after revalidation timer expires. If subscriber sessions is a negative session, then Netscaler will send a CCR-I after TTL expires. Negative Sessions are sessions which have not been resolved by PCRF and instead of polling PCRF continously, Netscaler has installed a negative session. If default subscriber is configued, then Negative Sessions inherits default subscriber profile. .
idlettl	<Double>	Read-only	Subscriber Session Activity Timeout remaining. Netscaler will take an idleAction after ttl expires. idleaction could be -->\ 1. ccrTerminate: (default) send CCR-T to inform PCRF about session termination and delete the session. 2. delete: Just delete the subscriber session without informing PCRF. 3. ccrUpdate: Do not delete the session and instead send a CCR-U to PCRF requesting for an updated session. But if this is a negative session and idleaction is ccrUpdate then NS won't take any action. Also on negative sessions ccrTerminate translates to delete.
avpdisplaybuffer	<String>	Read-only	Subscriber Attributes Display.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

CLEAR | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

clear

URL: http://<netscaler-ip-address>/nitro/v1/config/subscribersessions?**action=clear**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"subscribersessions":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ip":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/subscribersessions

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/subscribersessions?**args=ip:<String_value>**

Use this query-parameter to get subscribersessions resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/subscribersessions?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/subscribersessions?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of subscribersessions resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/subscribersessions?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/subscribersessions?**pagesize=#no&pageno=#no**

Use this query-parameter to get the subscribersessions resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "subscribersessions": [ {
ip:<String_value>      "subscriptionidtype":<String_value>,
      "subscriptionidvalue":<String_value>,
      "subscriberrules":<String[]_value>,
      "flags":<Double_value>,
      "ttl":<Double_value>,
      "idlettl":<Double_value>,
      "avpdisplaybuffer":<String_value>,
      "servicepath":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/subscribersessions?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "subscribersessions": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

System

May 20, 2016

System configuration.

Name	Description
systembackup	Configuration for Backup Data for ns backup and restore
systemcmdpolicy	Configuration for command policy
systemcollectionparam	Configuration for collection parameter
systemcore	Configuration for core
systemcountergroup	Configuration for counter group
systemcounters	Configuration for counters
systemdatasource	Configuration for historical datasource
systementity	Configuration for entity
systementitydata	Configuration for entity data
systementitytype	Configuration for entity type
systemeventhistory	Configuration for event history
systemfile	Configuration for file
systemglobal_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to systemglobal
systemglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to systemglobal
systemglobal_authenticationldappolicy_binding	Binding object showing the authenticationldappolicy that can be bound to systemglobal

© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemglobal_authenticationlocalpolicy_binding	
Name	**Description**
systemglobal_authenticationpolicy_binding	Binding object showing the authenticationpolicy that can be bound to systemglobal
systemglobal_authenticationradiuspolicy_binding	Binding object showing the authenticationradiuspolicy that can be bound to systemglobal
systemglobal_authenticationtacacspolicy_binding	Binding object showing the authenticationtacacspolicy that can be bound to systemglobal
systemglobal_binding	Binding object showing the resources that can be bound to systemglobal
systemglobaldata	Configuration for global counter data
systemgroup	Configuration for system group
systemgroup_binding	Binding object showing the resources that can be bound to systemgroup
systemgroup_systemcmdpolicy_binding	Binding object showing the systemcmdpolicy that can be bound to systemgroup
systemgroup_systemuser_binding	Binding object showing the systemuser that can be bound to systemgroup
systemkek	
systemparameter	Configuration for system parameter
systemsession	Configuration for system session
systemsshkey	
systemuser	Configuration for system user
systemuser_binding	Binding object showing the resources that can be bound to systemuser
systemuser_nspartition_binding	Binding object showing the nspartition that can be bound to systemuser
systemuser_systemcmdpolicy_binding	Binding object showing the systemcmdpolicy that can be bound to systemuser
systemuser_systemgroup_binding	Binding object showing the systemgroup that can be bound to systemuser

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description

© 1999-2017 Citrix Systems, Inc. All rights reserved.

systembackup

May 20, 2016

Configuration for Backup Data for ns backup and restore resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
filename	<String>	Read-write	Name of the backup file(*.tgz) to be restored.
level	<String>	Read-write	Level of data to be backed up. Default value: basic Possible values = basic, full
comment	<String>	Read-write	Comment specified at the time of creation of the backup file(*.tgz).
skipbackup	<Boolean>	Read-write	Use this option to skip taking backup during restore operation.
size	<Double>	Read-only	Size of the backup file(*.tgz) in KB.
creationtime	<String>	Read-only	Creation time of the backup file(*.tgz).
version	<String>	Read-only	Build version of the backup file(*.tgz).
createdby	<String>	Read-only	Name of user who created the backup file(*.tgz).
ipaddress	<String>	Read-only	Ip of Netscaler box where the backup file(*.tgz) was created.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

CREATE | RESTORE | ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

create

URL: http://<netscaler-ip-address>/nitro/v1/config/systembackup?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systembackup":{
    "filename":<String_value>,
    "level":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

restore

URL: http://<netscaler-ip-address>/nitro/v1/config/systembackup?**action=restore**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systembackup":{
    "filename":<String_value>,
    "skipbackup":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

add

URL: http://<netscaler-ip-address>/nitro/v1/config/systembackup

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systembackup":{
    "filename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/systembackup/filename_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systembackup

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/systembackup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/systembackup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of systembackup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/systembackup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/systembackup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systembackup resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systembackup": [ {
    "filename":<String_value>,
    "level":<String_value>,
    "comment":<String_value>,
    "size":<Double_value>,
    "creationtime":<String_value>,
    "version":<String_value>,
    "createdby":<String_value>,
    "ipaddress":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systembackup/filename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/systembackup/filename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/systembackup/filename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "systembackup": [ {
    "filename":<String_value>,
    "level":<String_value>,
    "comment":<String_value>,
    "size":<Double_value>,
    "creationtime":<String_value>,
    "version":<String_value>,
    "createdby":<String_value>,
    "ipaddress":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systembackup?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systembackup": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemcmdpolicy

May 20, 2016

Configuration for command policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name for a command policy. Must begin with a letter, number, or the underscore (_) character, and must contain only alphanumeric, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=), colon (:), and underscore characters. Cannot be changed after the policy is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
action	<String>	Read-write	Action to perform when a request matches the policy. Possible values = ALLOW, DENY
cmdspec	<String>	Read-write	Regular expression specifying the data that matches the policy. Minimum length = 1
builtin	<String[]>	Read-only	. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemcmdpolicy":{
    "policyname":<String_value>,
    "action":<String_value>,
    "cmdspec":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy/policyname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemcmdpolicy":{
    "policyname":<String_value>,
    "action":<String_value>,
    "cmdspec":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of systemcmdpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemcmdpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemcmdpolicy": [ {
    "policyname":<String_value>,
    "action":<String_value>,
    "cmdspec":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy/policyname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy/policyname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy/policyname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemcmdpolicy": [ {
    "policyname":<String_value>,
    "action":<String_value>,
    "cmdspec":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcmdpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "systemcmdpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemcollectionparam

Jul 07, 2017

Configuration for collection parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
communityname	<String>	Read-write	SNMPv1 community name for authentication.
loglevel	<String>	Read-write	specify the log level. Possible values CRITICAL,WARNING,INFO,DEBUG1,DEBUG2. Minimum length = 1
datapath	<String>	Read-write	specify the data path to the database. Minimum length = 1

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcollectionparam

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemcollectionparam":{
    "communityname":<String_value>,
    "loglevel":<String_value>,
    "datapath":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcollectionparam?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemcollectionparam":{
    "communityname":true,
    "loglevel":true,
    "datapath":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcollectionparam

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemcollectionparam": [ {
    "communityname":<String_value>,
    "loglevel":<String_value>,
    "datapath":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemcore

Jul 07, 2017

Configuration for core resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcore

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/systemcore?**args=datasource:<String_value>**

Use this query-parameter to get systemcore resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemcore": [ {
datasource:<String_value>      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemcountergroup

Jul 07, 2017

Configuration for counter group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcountergroup

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/systemcountergroup?**args=datasource:<String_value>**

Use this query-parameter to get systemcountergroup resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemcountergroup": [ {
datasource:<String_value>      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemcounters

May 20, 2016

Configuration for counters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
countergroup	<String>	Read-write	Specify the (counter) group name which contains all the counters specific tot his particular group.
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemcounters

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/systemcounters?**args=countergroup:<String_value>,datasource:<String_value>**

Use this query-parameter to get systemcounters resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemcounters": [ {
countergroup:<String_value>,datasource:<String_value>      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemdatasource

May 20, 2016

Configuration for historical datasource resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemdatasource

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/systemdatasource?**args=datasource:<String_value>**

Use this query-parameter to get systemdatasource resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemdatasource": [ {
datasource:<String_value>      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systementity

May 20, 2016

Configuration for entity resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
type	<String>	Read-write	Specify the entity type.
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
core	<Integer>	Read-write	Specify core ID of the PE in nCore.
response	<String>	Read-only	.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systementity

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/systementity?**args=type:<String_value>,datasource:<String_value>,core:<Integer_value>**

Use this query-parameter to get systementity resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systementity": [ {
type:<String_value>,datasource:<String_value>,core:<Integer_value>      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systementitydata

May 20, 2016

Configuration for entity data resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
type	<String>	Read-write	Specify the entity type.
name	<String>	Read-write	Specify the entity name.
alldeleted	<Boolean>	Read-write	Specify this if you would like to delete information about all deleted entities from the database.
allinactive	<Boolean>	Read-write	Specify this if you would like to delete information about all inactive entities from the database.
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
core	<Integer>	Read-write	Specify core ID of the PE in nCore.
counters	<String>	Read-write	Specify the counters to be collected.
starttime	<String>	Read-write	Specify start time in mmddyyyyhhmm to start collecting values from that timestamp.
endtime	<String>	Read-write	Specify end time in mmddyyyyhhmm upto which values have to be collected.
last	<Integer>	Read-write	Last is literal way of saying a certain time period from the current moment. Example: -last 1 hour, -last 1 day, et cetera. Default value: 1
unit	<String>	Read-write	Specify the time period from current moment. Example 1 x where x = hours/ days/ years. Possible values = HOURS, DAYS, MONTHS
response	<String>	Read-only	.
startupdate	<Double>	Read-only	.
lastupdate	<Double>	Read-only	.

Operations (click to see Properties)

DELETE | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN= *<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/systementitydata

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systementitydata

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/systementitydata?**args=type:<String_value>,name:<String_value>,counters:<String_value>,starttime:<String_value>,endtime:<String_value>,last:<Integer_value>,unit:<String_value>,datasource:<String_value>,core:<Integer_value>**

Use this query-parameter to get systementitydata resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systementitydata": [ {
type:<String_value>,name:<String_value>,counters:
<String_value>,starttime:<String_value>,endtime:<String_value>,last:<Integer_value>,unit:<String_value>,datasource:<String_value>,core:<Integer_value>      "response":<String_value>,
    "startupdate":<Double_value>,
    "lastupdate":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

systementitytype

May 20, 2016

Configuration for entity type resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systementitytype

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/systementitytype?**args=datasource:<String_value>**

Use this query-parameter to get systementitytype resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systementitytype": [ {
datasource:<String_value>      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemeventhistory

May 20, 2016

Configuration for event history resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
starttime	<String>	Read-write	Specify start time in mmddyyyyhhmm to start collecting values from that timestamp.
endtime	<String>	Read-write	Specify end time in mmddyyyyhhmm upto which values have to be collected.
last	<Integer>	Read-write	Last is literal way of saying a certain time period from the current moment. Example: -last 1 hour, -last 1 day, et cetera. Default value: 1
unit	<String>	Read-write	Specify the time period from current moment. Example 1 x where x = hours/ days/ years. Possible values = HOURS, DAYS, MONTHS
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
response	<String>	Read-only	.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemeventhistory

Query-parameters:

args

http://<netscaler-ip-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

address>/nitro/v1/config/systemeventhistory?**args=starttime:<String_value>,endtime:<String_value>,last:<Integer_value>,unit:<String_value>,datasource:<String_value>,**

Use this query-parameter to get systemeventhistory resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemeventhistory": [ {
starttime:<String_value>,endtime:<String_value>,last:<Integer_value>,unit:<String_value>,datasource:<String_value>,      "response":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemfile

Jul 07, 2017

Configuration for file resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
filename	<String>	Read-write	Name of the file. It should not include filepath. Maximum length = 63
filecontent	<String>	Read-write	file content in Base64 format.
filelocation	<String>	Read-write	location of the file on NetScaler. Maximum length = 127
fileencoding	<String>	Read-write	encoding type of the file content. Default value: "BASE64"
fileaccesstime	<String>	Read-only	Last access time of the file.
filemodifiedtime	<String>	Read-only	last modified time of the file.
filemode	<String[]>	Read-only	file mode. Possible values = DIRECTORY
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/systemfile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemfile":{
    "filename":<String_value>,
    "filecontent":<String_value>,
    "filelocation":<String_value>,
    "fileencoding":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/systemfile/filename_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemfile

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/systemfile?**args=filename:<String_value>,filelocation:<String_value>**

Use this query-parameter to get systemfile resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemfile": [ {
filename:<String_value>,filelocation:<String_value>      "filecontent":<String_value>,
      "fileencoding":<String_value>,
      "fileaccesstime":<String_value>,
      "filemodifiedtime":<String_value>,
      "filemode":<String[]_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemfile/filename_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemfile": [ {
filename:<String_value>,filelocation:<String_value>      "filecontent":<String_value>,
      "fileencoding":<String_value>,
      "fileaccesstime":<String_value>,
      "filemodifiedtime":<String_value>,
      "filemode":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemglobaldata

May 20, 2016

Configuration for global counter data resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
counters	<String>	Read-write	Specify the counters to be collected.
countergroup	<String>	Read-write	Specify the (counter) group name which contains all the counters specific to this particular group.
starttime	<String>	Read-write	Specify start time in mmddyyyyhhmm to start collecting values from that timestamp.
endtime	<String>	Read-write	Specify end time in mmddyyyyhhmm upto which values have to be collected.
last	<Integer>	Read-write	Last is literal way of saying a certain time period from the current moment. Example: -last 1 hour, -last 1 day, et cetera. Default value: 1
unit	<String>	Read-write	Specify the time period from current moment. Example 1 x where x = hours/ days/ years. Possible values = HOURS, DAYS, MONTHS
datasource	<String>	Read-write	Specifies the source which contains all the stored counter values.
core	<Integer>	Read-write	Specify core ID of the PE in nCore.
response	<String>	Read-only	.
startupdate	<Double>	Read-only	.
lastupdate	<Double>	Read-only	.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobaldata

Query-parameters:

args

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/systemglobaldata?**args=counters:<String_value>,countergroup:<String_value>,starttime:<String_value>,endtime:<String_value>,last:<Integer_value>,unit:<String_value>,datasource:<String_value>,core:<Integer**

Use this query-parameter to get systemglobaldata resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemglobaldata": [ {
counters:<String_value>,countergroup:<String_value>,starttime:<String_value>,endtime:<String_value>,last:<Integer_value>,unit:<String_value>,datasource:<String_value>,core:<Integer_value>      "response":<String_value>,
    "startupdate":<Double_value>,
    "lastupdate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemglobal_auditnslogpolicy_binding

May 20, 2016

Binding object showing the auditnslogpolicy that can be bound to systemglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the command policy.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemglobal_auditnslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemglobal_auditnslogpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_auditnslogpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_auditnslogpolicy_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_auditnslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_auditnslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_auditnslogpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemglobal_auditnslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemglobal_auditnslogpolicy_binding": [ {
    "priority":<Double_value>,
    "builtin":<String[]_value>,
    "policyname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_auditnslogpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "systemglobal_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemkek

May 20, 2016

Configuration for 0 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
passphrase	<String>	Read-write	Passphrase required to generate the key encryption key. Minimum length = 8 Maximum length = 32
password	<String>	Read-write	Password required to import the key encryption key. Minimum length = 8 Maximum length = 32

Operations (click to see Properties)

CREATE | EXPORT | IMPORT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

create

URL: http://<netscaler-ip-address>/nitro/v1/config/systemkek?**action=create**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemkek":{
    "passphrase":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

export

URL: http://<netscaler-ip-address>/nitro/v1/config/systemkek?**action=export**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemkek":{
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Import

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/systemkek?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemkek":{
    "password":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemglobal_auditsyslogpolicy_binding

May 20, 2016

Binding object showing the auditsyslogpolicy that can be bound to systemglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the command policy.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemglobal_auditsyslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemglobal_auditsyslogpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemglobal_auditsyslogpolicy_binding": [ {
    "priority":<Double_value>,
    "builtin":<String[]_value>,
    "policyname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "systemglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemglobal_authenticationldappolicy_binding

May 20, 2016

Binding object showing the authenticationldappolicy that can be bound to systemglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the command policy.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemglobal_authenticationldappolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemglobal_authenticationldappolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationldappolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationldappolicy_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationldappolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_authenticationldappolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationldappolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemglobal_authenticationldappolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemglobal_authenticationldappolicy_binding": [ {
    "priority":<Double_value>,
    "builtin":<String[]_value>,
    "policyname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationldappolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "systemglobal_authenticationldappolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemglobal_authenticationlocalpolicy_binding

May 20, 2016

Binding object showing the authenticationlocalpolicy that can be bound to systemglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the command policy.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemglobal_authenticationlocalpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_authenticationlocalpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemglobal_authenticationlocalpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemglobal_authenticationlocalpolicy_binding": [ {
    "priority":<Double_value>,
    "builtin":<String[]_value>,
    "policyname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationlocalpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "systemglobal_authenticationlocalpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemglobal_authenticationpolicy_binding

May 20, 2016

Binding object showing the authenticationpolicy that can be bound to systemglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the command policy.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemglobal_authenticationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemglobal_authenticationpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationpolicy_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_authenticationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemglobal_authenticationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemglobal_authenticationpolicy_binding": [ {
    "priority":<Double_value>,
    "builtin":<String[]_value>,
    "policyname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "systemglobal_authenticationpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemglobal_authenticationradiuspolicy_binding

May 20, 2016

Binding object showing the authenticationradiuspolicy that can be bound to systemglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the command policy.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemglobal_authenticationradiuspolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_authenticationradiuspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemglobal_authenticationradiuspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemglobal_authenticationradiuspolicy_binding": [ {
    "priority":<Double_value>,
    "builtin":<String[]_value>,
    "policyname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationradiuspolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "systemglobal_authenticationradiuspolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemglobal_authenticationtacacspolicy_binding

May 20, 2016

Binding object showing the authenticationtacacspolicy that can be bound to systemglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the command policy.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	The name of the command policy.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemglobal_authenticationtacacspolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemglobal_authenticationtacacspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemglobal_authenticationtacacspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemglobal_authenticationtacacspolicy_binding": [ {
    "priority":<Double_value>,
    "builtin":<String[]_value>,
    "policyname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_authenticationtacacspolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "systemglobal_authenticationtacacspolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemglobal_binding

May 20, 2016

Binding object which returns the resources bound to systemglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
systemglobal_authenticationradiuspolicy_binding	<systemglobal_authenticationradiuspolicy_binding[]>	Read-only	authenticationradiuspolicy that can be bound to systemglobal.
systemglobal_authenticationldappolicy_binding	<systemglobal_authenticationldappolicy_binding[]>	Read-only	authenticationldappolicy that can be bound to systemglobal.
systemglobal_authenticationlocalpolicy_binding	<systemglobal_authenticationlocalpolicy_binding[]>	Read-only	authenticationlocalpolicy that can be bound to systemglobal.
systemglobal_auditsyslogpolicy_binding	<systemglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to systemglobal.
systemglobal_auditnslogpolicy_binding	<systemglobal_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to systemglobal.
systemglobal_authenticationpolicy_binding	<systemglobal_authenticationpolicy_binding[]>	Read-only	authenticationpolicy that can be bound to systemglobal.
systemglobal_authenticationtacacspolicy_binding	<systemglobal_authenticationtacacspolicy_binding[]>	Read-only	authenticationtacacspolicy that can be bound to systemglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "systemglobal_binding": [ {
    "systemglobal_authenticationldappolicy_binding":<systemglobal_authenticationldappolicy_binding[]_value>,
    "systemglobal_auditsyslogpolicy_binding":<systemglobal_auditsyslogpolicy_binding[]_value>,
    "systemglobal_authenticationlocalpolicy_binding":<systemglobal_authenticationlocalpolicy_binding[]_value>,
    "systemglobal_auditnslogpolicy_binding":<systemglobal_auditnslogpolicy_binding[]_value>,
    "systemglobal_authenticationtacacspolicy_binding":<systemglobal_authenticationtacacspolicy_binding[]_value>,
    "systemglobal_authenticationpolicy_binding":<systemglobal_authenticationpolicy_binding[]_value>,
    "systemglobal_authenticationradiuspolicy_binding":<systemglobal_authenticationradiuspolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemgroup

May 20, 2016

Configuration for system group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
groupname	<String>	Read-write	Name for the group. Must begin with a letter, number, or the underscore (_) character, and must contain only alphanumeric, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=), colon (:), and underscore characters. Cannot be changed after the group is created. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my group" or 'my group'). Minimum length = 1
promptstring	<String>	Read-write	String to display at the command-line prompt. Can consist of letters, numbers, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=), colon (:), underscore (_), and the following variables: * %u - Will be replaced by the user name. * %h - Will be replaced by the hostname of the NetScaler appliance. * %t - Will be replaced by the current time in 12-hour format. * %T - Will be replaced by the current time in 24-hour format. * %d - Will be replaced by the current date. * %s - Will be replaced by the state of the NetScaler appliance. Note: The 63-character limit for the length of the string does not apply to the characters that replace the variables. Minimum length = 1
timeout	<Double>	Read-write	CLI session inactivity timeout, in seconds. If Restrictedtimeout argument of system parameter is enabled, Timeout can have values in the range [300-86400] seconds.If Restrictedtimeout argument of system parameter is disabled, Timeout can have values in the range [0, 10-100000000] seconds. Default value is 900 seconds.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemgroup":{
    "groupname":<String_value>,
    "promptstring":<String_value>,
    "timeout":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemgroup":{
    "groupname":<String_value>,
    "promptstring":<String_value>,
    "timeout":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"systemgroup":{
    "groupname":<String_value>,
    "promptstring":true,
    "timeout":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/systemgroup?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/systemgroup?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of systemgroup resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/systemgroup?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemgroup?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemgroup resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemgroup": [ {
      "groupname":<String_value>,
      "promptstring":<String_value>,
      "timeout":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup/groupname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/systemgroup/groupname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/systemgroup/groupname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemgroup": [ {
      "groupname":<String_value>,
      "promptstring":<String_value>,
      "timeout":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemgroup": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemgroup_binding

May 20, 2016

Binding object which returns the resources bound to systemgroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
groupname	<String>	Read-write	Name of the system group about which to display information. Minimum length = 1
systemgroup_systemuser_binding	<systemgroup_systemuser_binding[]>	Read-only	systemuser that can be bound to systemgroup.
systemgroup_systemcmdpolicy_binding	<systemgroup_systemcmdpolicy_binding[]>	Read-only	systemcmdpolicy that can be bound to systemgroup.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup_binding/groupname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "systemgroup_binding": [ {
    "groupname":<String_value>,
    "systemgroup_systemcmdpolicy_binding":<systemgroup_systemcmdpolicy_binding[]_value>,
    "systemgroup_systemuser_binding":<systemgroup_systemuser_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemgroup_systemcmdpolicy_binding

May 20, 2016

Binding object showing the systemcmdpolicy that can be bound to systemgroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the command policy.
policyname	<String>	Read-write	The name of command policy.
groupname	<String>	Read-write	Name of the system group. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemgroup_systemcmdpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemgroup_systemcmdpolicy_binding":{
    "groupname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>

Query-parameters:

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemgroup_systemcmdpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemgroup_systemcmdpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemgroup_systemcmdpolicy_binding": [ {
      "priority":<Double_value>,
      "policyname":<String_value>,
      "groupname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup_systemcmdpolicy_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"systemgroup_systemcmdpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemgroup_systemuser_binding

Jul 07, 2017

Binding object showing the systemuser that can be bound to systemgroup.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	The system user.
groupname	<String>	Read-write	Name of the system group. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemgroup_systemuser_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemgroup_systemuser_binding":{
    "groupname":<String_value>,
    "username":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>

Query-parameters:

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemgroup_systemuser_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemgroup_systemuser_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemgroup_systemuser_binding": [ {
    "username":<String_value>,
    "groupname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemgroup_systemuser_binding/groupname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"systemgroup_systemuser_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemparameter

May 20, 2016

Configuration for system parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
rbaonresponse	<String>	Read-write	Enable or disable Role-Based Authentication (RBA) on responses. Default value: ENABLED Possible values = ENABLED, DISABLED
promptstring	<String>	Read-write	String to display at the command-line prompt. Can consist of letters, numbers, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=), colon (:), underscore (_), and the following variables: * %u - Will be replaced by the user name. * %h - Will be replaced by the hostname of the NetScaler appliance. * %t - Will be replaced by the current time in 12-hour format. * %T - Will be replaced by the current time in 24-hour format. * %d - Will be replaced by the current date. * %s - Will be replaced by the state of the NetScaler appliance. Note: The 63-character limit for the length of the string does not apply to the characters that replace the variables. Minimum length = 1
natpcbforceflushlimit	<Double>	Read-write	Flush the system if the number of Network Address Translation Protocol Control Blocks (NATPCBs) exceeds this value. Default value: 2147483647 Minimum value = 1000
natpcbrstontimeout	<String>	Read-write	Send a reset signal to client and server connections when their NATPCBs time out. Avoids the buildup of idle TCP connections on both the sides. Default value: DISABLED Possible values = ENABLED, DISABLED
timeout	<Double>	Read-write	CLI session inactivity timeout, in seconds. If Restrictedtimeout argument is enabled, Timeout can have values in the range [300-86400] seconds. If Restrictedtimeout argument is disabled, Timeout can have values in the range [0, 10-100000000] seconds. Default value is 900 seconds.
localauth	<String>	Read-write	When enabled, local users can access NetScaler even when external authentication is configured. When disabled, local users are not allowed to access the NetScaler, Local users can access the NetScaler only when the configured external authentication servers are unavailable. Default value: ENABLED Possible values = ENABLED, DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
minpasswordlen	<Double>	Read-write	default minimum length is 4. User entered value can be greater than or equal to 4. Default minimum value is 1 when strong password is disabled. Maximum value is 127 in both cases. Minimum value = 1 Maximum value = 127
strongpassword	<String>	Read-write	Option to enable/disable strong password. When this is enabled, system user password should contain atleast one lower case character, one upper case character, one numeric character and one special character from the set (!, @, #, (,), $, %, ^, &, *). Default value: disabled Possible values = enableall, enablelocal, disabled
restrictedtimeout	<String>	Read-write	Enable/Disable the restricted timeout behaviour. When enabled, timeout cannot be configured beyond admin configured timeout and also it will have the [minimum - maximum] range check. When disabled, timeout will have the old behaviour. By default the value is disabled. Default value: DISABLED Possible values = ENABLED, DISABLED
fipsusermode	<String>	Read-write	Use this option to set the FIPS mode for key user-land processes. When enabled, these user-land processes will operate in FIPS mode. In this mode, theses processes will use FIPS 140-2 Level-1 certified crypto algorithms. Default is disabled, wherein, these user-land processes will not operate in FIPS mode. Default value: DISABLED Possible values = ENABLED, DISABLED
doppler	<String>	Read-write	Enable or disable Doppler. Default value: 0 Possible values = ENABLED, DISABLED
googleanalytics	<String>	Read-write	Enable or disable Google analytics. Default value: DISABLED Possible values = ENABLED, DISABLED
cliloglevel	<String>	Read-write	Audit log level, which specifies the types of events to log for cli executed commands. Available values function as follows: * EMERGENCY - Events that indicate an immediate crisis on the server. * ALERT - Events that might require action. * CRITICAL - Events that indicate an imminent server crisis. * ERROR - Events that indicate some type of error. * WARNING - Events that require action in the near future. * NOTICE - Events that the administrator should know about. * INFORMATIONAL - All but low-level events. * DEBUG - All events, in extreme detail. Default value: DEFAULT_LOGLEVEL_CLI Possible values = EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, NOTICE, INFORMATIONAL, DEBUG
maxclient	<Double>	Read-only	Maximum number of client connection allowed by the system. Minimum value = 20 Maximum value = 40

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/systemparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemparameter":{
    "rbaonresponse":<String_value>,
    "promptstring":<String_value>,
    "natpcbforceflushlimit":<Double_value>,
    "natpcbrstontimeout":<String_value>,
    "timeout":<Double_value>,
    "localauth":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "minpasswordlen":<Double_value>,
    "strongpassword":<String_value>,
    "restrictedtimeout":<String_value>,
    "fipsusermode":<String_value>,
    "doppler":<String_value>,
    "googleanalytics":<String_value>,
    "cliloglevel":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/systemparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemparameter":{
    "minpasswordlen":true,
    "rbaonresponse":true,
    "promptstring":true,
    "natpcbforceflushlimit":true,
    "natpcbrstontimeout":true,
    "timeout":true,
    "localauth":true,
    "strongpassword":true,
    "restrictedtimeout":true,
    "fipsusermode":true,
    "doppler":true,
    "googleanalytics":true,
    "cliloglevel":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/systemparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemparameter": [ {
      "rbaonresponse":<String_value>,
      "promptstring":<String_value>,
      "natpcbforceflushlimit":<Double_value>,
      "natpcbrstontimeout":<String_value>,
      "timeout":<Double_value>,
      "maxclient":<Double_value>,
      "localauth":<String_value>,
      "minpasswordlen":<Double_value>,
      "strongpassword":<String_value>,
      "restrictedtimeout":<String_value>,
      "fipsusermode":<String_value>,
      "doppler":<String_value>,
      "googleanalytics":<String_value>,
      "cliloglevel":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemsession

May 20, 2016

Configuration for system session resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sid	<Double>	Read-write	ID of the system session about which to display information. Minimum value = 1
all	<Boolean>	Read-write	Terminate all the system sessions except the current session.
username	<String>	Read-only	user name of the session.
logintime	<String>	Read-only	logged-in time of this session.
lastactivitytime	<String>	Read-only	last activity time of on this session.
expirytime	<Integer>	Read-only	Time left in expire the session in seconds.
numofconnections	<Double>	Read-only	number of connection using this token.
currentconn	<Boolean>	Read-only	True if the token is used for current session.
clienttype	<String>	Read-only	client type of the session. Possible values = CLI, API, GUI
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

KILL | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

kill

URL: http://<netscaler-ip-address>/nitro/v1/config/systemsession?**action=kill**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemsession":{
    "sid":<Double_value>,
    "all":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemsession

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/systemsession?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/systemsession?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of systemsession resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/systemsession?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemsession?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemsession resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemsession": [ {
    "sid":<Double_value>,
    "username":<String_value>,
    "logintime":<String_value>,
    "lastactivitytime":<String_value>,
    "expirytime":<Integer_value>,
    "numofconnections":<Double_value>,
    "currentconn":<Boolean_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "clienttype":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemsession/sid_value<Double>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/systemsession/sid_value<Double>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/systemsession/sid_value<Double>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemsession": [ {
    "sid":<Double_value>,
    "username":<String_value>,
    "logintime":<String_value>,
    "lastactivitytime":<String_value>,
    "expirytime":<Integer_value>,
    "numofconnections":<Double_value>,
    "currentconn":<Boolean_value>,
    "clienttype":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemsession?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemsession": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemsshkey

Jul 07, 2017

Configuration for 0 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	URL \(protocol, host, path, and file name\) from where the location file will be imported. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 255
sshkeytype	<String>	Read-write	The type of the ssh key whether public or private key. Possible values = PRIVATE, PUBLIC
src	<String>	Read-write	URL \(protocol, host, path, and file name\) from where the location file will be imported. NOTE: The import fails if the object to be imported is on an HTTPS server that requires client certificate authentication for access. Minimum length = 1 Maximum length = 2047

Operations (click to see Properties)

DELETE | IMPORT | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/systemsshkey/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Import

URL: http://<netscaler-ip-address>/nitro/v1/config/systemsshkey?**action=Import**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemsshkey":{
    "name":<String_value>,
    "src":<String_value>,
    "sshkeytype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/systemsshkey

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/systemsshkey?**args=sshkeytype:<String_value>**

Use this query-parameter to get systemsshkey resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemsshkey": [ {
sshkeytype:<String_value>      "name":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemuser

Jul 07, 2017

Configuration for system user resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	Name for a user. Must begin with a letter, number, or the underscore (_) character, and must contain only alphanumeric, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=), colon (:), and underscore characters. Cannot be changed after the user is added. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my user" or 'my user'). Minimum length = 1
password	<String>	Read-write	Password for the system user. Can include any ASCII character. Minimum length = 1
externalauth	<String>	Read-write	Whether to use external authentication servers for the system user authentication or not. Default value: ENABLED Possible values = ENABLED, DISABLED
promptstring	<String>	Read-write	String to display at the command-line prompt. Can consist of letters, numbers, hyphen (-), period (.), hash (#), space ( ), at (@), equal (=), colon (:), underscore (_), and the following variables: * %u - Will be replaced by the user name. * %h - Will be replaced by the hostname of the NetScaler appliance. * %t - Will be replaced by the current time in 12-hour format. * %T - Will be replaced by the current time in 24-hour format. * %d - Will be replaced by the current date. * %s - Will be replaced by the state of the NetScaler appliance. Note: The 63-character limit for the length of the string does not apply to the characters that replace the variables. Minimum length = 1
timeout	<Double>	Read-write	CLI session inactivity timeout, in seconds. If Restrictedtimeout argument of system parameter is enabled, Timeout can have values in the range [300-86400] seconds. If Restrictedtimeout argument of system parameter is disabled, Timeout can have values in the range [0, 10-100000000] seconds. Default value is 900 seconds.
logging	<String>	Read-write	Users logging privilege. Default value: DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
maxsession	<Double>	Read-write	Maximum number of client connection allowed per user. Default value: 20 Minimum value = 1 Maximum value = 40
encrypted	<Boolean>	Read-only	.
promptinheritedfrom	<String>	Read-only	From where the prompt has been inherited. Possible values = User, Group, Global, Climode
timeoutkind	<String>	Read-only	From where the timeout has been inherited. Possible values = User, Group, Global, Climode
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemuser":{
    "username":<String_value>,
    "password":<String_value>,
    "externalauth":<String_value>,
    "promptstring":<String_value>,
    "timeout":<Double_value>,
    "logging":<String_value>,
    "maxsession":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser

HTTP Method: PUT

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemuser":{
    "username":<String_value>,
    "password":<String_value>,
    "externalauth":<String_value>,
    "promptstring":<String_value>,
    "timeout":<Double_value>,
    "logging":<String_value>,
    "maxsession":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"systemuser":{
    "username":<String_value>,
    "externalauth":true,
    "promptstring":true,
    "timeout":true,
    "logging":true,
    "maxsession":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/systemuser?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/systemuser?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of systemuser resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/systemuser?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemuser?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemuser resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemuser": [ {
    "username":<String_value>,
    "password":<String_value>,
    "encrypted":<Boolean_value>,
    "externalauth":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "promptstring":<String_value>,
    "promptinheritedfrom":<String_value>,
    "timeout":<Double_value>,
    "timeoutkind":<String_value>,
    "logging":<String_value>,
    "maxsession":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser/username_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/systemuser/username_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/systemuser/username_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemuser": [ {
    "username":<String_value>,
    "password":<String_value>,
    "encrypted":<Boolean_value>,
    "externalauth":<String_value>,
    "promptstring":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "promptinheritedfrom":<String_value>,
    "timeout":<Double_value>,
    "timeoutkind":<String_value>,
    "logging":<String_value>,
    "maxsession":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemuser": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemuser_binding

Jul 07, 2017

Binding object which returns the resources bound to systemuser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	Name of a system user about whom to display information. Minimum length = 1
systemuser_systemgroup_binding	<systemuser_systemgroup_binding[]>	Read-only	systemgroup that can be bound to systemuser.
systemuser_nspartition_binding	<systemuser_nspartition_binding[]>	Read-only	nspartition that can be bound to systemuser.
systemuser_systemcmdpolicy_binding	<systemuser_systemcmdpolicy_binding[]>	Read-only	systemcmdpolicy that can be bound to systemuser.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser_binding/username_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "systemuser_binding": [ {
      "username":<String_value>,
      "systemuser_systemgroup_binding":<systemuser_systemgroup_binding[]_value>,
      "systemuser_nspartition_binding":<systemuser_nspartition_binding[]_value>,
      "systemuser_systemcmdpolicy_binding":<systemuser_systemcmdpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemuser_nspartition_binding

May 20, 2016

Binding object showing the nspartition that can be bound to systemuser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	Name of the system-user entry to which to bind the command policy. Minimum length = 1
partitionname	<String>	Read-write	Name of the Partition to bind to the system user.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemuser_nspartition_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemuser_nspartition_binding":{
    "username":<String_value>,
    "partitionname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemuser_nspartition_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemuser_nspartition_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemuser_nspartition_binding": [ {
    "username":<String_value>,
    "partitionname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser_nspartition_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"systemuser_nspartition_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemuser_systemcmdpolicy_binding

May 20, 2016

Binding object showing the systemcmdpolicy that can be bound to systemuser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of command policy.
username	<String>	Read-write	Name of the system-user entry to which to bind the command policy. Minimum length = 1
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/systemuser_systemcmdpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"systemuser_systemcmdpolicy_binding":{
    "username":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemuser_systemcmdpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemuser_systemcmdpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemuser_systemcmdpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "username":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser_systemcmdpolicy_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"systemuser_systemcmdpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemuser_systemgroup_binding

May 20, 2016

Binding object showing the systemgroup that can be bound to systemuser.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	Name of the system-user entry to which to bind the command policy. Minimum length = 1
groupname	<String>	Read-write	The system group. .
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser_systemgroup_binding/username_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/systemuser_systemgroup_binding/username_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of systemuser_systemgroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/systemuser_systemgroup_binding/username_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the systemuser_systemgroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemuser_systemgroup_binding": [ {
      "username":<String_value>,
      "groupname":<String_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/systemuser_systemgroup_binding/username_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"systemuser_systemgroup_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Traffic Management

May 20, 2016

TM session/policy configuration.

Name	Description
tmformssoaction	Configuration for Form sso action
tmglobal_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to tmglobal
tmglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to tmglobal
tmglobal_binding	Binding object showing the resources that can be bound to tmglobal
tmglobal_tmsessionpolicy_binding	Binding object showing the tmsessionpolicy that can be bound to tmglobal
tmglobal_tmtrafficpolicy_binding	Binding object showing the tmtrafficpolicy that can be bound to tmglobal
tmsamlssoprofile	Configuration for SAML sso action
tmsessionaction	Configuration for TM session action
tmsessionparameter	Configuration for session parameter
tmsessionpolicy	Configuration for TM session policy
tmsessionpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to tmsessionpolicy
tmsessionpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to tmsessionpolicy
tmsessionpolicy_authenticationvserver_binding	Binding object showing the authenticationvserver that can be bound to tmsessionpolicy
tmsessionpolicy_binding	Binding object showing the resources that can be bound to tmsessionpolicy
tmsessionpolicy_tmglobal_binding	Binding object showing the tmglobal that can be bound to tmsessionpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmtrafficaction	
Name	**Description**
tmtrafficpolicy	Configuration for TM traffic policy
tmtrafficpolicy_binding	Binding object showing the resources that can be bound to tmtrafficpolicy
tmtrafficpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to tmtrafficpolicy
tmtrafficpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to tmtrafficpolicy
tmtrafficpolicy_tmglobal_binding	Binding object showing the tmglobal that can be bound to tmtrafficpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmformssoaction

Jul 07, 2017

Configuration for Form sso action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the new form-based single sign-on profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an SSO action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
actionurl	<String>	Read-write	URL to which the completed form is submitted. Minimum length = 1
userfield	<String>	Read-write	Name of the form field in which the user types in the user ID. Minimum length = 1
passwdfield	<String>	Read-write	Name of the form field in which the user types in the password. Minimum length = 1
ssosuccessrule	<String>	Read-write	Expression, that checks to see if single sign-on is successful.
namevaluepair	<String>	Read-write	Name-value pair attributes to send to the server in addition to sending the username and password. Value names are separated by an ampersand (&) (for example, name1=value1&name2=value2).
responsesize	<Double>	Read-write	Number of bytes, in the response, to parse for extracting the forms. Default value: 8096
nvtype	<String>	Read-write	Type of processing of the name-value pair. If you specify STATIC, the values configured by the administrator are used. For DYNAMIC, the response is parsed, and the form is extracted and then submitted. Default value: DYNAMIC Possible values = STATIC, DYNAMIC
submitmethod	<String>	Read-write	HTTP method used by the single sign-on form to send the logon credentials to the logon server. Applies only to STATIC name-value type. Default value: GET Possible values = GET, POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmformssoaction":{
    "name":<String_value>,
    "actionurl":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "userfield":<String_value>,
    "passwdfield":<String_value>,
    "ssosuccessrule":<String_value>,
    "namevaluepair":<String_value>,
    "responsesize":<Double_value>,
    "nvtype":<String_value>,
    "submitmethod":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmformssoaction":{
    "name":<String_value>,
    "actionurl":<String_value>,
    "userfield":<String_value>,
    "passwdfield":<String_value>,
    "ssosuccessrule":<String_value>,
    "responsesize":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "namevaluepair":<String_value>,
    "nvtype":<String_value>,
    "submitmethod":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmformssoaction":{
    "name":<String_value>,
    "responsesize":true,
    "namevaluepair":true,
    "nvtype":true,
    "submitmethod":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmformssoaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmformssoaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmformssoaction": [ {
      "name":<String_value>,
      "actionurl":<String_value>,
      "userfield":<String_value>,
      "passwdfield":<String_value>,
      "responsesize":<Double_value>,
      "namevaluepair":<String_value>,
      "nvtype":<String_value>,
      "ssosuccessrule":<String_value>,
      "submitmethod":<String_value>
}]}
```

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmformssoaction": [ {
      "name":<String_value>,
      "actionurl":<String_value>,
      "userfield":<String_value>,
      "passwdfield":<String_value>,
      "responsesize":<Double_value>,
      "namevaluepair":<String_value>,
      "nvtype":<String_value>,
      "ssosuccessrule":<String_value>,
      "submitmethod":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmformssoaction?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "tmformssoaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmglobal_auditnslogpolicy_binding

May 20, 2016

Binding object showing the auditnslogpolicy that can be bound to tmglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance tmsession policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	The name of the policy.
bindpolicytype	<Double>	Read-only	Bound policy type.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/tmglobal_auditnslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"tmglobal_auditnslogpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_auditnslogpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_auditnslogpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmglobal_auditnslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmglobal_auditnslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmglobal_auditnslogpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmglobal_auditnslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "tmglobal_auditnslogpolicy_binding": [ {
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "policyname":<String_value>,
      "bindpolicytype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_auditnslogpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmglobal_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmglobal_auditsyslogpolicy_binding

Jul 07, 2017

Binding object showing the auditsyslogpolicy that can be bound to tmglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
gotopriorityexpression	<String>	Read-write	Applicable only to advance tmsession policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	The name of the policy.
bindpolicytype	<Double>	Read-only	Bound policy type.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/tmglobal_auditsyslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"tmglobal_auditsyslogpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmglobal_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "tmglobal_auditsyslogpolicy_binding": [ {
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "policyname":<String_value>,
      "bindpolicytype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmglobal_binding

May 20, 2016

Binding object which returns the resources bound to tmglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
tmglobal_tmtrafficpolicy_binding	<tmglobal_tmtrafficpolicy_binding[]>	Read-only	tmtrafficpolicy that can be bound to tmglobal.
tmglobal_auditsyslogpolicy_binding	<tmglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to tmglobal.
tmglobal_auditnslogpolicy_binding	<tmglobal_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to tmglobal.
tmglobal_tmsessionpolicy_binding	<tmglobal_tmsessionpolicy_binding[]>	Read-only	tmsessionpolicy that can be bound to tmglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "tmglobal_binding": [ {
      "tmglobal_tmtrafficpolicy_binding":<tmglobal_tmtrafficpolicy_binding[]_value>,
      "tmglobal_auditnslogpolicy_binding":<tmglobal_auditnslogpolicy_binding[]_value>,
      "tmglobal_auditsyslogpolicy_binding":<tmglobal_auditsyslogpolicy_binding[]_value>,
      "tmglobal_tmsessionpolicy_binding":<tmglobal_tmsessionpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmglobal_tmsessionpolicy_binding

May 20, 2016

Binding object showing the tmsessionpolicy that can be bound to tmglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	The name of the policy.
bindpolicytype	<Double>	Read-only	Bound policy type.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/tmglobal_tmsessionpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"tmglobal_tmsessionpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_tmsessionpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_tmsessionpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmglobal_tmsessionpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmglobal_tmsessionpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmglobal_tmsessionpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmglobal_tmsessionpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmglobal_tmsessionpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "builtin":<String[]_value>,
    "policyname":<String_value>,
    "bindpolicytype":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_tmsessionpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "tmglobal_tmsessionpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmglobal_tmtrafficpolicy_binding

May 20, 2016

Binding object showing the tmtrafficpolicy that can be bound to tmglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
gotopriorityexpression	<String>	Read-write	Applicable only to advance tmsession policy. Expression or other value specifying the next policy to be evaluated if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * A default syntax expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a priority number that is numerically higher than the highest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policyname	<String>	Read-write	The name of the policy.
type	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
bindpolicytype	<Double>	Read-only	Bound policy type.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/tmglobal_tmtrafficpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"tmglobal_tmtrafficpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmglobal_tmtrafficpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmglobal_tmtrafficpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmglobal_tmtrafficpolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "type":<String_value>,
    "bindpolicytype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmglobal_tmtrafficpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmglobal_tmtrafficpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmsamlssoprofile

Jul 07, 2017

Configuration for SAML sso action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the new saml single sign-on profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an SSO action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
samlsigningcertname	<String>	Read-write	Name of the SSL certificate that is used to Sign Assertion. Minimum length = 1
assertionconsumerserviceurl	<String>	Read-write	URL to which the assertion is to be sent. Minimum length = 1
relaystaterule	<String>	Read-write	Expression to extract relaystate to be sent along with assertion. Evaluation of this expression should return TEXT content. This is typically a targ et url to which user is redirected after the recipient validates SAML token.
sendpassword	<String>	Read-write	Option to send password in assertion. Default value: OFF Possible values = ON, OFF
samlissuername	<String>	Read-write	The name to be used in requests sent from Netscaler to IdP to uniquely identify Netscaler. Minimum length = 1
signaturealg	<String>	Read-write	Algorithm to be used to sign/verify SAML transactions. Default value: RSA-SHA1 Possible values = RSA-SHA1, RSA-SHA256
digestmethod	<String>	Read-write	Algorithm to be used to compute/verify digest for SAML transactions. Default value: SHA1 Possible values = SHA1, SHA256
			Audience for which assertion sent by IdP is applicable. This is typically

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
nameidformat	<String>	Read-write	Format of Name Identifier sent in Assertion. Default value: transient Possible values = Unspecified, emailAddress, X509SubjectName, WindowsDomainQualifiedName, kerberos, entity, persistent, transient
nameidexpr	<String>	Read-write	Expression that will be evaluated to obtain NameIdentifier to be sent in assertion. Maximum length = 128
attribute1	<String>	Read-write	Name of attribute1 that needs to be sent in SAML Assertion.
attribute1expr	<String>	Read-write	Expression that will be evaluated to obtain attribute1's value to be sent in Assertion. Maximum length = 128
attribute1friendlyname	<String>	Read-write	User-Friendly Name of attribute1 that needs to be sent in SAML Assertion.
attribute1format	<String>	Read-write	Format of Attribute1 to be sent in Assertion. Possible values = URI, Basic
attribute2	<String>	Read-write	Name of attribute2 that needs to be sent in SAML Assertion.
attribute2expr	<String>	Read-write	Expression that will be evaluated to obtain attribute2's value to be sent in Assertion. Maximum length = 128
attribute2friendlyname	<String>	Read-write	User-Friendly Name of attribute2 that needs to be sent in SAML Assertion.
attribute2format	<String>	Read-write	Format of Attribute2 to be sent in Assertion. Possible values = URI, Basic
attribute3	<String>	Read-write	Name of attribute3 that needs to be sent in SAML Assertion.
attribute3expr	<String>	Read-write	Expression that will be evaluated to obtain attribute3's value to be sent in Assertion. Maximum length = 128
attribute3friendlyname	<String>	Read-write	User-Friendly Name of attribute3 that needs to be sent in SAML Assertion.
attribute3format	<String>	Read-write	Format of Attribute3 to be sent in Assertion. Possible values = URI, Basic
attribute4	<String>	Read-write	Name of attribute4 that needs to be sent in SAML Assertion.
attribute4expr	<String>	Read-write	Expression that will be evaluated to obtain attribute4's value to be sent in Assertion.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute4friendlyname	<String>	Read-write	User-Friendly Name of attribute4 that needs to be sent in SAML Assertion.
attribute4format	<String>	Read-write	Format of Attribute4 to be sent in Assertion. Possible values = URI, Basic
attribute5	<String>	Read-write	Name of attribute5 that needs to be sent in SAML Assertion.
attribute5expr	<String>	Read-write	Expression that will be evaluated to obtain attribute5's value to be sent in Assertion. Maximum length = 128
attribute5friendlyname	<String>	Read-write	User-Friendly Name of attribute5 that needs to be sent in SAML Assertion.
attribute5format	<String>	Read-write	Format of Attribute5 to be sent in Assertion. Possible values = URI, Basic
attribute6	<String>	Read-write	Name of attribute6 that needs to be sent in SAML Assertion.
attribute6expr	<String>	Read-write	Expression that will be evaluated to obtain attribute6's value to be sent in Assertion. Maximum length = 128
attribute6friendlyname	<String>	Read-write	User-Friendly Name of attribute6 that needs to be sent in SAML Assertion.
attribute6format	<String>	Read-write	Format of Attribute6 to be sent in Assertion. Possible values = URI, Basic
attribute7	<String>	Read-write	Name of attribute7 that needs to be sent in SAML Assertion.
attribute7expr	<String>	Read-write	Expression that will be evaluated to obtain attribute7's value to be sent in Assertion. Maximum length = 128
attribute7friendlyname	<String>	Read-write	User-Friendly Name of attribute7 that needs to be sent in SAML Assertion.
attribute7format	<String>	Read-write	Format of Attribute7 to be sent in Assertion. Possible values = URI, Basic
attribute8	<String>	Read-write	Name of attribute8 that needs to be sent in SAML Assertion.
attribute8expr	<String>	Read-write	Expression that will be evaluated to obtain attribute8's value to be sent in Assertion. Maximum length = 128

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute8format	<String>	Read-write	Format of Attribute8 to be sent in Assertion. Possible values = URI, Basic
attribute9	<String>	Read-write	Name of attribute9 that needs to be sent in SAML Assertion.
attribute9expr	<String>	Read-write	Expression that will be evaluated to obtain attribute9's value to be sent in Assertion. Maximum length = 128
attribute9friendlyname	<String>	Read-write	User-Friendly Name of attribute9 that needs to be sent in SAML Assertion.
attribute9format	<String>	Read-write	Format of Attribute9 to be sent in Assertion. Possible values = URI, Basic
attribute10	<String>	Read-write	Name of attribute10 that needs to be sent in SAML Assertion.
attribute10expr	<String>	Read-write	Expression that will be evaluated to obtain attribute10's value to be sent in Assertion. Maximum length = 128
attribute10friendlyname	<String>	Read-write	User-Friendly Name of attribute10 that needs to be sent in SAML Assertion.
attribute10format	<String>	Read-write	Format of Attribute10 to be sent in Assertion. Possible values = URI, Basic
attribute11	<String>	Read-write	Name of attribute11 that needs to be sent in SAML Assertion.
attribute11expr	<String>	Read-write	Expression that will be evaluated to obtain attribute11's value to be sent in Assertion. Maximum length = 128
attribute11friendlyname	<String>	Read-write	User-Friendly Name of attribute11 that needs to be sent in SAML Assertion.
attribute11format	<String>	Read-write	Format of Attribute11 to be sent in Assertion. Possible values = URI, Basic
attribute12	<String>	Read-write	Name of attribute12 that needs to be sent in SAML Assertion.
attribute12expr	<String>	Read-write	Expression that will be evaluated to obtain attribute12's value to be sent in Assertion. Maximum length = 128
attribute12friendlyname	<String>	Read-write	User-Friendly Name of attribute12 that needs to be sent in SAML Assertion.
attribute12format	<String>	Read-write	Format of Attribute12 to be sent in Assertion.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute13	<String>	Read-write	Name of attribute13 that needs to be sent in SAML Assertion.
attribute13expr	<String>	Read-write	Expression that will be evaluated to obtain attribute13's value to be sent in Assertion. Maximum length = 128
attribute13friendlyname	<String>	Read-write	User-Friendly Name of attribute13 that needs to be sent in SAML Assertion.
attribute13format	<String>	Read-write	Format of Attribute13 to be sent in Assertion. Possible values = URI, Basic
attribute14	<String>	Read-write	Name of attribute14 that needs to be sent in SAML Assertion.
attribute14expr	<String>	Read-write	Expression that will be evaluated to obtain attribute14's value to be sent in Assertion. Maximum length = 128
attribute14friendlyname	<String>	Read-write	User-Friendly Name of attribute14 that needs to be sent in SAML Assertion.
attribute14format	<String>	Read-write	Format of Attribute14 to be sent in Assertion. Possible values = URI, Basic
attribute15	<String>	Read-write	Name of attribute15 that needs to be sent in SAML Assertion.
attribute15expr	<String>	Read-write	Expression that will be evaluated to obtain attribute15's value to be sent in Assertion. Maximum length = 128
attribute15friendlyname	<String>	Read-write	User-Friendly Name of attribute15 that needs to be sent in SAML Assertion.
attribute15format	<String>	Read-write	Format of Attribute15 to be sent in Assertion. Possible values = URI, Basic
attribute16	<String>	Read-write	Name of attribute16 that needs to be sent in SAML Assertion.
attribute16expr	<String>	Read-write	Expression that will be evaluated to obtain attribute16's value to be sent in Assertion. Maximum length = 128
attribute16friendlyname	<String>	Read-write	User-Friendly Name of attribute16 that needs to be sent in SAML Assertion.
attribute16format	<String>	Read-write	Format of Attribute16 to be sent in Assertion. Possible values = URI, Basic
			Option to encrypt assertion when Netscaler sends one.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = ON, OFF
samlspcertname	<String>	Read-write	Name of the SSL certificate of peer/receving party using which Assertion is encrypted. Minimum length = 1
encryptionalgorithm	<String>	Read-write	Algorithm to be used to encrypt SAML assertion. Default value: AES256 Possible values = DES3, AES128, AES192, AES256
skewtime	<Double>	Read-write	This option specifies the number of minutes on either side of current time that the assertion would be valid. For example, if skewTime is 10, then assertion would be valid from (current time - 10) min to (current time + 10) min, ie 20min in all. Default value: 5
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmsamlssoprofile":{
    "name":<String_value>,
    "samlsigningcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "relaystaterule":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute1expr":<String_value>,
    "attribute1friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2":<String_value>,
    "attribute2expr":<String_value>,
    "attribute2friendlyname":<String_value>,
    "attribute2format":<String_value>,
    "attribute3":<String_value>,
    "attribute3expr":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute3format":<String_value>,
    "attribute4":<String_value>,
    "attribute4expr":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute4format":<String_value>,
    "attribute5":<String_value>,
    "attribute5expr":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute5format":<String_value>,
    "attribute6":<String_value>,
    "attribute6expr":<String_value>,
    "attribute6friendlyname":<String_value>,
    "attribute6format":<String_value>,
    "attribute7":<String_value>,
    "attribute7expr":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute7format":<String_value>,
    "attribute8":<String_value>,
    "attribute8expr":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute8format":<String_value>,
    "attribute9":<String_value>,
    "attribute9expr":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute9friendlyname":<String_value>,
    "attribute9format":<String_value>,
    "attribute10":<String_value>,
    "attribute10expr":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute10format":<String_value>,
    "attribute11":<String_value>,
    "attribute11expr":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute11format":<String_value>,
    "attribute12":<String_value>,
    "attribute12expr":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute12format":<String_value>,
    "attribute13":<String_value>,
    "attribute13expr":<String_value>,
    "attribute13friendlyname":<String_value>,
    "attribute13format":<String_value>,
    "attribute14":<String_value>,
    "attribute14expr":<String_value>,
    "attribute14friendlyname":<String_value>,
    "attribute14format":<String_value>,
    "attribute15":<String_value>,
    "attribute15expr":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute15format":<String_value>,
    "attribute16":<String_value>,
    "attribute16expr":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute16format":<String_value>,
    "encryptassertion":<String_value>,
    "samlspcertname":<String_value>,
    "encryptionalgorithm":<String_value>,
    "skewtime":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

update

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmsamlssoprofile":{
    "name":<String_value>,
    "samlsigningcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "relaystaterule":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute1expr":<String_value>,
    "attribute1friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2":<String_value>,
    "attribute2expr":<String_value>,
    "attribute2friendlyname":<String_value>,
    "attribute2format":<String_value>,
    "attribute3":<String_value>,
    "attribute3expr":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute3format":<String_value>,
    "attribute4":<String_value>,
    "attribute4expr":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute4format":<String_value>,
    "attribute5":<String_value>,
    "attribute5expr":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute5format":<String_value>,
    "attribute6":<String_value>,
    "attribute6expr":<String_value>,
    "attribute6friendlyname":<String_value>,
    "attribute6format":<String_value>,
    "attribute7":<String_value>,
    "attribute7expr":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute7format":<String_value>,
    "attribute8":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute8expr":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute8format":<String_value>,
    "attribute9":<String_value>,
    "attribute9expr":<String_value>,
    "attribute9friendlyname":<String_value>,
    "attribute9format":<String_value>,
    "attribute10":<String_value>,
    "attribute10expr":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute10format":<String_value>,
    "attribute11":<String_value>,
    "attribute11expr":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute11format":<String_value>,
    "attribute12":<String_value>,
    "attribute12expr":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute12format":<String_value>,
    "attribute13":<String_value>,
    "attribute13expr":<String_value>,
    "attribute13friendlyname":<String_value>,
    "attribute13format":<String_value>,
    "attribute14":<String_value>,
    "attribute14expr":<String_value>,
    "attribute14friendlyname":<String_value>,
    "attribute14format":<String_value>,
    "attribute15":<String_value>,
    "attribute15expr":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute15format":<String_value>,
    "attribute16":<String_value>,
    "attribute16expr":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute16format":<String_value>,
    "encryptassertion":<String_value>,
    "samlspcertname":<String_value>,
    "encryptionalgorithm":<String_value>,
    "skewtime":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"tmsamlssoprofile":{
    "name":<String_value>,
    "samlsigningcertname":true,
    "sendpassword":true,
    "samlissuername":true,
    "signaturealg":true,
    "digestmethod":true,
    "audience":true,
    "nameidformat":true,
    "nameidexpr":true,
    "attribute1":true,
    "attribute1friendlyname":true,
    "attribute1format":true,
    "attribute2":true,
    "attribute2friendlyname":true,
    "attribute2format":true,
    "attribute3":true,
    "attribute3friendlyname":true,
    "attribute3format":true,
    "attribute4":true,
    "attribute4friendlyname":true,
    "attribute4format":true,
    "attribute5":true,
    "attribute5friendlyname":true,
    "attribute5format":true,
    "attribute6":true,
    "attribute6friendlyname":true,
    "attribute6format":true,
    "attribute7":true,
    "attribute7friendlyname":true,
    "attribute7format":true,
    "attribute8":true,
    "attribute8friendlyname":true,
    "attribute8format":true,
    "attribute9":true,
    "attribute9friendlyname":true,
    "attribute9format":true,
    "attribute10":true,
    "attribute10friendlyname":true,
    "attribute10format":true,
    "attribute11":true,
    "attribute11friendlyname":true,
    "attribute11format":true,
    "attribute12":true,
    "attribute12friendlyname":true,
    "attribute12format":true,
    "attribute13":true,
    "attribute13friendlyname":true,
    "attribute13format":true,
    "attribute14":true,
    "attribute14friendlyname":true,
    "attribute14format":true,
    "attribute15":true,
    "attribute15friendlyname":true,
    "attribute15format":true,
    "attribute16":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "attribute16friendlyname":true,
        "attribute16format":true,
        "encryptassertion":true,
        "samlspcertname":true,
        "encryptionalgorithm":true,
        "skewtime":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmsamlssoprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmsamlssoprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsamlssoprofile": [ {
    "name":<String_value>,
    "samlsigningcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "relaystaterule":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>,
    "attribute1friendlyname":<String_value>,
    "attribute2friendlyname":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute6friendlyname":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute9friendlyname":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute13friendlyname":<String_value>,
    "attribute14friendlyname":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2format":<String_value>,
    "attribute3format":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute4format":<String_value>,
    "attribute5format":<String_value>,
    "attribute6format":<String_value>,
    "attribute7format":<String_value>,
    "attribute8format":<String_value>,
    "attribute9format":<String_value>,
    "attribute10format":<String_value>,
    "attribute11format":<String_value>,
    "attribute12format":<String_value>,
    "attribute13format":<String_value>,
    "attribute14format":<String_value>,
    "attribute15format":<String_value>,
    "attribute16format":<String_value>,
    "attribute1expr":<String_value>,
    "attribute2expr":<String_value>,
    "attribute3expr":<String_value>,
    "attribute4expr":<String_value>,
    "attribute5expr":<String_value>,
    "attribute6expr":<String_value>,
    "attribute7expr":<String_value>,
    "attribute8expr":<String_value>,
    "attribute9expr":<String_value>,
    "attribute10expr":<String_value>,
    "attribute11expr":<String_value>,
    "attribute12expr":<String_value>,
    "attribute13expr":<String_value>,
    "attribute14expr":<String_value>,
    "attribute15expr":<String_value>,
    "attribute16expr":<String_value>,
    "encryptassertion":<String_value>,
    "samlspcertname":<String_value>,
    "encryptionalgorithm":<String_value>,
    "skewtime":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsamlssoprofile": [ {
    "name":<String_value>,
    "samlsigningcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "relaystaterule":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>,
    "attribute1friendlyname":<String_value>,
    "attribute2friendlyname":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute6friendlyname":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute9friendlyname":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute13friendlyname":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute14friendlyname":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2format":<String_value>,
    "attribute3format":<String_value>,
    "attribute4format":<String_value>,
    "attribute5format":<String_value>,
    "attribute6format":<String_value>,
    "attribute7format":<String_value>,
    "attribute8format":<String_value>,
    "attribute9format":<String_value>,
    "attribute10format":<String_value>,
    "attribute11format":<String_value>,
    "attribute12format":<String_value>,
    "attribute13format":<String_value>,
    "attribute14format":<String_value>,
    "attribute15format":<String_value>,
    "attribute16format":<String_value>,
    "attribute1expr":<String_value>,
    "attribute2expr":<String_value>,
    "attribute3expr":<String_value>,
    "attribute4expr":<String_value>,
    "attribute5expr":<String_value>,
    "attribute6expr":<String_value>,
    "attribute7expr":<String_value>,
    "attribute8expr":<String_value>,
    "attribute9expr":<String_value>,
    "attribute10expr":<String_value>,
    "attribute11expr":<String_value>,
    "attribute12expr":<String_value>,
    "attribute13expr":<String_value>,
    "attribute14expr":<String_value>,
    "attribute15expr":<String_value>,
    "attribute16expr":<String_value>,
    "encryptassertion":<String_value>,
    "samlspcertname":<String_value>,
    "encryptionalgorithm":<String_value>,
    "skewtime":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsamlssoprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "tmsamlssoprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmsessionaction

May 20, 2016

Configuration for TM session action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the session action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after a session action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
sesstimeout	<Double>	Read-write	Session timeout, in minutes. If there is no traffic during the timeout period, the user is disconnected and must reauthenticate to access intranet resources. Minimum value = 1
defaultauthorizationaction	<String>	Read-write	Allow or deny access to content for which there is no specific authorization policy. Possible values = ALLOW, DENY
sso	<String>	Read-write	Use single sign-on (SSO) to log users on to all web applications automatically after they authenticate, or pass users to the web application logon page to authenticate to each application individually. Default value: OFF Possible values = ON, OFF
ssocredential	<String>	Read-write	Use the primary or secondary authentication credentials for single sign-on (SSO). Possible values = PRIMARY, SECONDARY
ssodomain	<String>	Read-write	Domain to use for single sign-on (SSO). Minimum length = 1 Maximum length = 32
httponlycookie	<String>	Read-write	Allow only an HTTP session cookie, in which case the cookie cannot be accessed by scripts. Possible values = YES, NO
kcdaccount	<String>	Read-write	Kerberos constrained delegation account name. Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
persistentcookie	<String>	Read-write	Enable or disable persistent SSO cookies for the traffic management (TM) session. A persistent cookie remains on the user device and is sent with each HTTP request. The cookie becomes stale if the session ends. This setting is overwritten if a traffic action sets persistent cookie to OFF. Note: If persistent cookie is enabled, make sure you set the persistent cookie validity. Possible values = ON, OFF
persistentcookievalidity	<Double>	Read-write	Integer specifying the number of minutes for which the persistent cookie remains valid. Can be set only if the persistent cookie setting is enabled. Minimum value = 1
homepage	<String>	Read-write	Web address of the home page that a user is displayed when authentication vserver is bookmarked and used to login.
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmsessionaction":{
    "name":<String_value>,
    "sesstimeout":<Double_value>,
    "defaultauthorizationaction":<String_value>,
    "sso":<String_value>,
    "ssocredential":<String_value>,
    "ssodomain":<String_value>,
    "httponlycookie":<String_value>,
    "kcdaccount":<String_value>,
    "persistentcookie":<String_value>,
    "persistentcookievalidity":<Double_value>,
    "homepage":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmsessionaction":{
    "name":<String_value>,
    "sesstimeout":<Double_value>,
    "defaultauthorizationaction":<String_value>,
    "sso":<String_value>,
    "ssocredential":<String_value>,
    "ssodomain":<String_value>,
    "kcdaccount":<String_value>,
    "httponlycookie":<String_value>,
    "persistentcookie":<String_value>,
    "persistentcookievalidity":<Double_value>,
    "homepage":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmsessionaction":{
    "name":<String_value>,
    "sesstimeout":true,
    "defaultauthorizationaction":true,
    "sso":true,
    "ssocredential":true,
    "ssodomain":true,
    "kcdaccount":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "httponlycookie":true,
        "persistentcookie":true,
        "persistentcookievalidity":true,
        "homepage":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmsessionaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmsessionaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionaction": [ {
      "name":<String_value>,
      "sesstimeout":<Double_value>,
      "defaultauthorizationaction":<String_value>,
      "sso":<String_value>,
      "ssocredential":<String_value>,
      "ssodomain":<String_value>,
      "kcdaccount":<String_value>,
      "httponlycookie":<String_value>,
      "persistentcookie":<String_value>,
      "persistentcookievalidity":<Double_value>,
      "homepage":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionaction": [ {
    "name":<String_value>,
    "sesstimeout":<Double_value>,
    "defaultauthorizationaction":<String_value>,
    "sso":<String_value>,
    "ssocredential":<String_value>,
    "ssodomain":<String_value>,
    "kcdaccount":<String_value>,
    "httponlycookie":<String_value>,
    "persistentcookie":<String_value>,
    "persistentcookievalidity":<Double_value>,
    "homepage":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmsessionparameter

May 20, 2016

Configuration for session parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sesstimeout	<Double>	Read-write	Session timeout, in minutes. If there is no traffic during the timeout period, the user is disconnected and must reauthenticate to access the intranet resources. Default value: 30 Minimum value = 1
defaultauthorizationaction	<String>	Read-write	Allow or deny access to content for which there is no specific authorization policy. Default value: ALLOW Possible values = ALLOW, DENY
sso	<String>	Read-write	Log users on to all web applications automatically after they authenticate, or pass users to the web application logon page to authenticate for each application. Default value: OFF Possible values = ON, OFF
ssocredential	<String>	Read-write	Use primary or secondary authentication credentials for single sign-on. Default value: PRIMARY Possible values = PRIMARY, SECONDARY
ssodomain	<String>	Read-write	Domain to use for single sign-on. Minimum length = 1 Maximum length = 32
kcdaccount	<String>	Read-write	Kerberos constrained delegation account name. Minimum length = 1 Maximum length = 32
httponlycookie	<String>	Read-write	Allow only an HTTP session cookie, in which case the cookie cannot be accessed by scripts. Default value: YES Possible values = YES, NO
persistentcookie	<String>	Read-write	Use persistent SSO cookies for the traffic session. A persistent cookie remains on the user device and is sent with each HTTP request. The cookie becomes stale if the session ends. Default value: OFF Possible values = ON, OFF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
persistentcookievalidity	<Double>	Read-write	remains valid. Can be set only if the persistence cookie setting is enabled. Minimum value = 1
homepage	<String>	Read-write	Web address of the home page that a user is displayed when authentication vserver is bookmarked and used to login. Default value: "None"
name	<String>	Read-only	.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionparameter

HTTP Method: PUT

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmsessionparameter":{
    "sesstimeout":<Double_value>,
    "defaultauthorizationaction":<String_value>,
    "sso":<String_value>,
    "ssocredential":<String_value>,
    "ssodomain":<String_value>,
    "kcdaccount":<String_value>,
    "httponlycookie":<String_value>,
    "persistentcookie":<String_value>,
    "persistentcookievalidity":<Double_value>,
    "homepage":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionparameter?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmsessionparameter":{
    "sesstimeout":true,
    "sso":true,
    "ssodomain":true,
    "kcdaccount":true,
    "persistentcookie":true,
    "homepage":true,
    "defaultauthorizationaction":true,
    "ssocredential":true,
    "httponlycookie":true,
    "persistentcookievalidity":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionparameter": [ {
    "name":<String_value>,
    "sesstimeout":<Double_value>,
    "defaultauthorizationaction":<String_value>,
    "sso":<String_value>,
    "ssocredential":<String_value>,
    "ssodomain":<String_value>,
    "kcdaccount":<String_value>,
    "httponlycookie":<String_value>,
    "homepage":<String_value>,
    "persistentcookie":<String_value>,
    "persistentcookievalidity":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmsessionpolicy

May 20, 2016

Configuration for TM session policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the session policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. Cannot be changed after a session policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression, against which traffic is evaluated. Both classic and advance expressions are supported in default partition but only advance expressions in non-default partition. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Action to be applied to connections that match this policy. Minimum length = 1
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
expressiontype	<String>	Read-only	Type of policy (Classic/Advanced) . Possible values = Classic Policy, Advanced Policy
hits	<Double>	Read-only	Number of hits.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			current policy rule evaluates to TRUE.
Name	**Data Type**	**Permissions**	**Description**
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmsessionpolicy":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmsessionpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmsessionpolicy":{
    "name":<String_value>,
    "rule":true,
    "action":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmsessionpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy?**view=summary**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmsessionpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "builtin":<String[]_value>,
      "expressiontype":<String_value>,
      "hits":<Double_value>,
      "gotopriorityexpression":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "builtin":<String[]_value>,
    "expressiontype":<String_value>,
    "hits":<Double_value>,
    "gotopriorityexpression":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "tmsessionpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmsessionpolicy_aaagroup_binding

May 20, 2016

Binding object showing the aaagroup that can be bound to tmsessionpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmsessionpolicy_aaagroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_aaagroup_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmsessionpolicy_aaagroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionpolicy_aaagroup_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"tmsessionpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmsessionpolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to tmsessionpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the session policy for which to display detailed information. Minimum length = 1
tmsessionpolicy_authenticationvserver_binding	<tmsessionpolicy_authenticationvserver_binding[]>	Read-only	authenticationvserver that can be bound to tmsessionpolicy.
tmsessionpolicy_aaagroup_binding	<tmsessionpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to tmsessionpolicy.
tmsessionpolicy_aaauser_binding	<tmsessionpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to tmsessionpolicy.
tmsessionpolicy_tmglobal_binding	<tmsessionpolicy_tmglobal_binding[]>	Read-only	tmglobal that can be bound to tmsessionpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "tmsessionpolicy_binding": [ {
    "name":<String_value>,
    "tmsessionpolicy_aaauser_binding":<tmsessionpolicy_aaauser_binding[]_value>,
    "tmsessionpolicy_tmglobal_binding":<tmsessionpolicy_tmglobal_binding[]_value>,
    "tmsessionpolicy_authenticationvserver_binding":<tmsessionpolicy_authenticationvserver_binding[]_value>,
    "tmsessionpolicy_aaagroup_binding":<tmsessionpolicy_aaagroup_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmsessionpolicy_aaauser_binding

May 20, 2016

Binding object showing the aaauser that can be bound to tmsessionpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmsessionpolicy_aaauser_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_aaauser_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmsessionpolicy_aaauser_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionpolicy_aaauser_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"tmsessionpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmsessionpolicy_authenticationvserver_binding

May 20, 2016

Binding object showing the authenticationvserver that can be bound to tmsessionpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_authenticationvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_authenticationvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmsessionpolicy_authenticationvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_authenticationvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmsessionpolicy_authenticationvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionpolicy_authenticationvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_authenticationvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"tmsessionpolicy_authenticationvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmsessionpolicy_tmglobal_binding

Jul 07, 2017

Binding object showing the tmglobal that can be bound to tmsessionpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_tmglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_tmglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmsessionpolicy_tmglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_tmglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmsessionpolicy_tmglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionpolicy_tmglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmsessionpolicy_tmglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"tmsessionpolicy_tmglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmtrafficaction

May 20, 2016

Configuration for TM traffic action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the traffic action. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after a traffic action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
apptimeout	<Double>	Read-write	Time interval, in minutes, of user inactivity after which the connection is closed. Minimum value = 1 Maximum value = 715827
sso	<String>	Read-write	Use single sign-on for the resource that the user is accessing now. Possible values = ON, OFF
formssoaction	<String>	Read-write	Name of the configured form-based single sign-on profile.
persistentcookie	<String>	Read-write	Use persistent cookies for the traffic session. A persistent cookie remains on the user device and is sent with each HTTP request. The cookie becomes stale if the session ends. Possible values = ON, OFF
initiatelogout	<String>	Read-write	Initiate logout for the traffic management (TM) session if the policy evaluates to true. The session is then terminated after two minutes. Possible values = ON, OFF
kcdaccount	<String>	Read-write	Kerberos constrained delegation account name. Default value: "None" Minimum length = 1 Maximum length = 32
samlssoprofile	<String>	Read-write	Profile to be used for doing SAML SSO to remote relying party. Minimum length = 1
forcedtimeout	<String>	Read-write	Setting to start, stop or reset TM session force timer. Possible values = START, STOP, RESET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
forcedtimeoutval	<Double>	Read-write	Time interval, in minutes, for which force timer should be set.
userexpression	<String>	Read-write	expression that will be evaluated to obtain username for SingleSignOn. Maximum length = 256
passwdexpression	<String>	Read-write	expression that will be evaluated to obtain password for SingleSignOn. Maximum length = 256
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmtrafficaction":{
    "name":<String_value>,
    "apptimeout":<Double_value>,
    "sso":<String_value>,
    "formssoaction":<String_value>,
    "persistentcookie":<String_value>,
    "initiatelogout":<String_value>,
    "kcdaccount":<String_value>,
    "samlssoprofile":<String_value>,
    "forcedtimeout":<String_value>,
    "forcedtimeoutval":<Double_value>,
    "userexpression":<String_value>,
    "passwdexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"tmtrafficaction":{
    "name":<String_value>,
    "apptimeout":<Double_value>,
    "sso":<String_value>,
    "formssoaction":<String_value>,
    "persistentcookie":<String_value>,
    "initiatelogout":<String_value>,
    "kcdaccount":<String_value>,
    "samlssoprofile":<String_value>,
    "forcedtimeout":<String_value>,
    "forcedtimeoutval":<Double_value>,
    "userexpression":<String_value>,
    "passwdexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmtrafficaction":{
    "name":<String_value>,
    "persistentcookie":true,
    "kcdaccount":true,
    "forcedtimeout":true,
    "userexpression":true,
    "passwdexpression":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmtrafficaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmtrafficaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmtrafficaction": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "apptimeout":<Double_value>,
    "sso":<String_value>,
    "formssoaction":<String_value>,
    "persistentcookie":<String_value>,
    "initiatelogout":<String_value>,
    "kcdaccount":<String_value>,
    "samlssoprofile":<String_value>,
    "forcedtimeout":<String_value>,
    "forcedtimeoutval":<Double_value>,
    "userexpression":<String_value>,
    "passwdexpression":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "tmtrafficaction": [ {
    "name":<String_value>,
    "apptimeout":<Double_value>,
    "sso":<String_value>,
    "formssoaction":<String_value>,
    "persistentcookie":<String_value>,
    "initiatelogout":<String_value>,
    "kcdaccount":<String_value>,
    "samlssoprofile":<String_value>,
    "forcedtimeout":<String_value>,
    "forcedtimeoutval":<Double_value>,
    "userexpression":<String_value>,
    "passwdexpression":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmtrafficaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmtrafficpolicy

May 20, 2016

Configuration for TM traffic policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the traffic policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression, against which traffic is evaluated. Written in the classic syntax. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the action to apply to requests or connections that match this policy. Minimum length = 1
hits	<Double>	Read-only	Number of hits.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmtrafficpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tmtrafficpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"tmtrafficpolicy":{
    "name":<String_value>,
    "rule":true,
    "action":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tmtrafficpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmtrafficpolicy resources in chunks.

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmtrafficpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "hits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmtrafficpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "hits":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmtrafficpolicy": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmtrafficpolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to tmtrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
tmtrafficpolicy_csvserver_binding	<tmtrafficpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to tmtrafficpolicy.
tmtrafficpolicy_lbvserver_binding	<tmtrafficpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to tmtrafficpolicy.
tmtrafficpolicy_tmglobal_binding	<tmtrafficpolicy_tmglobal_binding[]>	Read-only	tmglobal that can be bound to tmtrafficpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "tmtrafficpolicy_binding": [ {
    "name":<String_value>,
    "tmtrafficpolicy_csvserver_binding":<tmtrafficpolicy_csvserver_binding[]_value>,
    "tmtrafficpolicy_lbvserver_binding":<tmtrafficpolicy_lbvserver_binding[]_value>,
    "tmtrafficpolicy_tmglobal_binding":<tmtrafficpolicy_tmglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmtrafficpolicy_csvserver_binding

Jul 07, 2017

Binding object showing the csvserver that can be bound to tmtrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmtrafficpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmtrafficpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmtrafficpolicy_csvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"tmtrafficpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmtrafficpolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to tmtrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmtrafficpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmtrafficpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmtrafficpolicy_lbvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"tmtrafficpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmtrafficpolicy_tmglobal_binding

May 20, 2016

Binding object showing the tmglobal that can be bound to tmtrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_tmglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_tmglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tmtrafficpolicy_tmglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_tmglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tmtrafficpolicy_tmglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmtrafficpolicy_tmglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tmtrafficpolicy_tmglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"tmtrafficpolicy_tmglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Transform

May 20, 2016

URL Transformation configuration.

Name	Description
transformaction	Configuration for transform action
transformglobal_binding	Binding object showing the resources that can be bound to transformglobal
transformglobal_transformpolicy_binding	Binding object showing the transformpolicy that can be bound to transformglobal
transformpolicy	Configuration for URL Transformation policy
transformpolicy_binding	Binding object showing the resources that can be bound to transformpolicy
transformpolicy_csvserver_binding	Binding object showing the csvserver that can be bound to transformpolicy
transformpolicy_lbvserver_binding	Binding object showing the lbvserver that can be bound to transformpolicy
transformpolicy_transformglobal_binding	Binding object showing the transformglobal that can be bound to transformpolicy
transformpolicy_transformpolicylabel_binding	Binding object showing the transformpolicylabel that can be bound to transformpolicy
transformpolicylabel	Configuration for transform policy label
transformpolicylabel_binding	Binding object showing the resources that can be bound to transformpolicylabel
transformpolicylabel_policybinding_binding	Binding object showing the policybinding that can be bound to transformpolicylabel
transformpolicylabel_transformpolicy_binding	Binding object showing the transformpolicy that can be bound to transformpolicylabel
transformprofile	Configuration for URL Transformation profile
transformprofile_binding	Binding object showing the resources that can be bound to transformprofile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformprofile_transformaction_binding	
Name	**Description**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformaction

Jul 07, 2017

Configuration for transform action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the URL transformation action. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the URL Transformation action is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform action? or ?my transform action). Minimum length = 1
profilename	<String>	Read-write	Name of the URL Transformation profile with which to associate this action. Minimum length = 1
priority	<Double>	Read-write	Positive integer specifying the priority of the action within the profile. A lower number specifies a higher priority. Must be unique within the list of actions bound to the profile. Policies are evaluated in the order of their priority numbers, and the first policy that matches is applied. Minimum value = 1 Maximum value = 2147483647
state	<String>	Read-write	Enable or disable this action. Default value: ENABLED Possible values = ENABLED, DISABLED
requrlfrom	<String>	Read-write	PCRE-format regular expression that describes the request URL pattern to be transformed. Minimum length = 1
requrlinto	<String>	Read-write	PCRE-format regular expression that describes the transformation to be performed on URLs that match the reqUrlFrom pattern. Minimum length = 1
resurlfrom	<String>	Read-write	PCRE-format regular expression that describes the response URL pattern to be transformed. Minimum length = 1
resurlinto	<String>	Read-write	PCRE-format regular expression that describes the transformation to be performed on URLs that match the resUrlFrom pattern.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cookiedomainfrom	<String>	Read-write	Pattern that matches the domain to be transformed in Set-Cookie headers. Minimum length = 1
cookiedomaininto	<String>	Read-write	PCRE-format regular expression that describes the transformation to be performed on cookie domains that match the cookieDomainFrom pattern. NOTE: The cookie domain to be transformed is extracted from the request. Minimum length = 1
comment	<String>	Read-write	Any comments to preserve information about this URL Transformation action.
continuematching	<String>	Read-only	Continue transforming using the next rule in the list. Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/transformaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"transformaction":{
    "name":<String_value>,
    "profilename":<String_value>,
    "priority":<Double_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/transformaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/transformaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"transformaction":{
    "name":<String_value>,
    "priority":<Double_value>,
    "requrlfrom":<String_value>,
    "requrlinto":<String_value>,
    "resurlfrom":<String_value>,
    "resurlinto":<String_value>,
    "cookiedomainfrom":<String_value>,
    "cookiedomaininto":<String_value>,
    "state":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/transformaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"transformaction":{
    "name":<String_value>,
    "requrlfrom":true,
    "requrlinto":true,
    "resurlfrom":true,
    "resurlinto":true,
    "cookiedomainfrom":true,
    "cookiedomaininto":true,
    "state":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/transformaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/transformaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/transformaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of transformaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/transformaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformaction": [ {
      "name":<String_value>,
      "profilename":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "requrlfrom":<String_value>,
    "requrlinto":<String_value>,
    "resurlfrom":<String_value>,
    "resurlinto":<String_value>,
    "cookiedomainfrom":<String_value>,
    "cookiedomaininto":<String_value>,
    "continuematching":<String_value>,
    "state":<String_value>,
    "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/transformaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/transformaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformaction": [ {
    "name":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "profilename":<String_value>,
    "priority":<Double_value>,
    "requrlfrom":<String_value>,
    "requrlinto":<String_value>,
    "resurlfrom":<String_value>,
    "resurlinto":<String_value>,
    "cookiedomainfrom":<String_value>,
    "cookiedomaininto":<String_value>,
    "continuematching":<String_value>,
    "state":<String_value>,
    "comment":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformglobal_binding

May 20, 2016

Binding object which returns the resources bound to transformglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
transformglobal_transformpolicy_binding	<transformglobal_transformpolicy_binding[]>	Read-only	transformpolicy that can be bound to transformglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/transformglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "transformglobal_binding": [ {
      "transformglobal_transformpolicy_binding":<transformglobal_transformpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformglobal_transformpolicy_binding

Jul 07, 2017

Binding object showing the transformpolicy that can be bound to transformglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
globalbindtype	<String>	Read-write	. Default value: SYSTEM_GLOBAL Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
policyname	<String>	Read-write	Name of the transform policy.
labelname	<String>	Read-write	Name of the policy label to invoke if the current policy evaluates to TRUE, the invoke parameter is set, and the label type is Policy Label.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forwards the request or response to the specified virtual server or evaluates the specified policy label.
type	<String>	Read-write	Specifies the bind point to which to bind the policy. Available settings function as follows: * REQ_OVERRIDE. Request override. Binds the policy to the priority request queue. * REQ_DEFAULT. Binds the policy to the default request queue. Possible values = REQ_OVERRIDE, REQ_DEFAULT
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Send the request to the specified request virtual server. * resvserver - Send the response to the specified response virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, resvserver, policylabel
numpol	<Double>	Read-only	The number of policies bound to the bindpoint.
flowtype	<Double>	Read-only	flowtype of the bound transform policy.
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/transformglobal_transformpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"transformglobal_transformpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "type":<String_value>,
    "invoke":<Boolean_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "labeltype":<String_value>,
    "labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/transformglobal_transformpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformglobal_transformpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/transformglobal_transformpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformglobal_transformpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformglobal_transformpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformglobal_transformpolicy_binding resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformglobal_transformpolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "policyname":<String_value>,
    "labelname":<String_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "type":<String_value>,
    "labeltype":<String_value>,
    "numpol":<Double_value>,
    "flowtype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformglobal_transformpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "transformglobal_transformpolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicy

May 20, 2016

Configuration for URL Transformation policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the URL Transformation policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the URL Transformation policy is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform policy? or ?my transform policy). Minimum length = 1
rule	<String>	Read-write	Expression, or name of a named expression, against which to evaluate traffic. Can be written in either default or classic syntax. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes blank spaces, the entire expression must be enclosed in double quotation marks. * If the expression itself includes double quotation marks, you must escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
profilename	<String>	Read-write	Name of the URL Transformation profile to use to transform requests and responses that match the policy. Minimum length = 1
comment	<String>	Read-write	Any comments to preserve information about this URL Transformation policy.
logaction	<String>	Read-write	Log server to use to log connections that match this policy.
			New name for the policy. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform policy? or ?my transform policy). Minimum length = 1
hits	<Double>	Read-only	Number of hits.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default transform policy.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"transformpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "profilename":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"transformpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "profilename":<String_value>,
    "comment":<String_value>,
    "logaction":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"transformpolicy":{
    "name":<String_value>,
    "comment":true,
    "logaction":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"transformpolicy":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of transformpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "profilename":<String_value>,
      "hits":<Double_value>,
      "comment":<String_value>,
      "logaction":<String_value>,
      "isdefault":<Boolean_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "profilename":<String_value>,
      "hits":<Double_value>,
      "comment":<String_value>,
      "logaction":<String_value>,
      "isdefault":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicy": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicylabel

Jul 07, 2017

Configuration for transform policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name for the policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the URL Transformation policy label is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform policylabel? or ?my transform policylabel).
policylabeltype	<String>	Read-write	Types of transformations allowed by the policies bound to the label. For URL transformation, always http_req (HTTP Request). Possible values = http_req
newname	<String>	Read-write	New name for the policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform policylabel? or ?my transform policylabel). Minimum length = 1
numpol	<Double>	Read-only	Number of polices bound to label.
hits	<Double>	Read-only	Number of times policy label was invoked.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
			Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Possible values = reqvserver, policylabel
Name	**Data Type**	**Permissions**	**Description**
invoke_labelname	<String>	Read-only	Name of the policy label.
description	<String>	Read-only	Description of the policylabel.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | RENAME | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"transformpolicylabel":{
    "labelname":<String_value>,
    "policylabeltype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"transformpolicylabel":{
    "labelname":<String_value>,
    "newname":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of transformpolicylabel resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformpolicylabel resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicylabel": [ {
      "labelname":<String_value>,
      "policylabeltype":<String_value>,
      "numpol":<Double_value>,
      "hits":<Double_value>,
      "priority":<Double_value>,
      "gotopriorityexpression":<String_value>,
      "labeltype":<String_value>,
      "invoke_labelname":<String_value>,
      "description":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel/labelname_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel/labelname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel/labelname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicylabel": [ {
    "labelname":<String_value>,
    "policylabeltype":<String_value>,
    "numpol":<Double_value>,
    "hits":<Double_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>,
    "description":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicylabel": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicylabel_binding

May 20, 2016

Binding object which returns the resources bound to transformpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name for the policy label. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the URL Transformation policy label is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform policylabel? or ?my transform policylabel).
transformpolicylabel_policybinding_binding	<transformpolicylabel_policybinding_binding[]>	Read-only	policybinding that can be bound to transformpolicylabel.
transformpolicylabel_transformpolicy_binding	<transformpolicylabel_transformpolicy_binding[]>	Read-only	transformpolicy that can be bound to transformpolicylabel.

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel_binding/labelname_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "transformpolicylabel_binding": [ {
    "labelname":<String_value>,
    "transformpolicylabel_policybinding_binding":<transformpolicylabel_policybinding_binding[]_value>,
    "transformpolicylabel_transformpolicy_binding":<transformpolicylabel_transformpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicylabel_policybinding_binding

Jul 07, 2017

Binding object showing the policybinding that can be bound to transformpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the URL Transformation policy to bind to the policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, policylabel
labelname	<String>	Read-write	Name of the URL Transformation policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	Name of the policy label.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that

© 1999-2017 Citrix Systems, Inc. All rights reserved.

session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel_policybinding_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel_policybinding_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicylabel_policybinding_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel_policybinding_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformpolicylabel_policybinding_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "transformpolicylabel_policybinding_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel_policybinding_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"transformpolicylabel_policybinding_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicylabel_transformpolicy_binding

May 20, 2016

Binding object showing the transformpolicy that can be bound to transformpolicylabel.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-write	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
policyname	<String>	Read-write	Name of the URL Transformation policy to bind to the policy label.
labeltype	<String>	Read-write	Type of invocation. Available settings function as follows: * reqvserver - Forward the request to the specified request virtual server. * policylabel - Invoke the specified policy label. Possible values = reqvserver, policylabel
labelname	<String>	Read-write	Name of the URL Transformation policy label to which to bind the policy.
invoke_labelname	<String>	Read-write	Name of the policy label.
invoke	<Boolean>	Read-write	If the current policy evaluates to TRUE, terminate evaluation of policies bound to the current policy label, and then forward the request to the specified virtual server or evaluate the specified policy label.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/transformpolicylabel_transformpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"transformpolicylabel_transformpolicy_binding":{
    "labelname":<String_value>,
    "policyname":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "invoke":<Boolean_value>,
    "labeltype":<String_value>,
    "invoke_labelname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicylabel_transformpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformpolicylabel_transformpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicylabel_transformpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policyname":<String_value>,
    "labeltype":<String_value>,
    "labelname":<String_value>,
    "invoke_labelname":<String_value>,
    "invoke":<Boolean_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicylabel_transformpolicy_binding/labelname_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"transformpolicylabel_transformpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicy_binding

Jul 07, 2017

Binding object which returns the resources bound to transformpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the URL Transformation policy. Minimum length = 1
transformpolicy_lbvserver_binding	<transformpolicy_lbvserver_binding[]>	Read-only	lbvserver that can be bound to transformpolicy.
transformpolicy_transformglobal_binding	<transformpolicy_transformglobal_binding[]>	Read-only	transformglobal that can be bound to transformpolicy.
transformpolicy_csvserver_binding	<transformpolicy_csvserver_binding[]>	Read-only	csvserver that can be bound to transformpolicy.
transformpolicy_transformpolicylabel_binding	<transformpolicy_transformpolicylabel_binding[]>	Read-only	transformpolicylabel that can be bound to transformpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "transformpolicy_binding": [ {
      "name":<String_value>,
      "transformpolicy_transformglobal_binding":<transformpolicy_transformglobal_binding[]_value>,
      "transformpolicy_lbvserver_binding":<transformpolicy_lbvserver_binding[]_value>,
      "transformpolicy_csvserver_binding":<transformpolicy_csvserver_binding[]_value>,
      "transformpolicy_transformpolicylabel_binding":<transformpolicy_transformpolicylabel_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicy_csvserver_binding

May 20, 2016

Binding object showing the csvserver that can be bound to transformpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the URL Transformation policy. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_csvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_csvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicy_csvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_csvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformpolicy_csvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicy_csvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_csvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"transformpolicy_csvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicy_lbvserver_binding

May 20, 2016

Binding object showing the lbvserver that can be bound to transformpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the URL Transformation policy. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_lbvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_lbvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicy_lbvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_lbvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformpolicy_lbvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicy_lbvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_lbvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"transformpolicy_lbvserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformprofile

May 20, 2016

Configuration for URL Transformation profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the URL transformation profile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore characters. Cannot be changed after the URL transformation profile is added. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, ?my transform profile? or ?my transform profile?). Minimum length = 1
type	<String>	Read-write	Type of transformation. Always URL for URL Transformation profiles. Possible values = URL
onlytransformabsurlinbody	<String>	Read-write	In the HTTP body, transform only absolute URLs. Relative URLs are ignored. Possible values = ON, OFF
comment	<String>	Read-write	Any comments to preserve information about this URL Transformation profile.
regexforfindingurlinjavascript	<String>	Read-only	Patclass having regexes to find the URLs in JavaScript.
regexforfindingurlincss	<String>	Read-only	Patclass having regexes to find the URLs in CSS.
regexforfindingurlinxcomponent	<String>	Read-only	Patclass having regexes to find the URLs in X-Component.
regexforfindingurlinxml	<String>	Read-only	Patclass having regexes to find the URLs in XML.
additionalreqheaderslist	<String>	Read-only	Patclass having a list of additional request header names that should transformed.
additionalrespheaderslist	<String>	Read-only	Patclass having a list of additional response header names that should transformed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/transformprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"transformprofile":{
    "name":<String_value>,
    "type":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/transformprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/transformprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"transformprofile":{
    "name":<String_value>,
    "type":<String_value>,
    "onlytransformabsurlinbody":<String_value>,
    "comment":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/transformprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"transformprofile":{
    "name":<String_value>,
    "type":true,
    "onlytransformabsurlinbody":true,
    "comment":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/transformprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/transformprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/transformprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of transformprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/transformprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformprofile": [ {
      "name":<String_value>,
      "type":<String_value>,
      "regexforfindingurlinjavascript":<String_value>,
      "regexforfindingurlincss":<String_value>,
      "regexforfindingurlinxcomponent":<String_value>,
      "regexforfindingurlinxml":<String_value>,
      "additionalreqheaderslist":<String_value>,
      "additionalrespheaderslist":<String_value>,
      "onlytransformabsurlinbody":<String_value>,
      "comment":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/transformprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/transformprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformprofile": [ {
    "name":<String_value>,
    "type":<String_value>,
    "regexforfindingurlinjavascript":<String_value>,
    "regexforfindingurlincss":<String_value>,
    "regexforfindingurlinxcomponent":<String_value>,
    "regexforfindingurlinxml":<String_value>,
    "additionalreqheaderslist":<String_value>,
    "additionalrespheaderslist":<String_value>,
    "onlytransformabsurlinbody":<String_value>,
    "comment":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "transformprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicy_transformglobal_binding

Jul 07, 2017

Binding object showing the transformglobal that can be bound to transformpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the URL Transformation policy. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotoprioritiyexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then

© 1999-2017 Citrix Systems, Inc. All rights reserved.

using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_transformglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_transformglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicy_transformglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_transformglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformpolicy_transformglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicy_transformglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_transformglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"transformpolicy_transformglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicy_transformpolicylabel_binding

May 20, 2016

Binding object showing the transformpolicylabel that can be bound to transformpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the URL Transformation policy. Minimum length = 1
boundto	<String>	Read-write	Location where policy is bound.
priority	<Double>	Read-only	Specifies the priority of the policy.
gotopriorityexpression	<String>	Read-only	Expression specifying the priority of the next policy which will get evaluated if the current policy rule evaluates to TRUE.
labeltype	<String>	Read-only	Type of policy label invocation. Possible values = reqvserver, policylabel
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
labelname	<String>	Read-only	Name of the label to invoke if the current policy rule evaluates to TRUE.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_transformpolicylabel_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_transformpolicylabel_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformpolicy_transformpolicylabel_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_transformpolicylabel_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformpolicy_transformpolicylabel_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "transformpolicy_transformpolicylabel_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
    "gotoprioritypression":<String_value>,
    "labeltype":<String_value>,
    "activepolicy":<Integer_value>,
    "labelname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformpolicy_transformpolicylabel_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"transformpolicy_transformpolicylabel_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformprofile_binding

Jul 07, 2017

Binding object which returns the resources bound to transformprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the profile. Minimum length = 1
transformprofile_transformaction_binding	<transformprofile_transformaction_binding[]>	Read-only	transformaction that can be bound to transformprofile.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformprofile_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "transformprofile_binding": [ {
      "name":<String_value>,
      "transformprofile_transformaction_binding":<transformprofile_transformaction_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformprofile_transformaction_binding

May 20, 2016

Binding object showing the transformaction that can be bound to transformprofile.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the profile. Minimum length = 1
actionname	<String>	Read-write	URL Transformation action name.
priority	<Double>	Read-only	Priority of the Action within the Profile.
resurlfrom	<String>	Read-only	Pattern of original response URLs. It corresponds to the way external users view the server, and acts as a source for response transformations.
profilename	<String>	Read-only	URL Transformation profile name.
state	<String>	Read-only	Enabled flag. Possible values = ENABLED, DISABLED
actioncomment	<String>	Read-only	Comments.
requrlinto	<String>	Read-only	Pattern of transformed request URLs. It corresponds to internal addresses and indicates how they are created.
cookiedomainfrom	<String>	Read-only	Pattern of the original domain in Set-Cookie headers.
cookiedomaininto	<String>	Read-only	Pattern of the transformed domain in Set-Cookie headers.
resurlinto	<String>	Read-only	Pattern of transformed response URLs. It corresponds to internal addresses and indicates how they are created.
requrlfrom	<String>	Read-only	Pattern of original request URLs. It corresponds to the way external users view the server, and acts as a source for request transformations.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/transformprofile_transformaction_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/transformprofile_transformaction_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of transformprofile_transformaction_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/transformprofile_transformaction_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the transformprofile_transformaction_binding resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformprofile_transformaction_binding": [ {
    "name":<String_value>,
    "actionname":<String_value>,
    "priority":<Double_value>,
    "resurlfrom":<String_value>,
    "profilename":<String_value>,
    "state":<String_value>,
    "actioncomment":<String_value>,
    "requrlinto":<String_value>,
    "cookiedomainfrom":<String_value>,
    "cookiedomaininto":<String_value>,
    "resurlinto":<String_value>,
    "requrlfrom":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/transformprofile_transformaction_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{"transformprofile_transformaction_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tunnel

Jul 07, 2017

SSL VPN Tunnel Configuration.

Name	Description
tunnelglobal_binding	Binding object showing the resources that can be bound to tunnelglobal
tunnelglobal_tunneltrafficpolicy_binding	Binding object showing the tunneltrafficpolicy that can be bound to tunnelglobal
tunneltrafficpolicy	Configuration for tunnel policy
tunneltrafficpolicy_binding	Binding object showing the resources that can be bound to tunneltrafficpolicy
tunneltrafficpolicy_tunnelglobal_binding	Binding object showing the tunnelglobal that can be bound to tunneltrafficpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tunnelglobal_binding

May 20, 2016

Binding object which returns the resources bound to tunnelglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
tunnelglobal_tunneltrafficpolicy_binding	<tunnelglobal_tunneltrafficpolicy_binding[]>	Read-only	tunneltrafficpolicy that can be bound to tunnelglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/tunnelglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "tunnelglobal_binding": [ {
    "tunnelglobal_tunneltrafficpolicy_binding":<tunnelglobal_tunneltrafficpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tunnelglobal_tunneltrafficpolicy_binding

May 20, 2016

Binding object showing the tunneltrafficpolicy that can be bound to tunnelglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	Priority.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	Policy name.
state	<String>	Read-write	Current state of the binding. If the binding is enabled, the policy is active. Possible values = ENABLED, DISABLED
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"tunnelglobal_tunneltrafficpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "state":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tunnelglobal_tunneltrafficpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tunnelglobal_tunneltrafficpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tunnelglobal_tunneltrafficpolicy_binding": [ {
    "priority":<Double_value>,
    "builtin":<String[]_value>,
    "policyname":<String_value>,
    "state":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tunnelglobal_tunneltrafficpolicy_binding?**count=yes**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "tunnelglobal_tunneltrafficpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tunneltrafficpolicy

May 20, 2016

Configuration for tunnel policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the tunnel traffic policy. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy)'. Minimum length = 1
rule	<String>	Read-write	Expression, against which traffic is evaluated. Written in classic or default syntax. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes blank spaces, the entire expression must be enclosed in double quotation marks. * If the expression itself includes double quotation marks, you must escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Name of the built-in compression action to associate with the policy. Minimum length = 1
hits	<Double>	Read-only	No of hits.
txbytes	<Double>	Read-only	Number of bytes transmitted.
rxbytes	<Double>	Read-only	Number of bytes received.
clientttlb	<Double>	Read-only	Total client TTLB value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
servertttlb	<Double>	Read-only	Total SERVERTTLB value.
servertransactions	<Double>	Read-only	Number of server transactions.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default tunnelpolicy.
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tunneltrafficpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{"tunneltrafficpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"tunneltrafficpolicy":{
    "name":<String_value>,
    "rule":true,
    "action":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of tunneltrafficpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tunneltrafficpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tunneltrafficpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "hits":<Double_value>,
      "txbytes":<Double_value>,
      "rxbytes":<Double_value>,
      "clientttlb":<Double_value>,
      "clienttransactions":<Double_value>,
      "serverttlb":<Double_value>,
      "servertransactions":<Double_value>,
      "isdefault":<Boolean_value>,
      "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tunneltrafficpolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>,
    "hits":<Double_value>,
    "txbytes":<Double_value>,
    "rxbytes":<Double_value>,
    "clientttlb":<Double_value>,
    "clienttransactions":<Double_value>,
    "serverttlb":<Double_value>,
    "servertransactions":<Double_value>,
    "isdefault":<Boolean_value>,
    "builtin":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tunneltrafficpolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tunneltrafficpolicy_binding

May 20, 2016

Binding object which returns the resources bound to tunneltrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the tunnel traffic policy for which to show detailed information. Minimum length = 1
tunneltrafficpolicy_tunnelglobal_binding	<tunneltrafficpolicy_tunnelglobal_binding[]>	Read-only	tunnelglobal that can be bound to tunneltrafficpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "tunneltrafficpolicy_binding": [ {
      "name":<String_value>,
      "tunneltrafficpolicy_tunnelglobal_binding":<tunneltrafficpolicy_tunnelglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tunneltrafficpolicy_tunnelglobal_binding

May 20, 2016

Binding object showing the tunnelglobal that can be bound to tunneltrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the tunnel traffic policy for which to show detailed information. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy_tunnelglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy_tunnelglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of tunneltrafficpolicy_tunnelglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy_tunnelglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the tunneltrafficpolicy_tunnelglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tunneltrafficpolicy_tunnelglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Double_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/tunneltrafficpolicy_tunnelglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"tunneltrafficpolicy_tunnelglobal_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Utility

May 20, 2016

Utilities.

Name	Description
callhome	Configuration for callhome
techsupport	Configuration for tech support

© 1999-2017 Citrix Systems, Inc. All rights reserved.

callhome

Jul 07, 2017

Configuration for callhome resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
emailaddress	<String>	Read-write	The contact person's E-mail address. Maximum length = 78
proxymode	<String>	Read-write	Deploy the callhome proxy mode. Default value: NO Possible values = YES, NO
ipaddress	<String>	Read-write	Proxy Server IP address. Minimum length = 1
port	<Integer>	Read-write	Proxy Server Port. Minimum value = 1 Range 1 - 65535
sslcardfirstfailure	<String>	Read-only	First occurrence SSL card failure.
sslcardlatestfailure	<String>	Read-only	Latest occurrence SSL card failure.
powfirstfail	<String>	Read-only	First occurrence power supply unit failure.
powlatestfailure	<String>	Read-only	Latest occurrence power supply unit failure.
hddfirstfail	<String>	Read-only	First occurrence hard disk drive failure.
hddlatestfailure	<String>	Read-only	Latest occurrence hard disk drive failure.
flashfirstfail	<String>	Read-only	First occurrence compact flash failure.
flashlatestfailure	<String>	Read-only	Latest occurrence compact flush failure.
restartlatestfail	<String>	Read-only	Latest occurrence warm restart failure.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = ENABLED, DISABLED, SUCCESSFUL, FAILED

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/callhome

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"callhome":{
    "emailaddress":<String_value>,
    "proxymode":<String_value>,
    "ipaddress":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "port":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/callhome?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"callhome":{
    "emailaddress":true,
    "proxymode":true,
    "ipaddress":true,
    "port":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/callhome

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "callhome": [ {
    "emailaddress":<String_value>,
    "proxymode":<String_value>,
    "ipaddress":<String_value>,
    "port":<Integer_value>,
    "sslcardfirstfailure":<String_value>,
    "sslcardlatestfailure":<String_value>,
    "powfirstfail":<String_value>,
    "powlatestfailure":<String_value>,
    "hddfirstfail":<String_value>,
    "hddlatestfailure":<String_value>,
    "flashfirstfail":<String_value>,
    "flashlatestfailure":<String_value>,
    "restartlatestfail":<String_value>,
    "callhomestatus":<String[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

techsupport

May 20, 2016

Configuration for tech support resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
scope	<String>	Read-write	Use this option to gather data on the present node, all cluster nodes, or for the specified partitions. The CLUSTER scope generates smaller abbreviated archives for all nodes. The PARTITION scope collects the admin partition in addition to those specified. The partitionName option is only required for the PARTITION scope. Default value: NODE Possible values = NODE, CLUSTER, PARTITION
partitionname	<String>	Read-write	Name of the partition. Minimum length = 1
upload	<Boolean>	Read-write	Securely upload the collector archive to Citrix Technical Support using SSL. MyCitrix credentials will be required. If used with the -file option, no new collector archive is generated. Instead, the specified archive is uploaded. Note that the upload operation time depends on the size of the archive file, and the connection bandwidth.
proxy	<String>	Read-write	Specifies the proxy server to be used when uploading a collector archive. Use this parameter if the NetScaler appliance does not have direct internet connectivity. The basic format of the proxy string is: "proxy_IP:<proxy_port>" (without quotes). If the proxy requires authentication the format is: "username:password@proxy_IP:<proxy_port>".
casenumber	<String>	Read-write	Specifies the associated case or service request number if it has already been opened with Citrix Technical Support.
file	<String>	Read-write	Specifies the name (with full path) of the collector archive file to be uploaded. If this is specified, no new collector archive is generated.
description	<String>	Read-write	Provides a text description for the the upload, and can be used for logging purposes.
username	<String>	Read-write	Specifies My Citrix user name, which is used to login to Citrix upload server.
password	<String>	Read-write	Specifies My Citrix password, which is used to login to Citrix upload server.
response	<String>	Read-only	Output as text printed to console and syslog at NOTICE level.
servername	<String>	Read-only	.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/techsupport

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/techsupport?**args=scope:<String_value>,partitionname:<String_value>,upload:<Boolean_value>,proxy:<String_value>,casenumber:<String_value>,file:<Stri**

Use this query-parameter to get techsupport resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "techsupport": [ {
scope:<String_value>,partitionname:<String_value>,upload:<Boolean_value>,proxy:<String_value>,casenumber:<String_value>,file:<String_value>,description:<String_value>,username:<String_value>,password:<String_value>      "respo
      "servername":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL VPN

May 20, 2016

Virtual Private Network configuration.

Name	Description
vpnclientlessaccesspolicy	Configuration for Clientless VPN rewrite policy
vpnclientlessaccesspolicy_binding	Binding object showing the resources that can be bound to vpnclientlessaccesspolicy
vpnclientlessaccesspolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to vpnclientlessaccesspolicy
vpnclientlessaccesspolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to vpnclientlessaccesspolicy
vpnclientlessaccessprofile	Configuration for Clientless VPN rewrite profile
vpnepaprofile	Configuration for Epa profile
vpneula	Configuration for EULA for vservers
vpnformssoaction	Configuration for Form sso action
vpnglobal_appcontroller_binding	Binding object showing the appcontroller that can be bound to vpnglobal
vpnglobal_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to vpnglobal
vpnglobal_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to vpnglobal
vpnglobal_authenticationcertpolicy_binding	Binding object showing the authenticationcertpolicy that can be bound to vpnglobal
vpnglobal_authenticationldappolicy_binding	Binding object showing the authenticationldappolicy that can be bound to vpnglobal
vpnglobal_authenticationlocalpolicy_binding	Binding object showing the authenticationlocalpolicy that can be bound to vpnglobal
vpnglobal_authenticationnegotiatepolicy_binding	Binding object showing the authenticationnegotiatepolicy that can be bound to vpnglobal

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_authenticationpolicy_binding	
Name	**Description**
vpnglobal_authenticationradiuspolicy_binding	Binding object showing the authenticationradiuspolicy that can be bound to vpnglobal
vpnglobal_authenticationsamlpolicy_binding	Binding object showing the authenticationsamlpolicy that can be bound to vpnglobal
vpnglobal_authenticationtacacspolicy_binding	Binding object showing the authenticationtacacspolicy that can be bound to vpnglobal
vpnglobal_binding	Binding object showing the resources that can be bound to vpnglobal
vpnglobal_domain_binding	Binding object showing the domain that can be bound to vpnglobal
vpnglobal_intranetip6_binding	Binding object showing the intranetip6 that can be bound to vpnglobal
vpnglobal_intranetip_binding	Binding object showing the intranetip that can be bound to vpnglobal
vpnglobal_sharefileserver_binding	Binding object showing the sharefileserver that can be bound to vpnglobal
vpnglobal_staserver_binding	Binding object showing the staserver that can be bound to vpnglobal
vpnglobal_vpnclientlessaccesspolicy_binding	Binding object showing the vpnclientlessaccesspolicy that can be bound to vpnglobal
vpnglobal_vpneula_binding	Binding object showing the vpneula that can be bound to vpnglobal
vpnglobal_vpnintranetapplication_binding	Binding object showing the vpnintranetapplication that can be bound to vpnglobal
vpnglobal_vpnnexthopserver_binding	Binding object showing the vpnnexthopserver that can be bound to vpnglobal
vpnglobal_vpnportaltheme_binding	Binding object showing the vpnportaltheme that can be bound to vpnglobal
vpnglobal_vpnsessionpolicy_binding	Binding object showing the vpnsessionpolicy that can be bound to vpnglobal
vpnglobal_vpntrafficpolicy_binding	Binding object showing the vpntrafficpolicy that can be bound to vpnglobal
vpnglobal_vpnurl_binding	Binding object showing the vpnurl that can be bound to vpnglobal
vpnicaconnection	Configuration for active ica connections

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
vpnicadtlsconnection	Configuration for active ica connections
vpnintranetapplication	Configuration for SSLVPN intranet application
vpnnexthopserver	Configuration for Next Hop Server
vpnparameter	Configuration for VPN parameter
vpnportaltheme	Configuration for portaltheme
vpnsamlssoprofile	Configuration for SAML sso action
vpnsessionaction	Configuration for VPN session action
vpnsessionpolicy	Configuration for VPN session policy
vpnsessionpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to vpnsessionpolicy
vpnsessionpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to vpnsessionpolicy
vpnsessionpolicy_binding	Binding object showing the resources that can be bound to vpnsessionpolicy
vpnsessionpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to vpnsessionpolicy
vpnsessionpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to vpnsessionpolicy
vpntrafficaction	Configuration for VPN traffic action
vpntrafficpolicy	Configuration for VPN traffic policy
vpntrafficpolicy_aaagroup_binding	Binding object showing the aaagroup that can be bound to vpntrafficpolicy
vpntrafficpolicy_aaauser_binding	Binding object showing the aaauser that can be bound to vpntrafficpolicy
vpntrafficpolicy_binding	Binding object showing the resources that can be bound to vpntrafficpolicy
vpntrafficpolicy_vpnglobal_binding	Binding object showing the vpnglobal that can be bound to vpntrafficpolicy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
vpntrafficpolicy_vpnvserver_binding	Binding object showing the vpnvserver that can be bound to vpntrafficpolicy
vpnurl	Configuration for VPN URL
vpnvserver	Configuration for VPN virtual server
vpnvserver_aaapreauthenticationpolicy_binding	Binding object showing the aaapreauthenticationpolicy that can be bound to vpnvserver
vpnvserver_appcontroller_binding	Binding object showing the appcontroller that can be bound to vpnvserver
vpnvserver_appflowpolicy_binding	Binding object showing the appflowpolicy that can be bound to vpnvserver
vpnvserver_auditnslogpolicy_binding	Binding object showing the auditnslogpolicy that can be bound to vpnvserver
vpnvserver_auditsyslogpolicy_binding	Binding object showing the auditsyslogpolicy that can be bound to vpnvserver
vpnvserver_authenticationcertpolicy_binding	Binding object showing the authenticationcertpolicy that can be bound to vpnvserver
vpnvserver_authenticationdfapolicy_binding	Binding object showing the authenticationdfapolicy that can be bound to vpnvserver
vpnvserver_authenticationldappolicy_binding	Binding object showing the authenticationldappolicy that can be bound to vpnvserver
vpnvserver_authenticationlocalpolicy_binding	Binding object showing the authenticationlocalpolicy that can be bound to vpnvserver
vpnvserver_authenticationloginschemapolicy_binding	Binding object showing the authenticationloginschemapolicy that can be bound to vpnvserver
vpnvserver_authenticationnegotiatepolicy_binding	Binding object showing the authenticationnegotiatepolicy that can be bound to vpnvserver
vpnvserver_authenticationpolicy_binding	Binding object showing the authenticationpolicy that can be bound to vpnvserver
vpnvserver_authenticationradiuspolicy_binding	Binding object showing the authenticationradiuspolicy that can be bound to vpnvserver
vpnvserver_authenticationsamlidppolicy_binding	Binding object showing the authenticationsamlidppolicy that can be bound to vpnvserver
vpnvserver_authenticationsamlpolicy_binding	Binding object showing the authenticationsamlpolicy that can be bound to vpnvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationtacacspolicy_binding	vpnvserver
Name	**Description**
vpnvserver_authenticationwebauthpolicy_binding	Binding object showing the authenticationwebauthpolicy that can be bound to vpnvserver
vpnvserver_binding	Binding object showing the resources that can be bound to vpnvserver
vpnvserver_cachepolicy_binding	Binding object showing the cachepolicy that can be bound to vpnvserver
vpnvserver_cspolicy_binding	Binding object showing the cspolicy that can be bound to vpnvserver
vpnvserver_feopolicy_binding	Binding object showing the feopolicy that can be bound to vpnvserver
vpnvserver_icapolicy_binding	Binding object showing the icapolicy that can be bound to vpnvserver
vpnvserver_intranetip6_binding	Binding object showing the intranetip6 that can be bound to vpnvserver
vpnvserver_intranetip_binding	Binding object showing the intranetip that can be bound to vpnvserver
vpnvserver_responderpolicy_binding	Binding object showing the responderpolicy that can be bound to vpnvserver
vpnvserver_rewritepolicy_binding	Binding object showing the rewritepolicy that can be bound to vpnvserver
vpnvserver_sharefileserver_binding	Binding object showing the sharefileserver that can be bound to vpnvserver
vpnvserver_staserver_binding	Binding object showing the staserver that can be bound to vpnvserver
vpnvserver_vpnclientlessaccesspolicy_binding	Binding object showing the vpnclientlessaccesspolicy that can be bound to vpnvserver
vpnvserver_vpnepaprofile_binding	Binding object showing the vpnepaprofile that can be bound to vpnvserver
vpnvserver_vpneula_binding	Binding object showing the vpneula that can be bound to vpnvserver
vpnvserver_vpnintranetapplication_binding	Binding object showing the vpnintranetapplication that can be bound to vpnvserver
vpnvserver_vpnnexthopserver_binding	Binding object showing the vpnnexthopserver that can be bound to vpnvserver
vpnvserver_vpnportaltheme_binding	Binding object showing the vpnportaltheme that can be bound to vpnvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
vpnvserver_vpnsessionpolicy_binding	Binding object showing the vpnsessionpolicy that can be bound to vpnvserver
vpnvserver_vpntrafficpolicy_binding	Binding object showing the vpntrafficpolicy that can be bound to vpnvserver
vpnvserver_vpnurl_binding	Binding object showing the vpnurl that can be bound to vpnvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnclientlessaccesspolicy

May 20, 2016

Configuration for Clientless VPN rewrite policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the new clientless access policy. Minimum length = 1
rule	<String>	Read-write	Expression, or name of a named expression, specifying the traffic that matches the policy. Can be written in either default or classic syntax. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
profilename	<String>	Read-write	Name of the profile to invoke for the clientless access.
undefaction	<String>	Read-only	The UNDEF action.
hits	<Double>	Read-only	The number of times the policy evaluated to true.
undefhits	<Double>	Read-only	The number of times the policy evaluation resulted in undefined processing.
description	<String>	Read-only	Description of the clientless access policy.
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default vpnclientlessrwpolicy.
builtin	<String[]>	Read-only	Flag to determine if vpn clientless rewrite policy is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description

Operations (click to see Properties)

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnclientlessaccesspolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "profilename":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnclientlessaccesspolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "profilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnclientlessaccesspolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnclientlessaccesspolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnclientlessaccesspolicy": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "rule":<String_value>,
    "profilename":<String_value>,
    "undefaction":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "description":<String_value>,
    "isdefault":<Boolean_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnclientlessaccesspolicy": [ {
    "name":<String_value>,
    "rule":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "profilename":<String_value>,
    "undefaction":<String_value>,
    "hits":<Double_value>,
    "undefhits":<Double_value>,
    "description":<String_value>,
    "isdefault":<Boolean_value>,
    "builtin":<String[]_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnclientlessaccesspolicy": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnclientlessaccesspolicy_binding

May 20, 2016

Binding object which returns the resources bound to vpnclientlessaccesspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the clientless access policy to display. Minimum length = 1
vpnclientlessaccesspolicy_vpnglobal_binding	<vpnclientlessaccesspolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to vpnclientlessaccesspolicy.
vpnclientlessaccesspolicy_vpnvserver_binding	<vpnclientlessaccesspolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to vpnclientlessaccesspolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "vpnclientlessaccesspolicy_binding": [ {
    "name":<String_value>,
    "vpnclientlessaccesspolicy_vpnglobal_binding":<vpnclientlessaccesspolicy_vpnglobal_binding[]_value>,
    "vpnclientlessaccesspolicy_vpnvserver_binding":<vpnclientlessaccesspolicy_vpnvserver_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnclientlessaccesspolicy_vpnglobal_binding

May 20, 2016

Binding object showing the vpnglobal that can be bound to vpnclientlessaccesspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the clientless access policy to display. Minimum length = 1
priority	<Double>	Read-only	Specifies the priority of the policy. Minimum value = 1 Maximum value = 2147483647
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnclientlessaccesspolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnclientlessaccesspolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnclientlessaccesspolicy_vpnglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Integer_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnclientlessaccesspolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnclientlessaccesspolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to vpnclientlessaccesspolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	Location where policy is bound.
name	<String>	Read-write	Name of the clientless access policy to display. Minimum length = 1
priority	<Double>	Read-only	Specifies the priority of the policy. Minimum value = 1 Maximum value = 2147483647
activepolicy	<Integer>	Read-only	Indicates whether policy is bound or not.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnclientlessaccesspolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnclientlessaccesspolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnclientlessaccesspolicy_vpnvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
    "activepolicy":<Integer_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccesspolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnclientlessaccesspolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnclientlessaccessprofile

Jul 07, 2017

Configuration for Clientless VPN rewrite profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
profilename	<String>	Read-write	Name for the NetScaler Gateway clientless access profile. Must begin with an ASCII alphabetic or underscore (_) character, and must consist only of ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my profile" or 'my profile'). Minimum length = 1
urlrewritepolicylabel	<String>	Read-write	Name of the configured URL rewrite policy label. If you do not specify a policy label name, then URLs are not rewritten. Minimum length = 1
javascriptrewritepolicylabel	<String>	Read-write	Name of the configured JavaScript rewrite policy label. If you do not specify a policy label name, then JAVA scripts are not rewritten. Minimum length = 1
reqhdrrewritepolicylabel	<String>	Read-write	Name of the configured Request rewrite policy label. If you do not specify a policy label name, then requests are not rewritten. Minimum length = 1
reshdrrewritepolicylabel	<String>	Read-write	Name of the configured Response rewrite policy label. Minimum length = 1
regexforfindingurlinjavascript	<String>	Read-write	Name of the pattern set that contains the regular expressions, which match the URL in Java script. Minimum length = 1
regexforfindingurlincss	<String>	Read-write	Name of the pattern set that contains the regular expressions, which match the URL in the CSS. Minimum length = 1
regexforfindingurlinxcomponent	<String>	Read-write	Name of the pattern set that contains the regular expressions, which match the URL in X Component. Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
regexforfindingurlinxml	String	Read-write	match the URL in XML. Minimum length = 1
regexforfindingcustomurls	<String>	Read-write	Name of the pattern set that contains the regular expressions, which match the URLs in the custom content type other than HTML, CSS, XML, XCOMP, and JavaScript. The custom content type should be included in the patset ns_cvpn_custom_content_types. Minimum length = 1
clientconsumedcookies	<String>	Read-write	Specify the name of the pattern set containing the names of the cookies, which are allowed between the client and the server. If a pattern set is not specified, NetSCaler Gateway does not allow any cookies between the client and the server. A cookie that is not specified in the pattern set is handled by NetScaler Gateway on behalf of the client. Minimum length = 1
requirepersistentcookie	<String>	Read-write	Specify whether a persistent session cookie is set and accepted for clientless access. If this parameter is set to ON, COM objects, such as MSOffice, which are invoked by the browser can access the files using clientless access. Use caution because the persistent cookie is stored on the disk. Default value: OFF Possible values = ON, OFF
cssrewritepolicylabel	<String>	Read-only	The configured CSS rewrite policylabel. Minimum length = 1
xmlrewritepolicylabel	<String>	Read-only	The configured XML rewrite policylabel. Minimum length = 1
xcomponentrewritepolicylabel	<String>	Read-only	The configured X-Component rewrite policylabel. Minimum length = 1
isdefault	<Boolean>	Read-only	A value of true is returned if it is a default vpnclientlessrwprofile.
description	<String>	Read-only	Description of the clientless access profile.
builtin	<String[]>	Read-only	Flag to determine if vpn clientless rewrite profile is built-in or not. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnclientlessaccessprofile":{
    "profilename":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile/profilename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnclientlessaccessprofile":{
    "profilename":<String_value>,
    "urlrewritepolicylabel":<String_value>,
    "javascriptrewritepolicylabel":<String_value>,
    "reqhdrrewritepolicylabel":<String_value>,
    "reshdrrewritepolicylabel":<String_value>,
    "regexforfindingurlinjavascript":<String_value>,
    "regexforfindingurlincss":<String_value>,
    "regexforfindingurlinxcomponent":<String_value>,
    "regexforfindingurlinxml":<String_value>,
    "regexforfindingcustomurls":<String_value>,
    "clientconsumedcookies":<String_value>,
    "requirepersistentcookie":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile?**action=unset**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnclientlessaccessprofile":{
    "profilename":<String_value>,
    "urlrewritepolicylabel":true,
    "javascriptrewritepolicylabel":true,
    "reqhdrrewritepolicylabel":true,
    "reshdrrewritepolicylabel":true,
    "regexforfindingurlinjavascript":true,
    "regexforfindingurlincss":true,
    "regexforfindingurlinxcomponent":true,
    "regexforfindingurlinxml":true,
    "regexforfindingcustomurls":true,
    "clientconsumedcookies":true,
    "requirepersistentcookie":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnclientlessaccessprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnclientlessaccessprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnclientlessaccessprofile": [ {
    "profilename":<String_value>,
    "urlrewritepolicylabel":<String_value>,
    "javascriptrewritepolicylabel":<String_value>,
    "cssrewritepolicylabel":<String_value>,
    "xmlrewritepolicylabel":<String_value>,
    "xcomponentrewritepolicylabel":<String_value>,
    "reqhdrrewritepolicylabel":<String_value>,
    "reshdrrewritepolicylabel":<String_value>,
    "regexforfindingurlinjavascript":<String_value>,
    "regexforfindingurlincss":<String_value>,
    "regexforfindingurlinxcomponent":<String_value>,
    "regexforfindingurlinxml":<String_value>,
    "regexforfindingcustomurls":<String_value>,
    "clientconsumedcookies":<String_value>,
    "requirepersistentcookie":<String_value>,
    "isdefault":<Boolean_value>,
    "description":<String_value>,
    "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile/profilename_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile/profilename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile/profilename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnclientlessaccessprofile": [ {
    "profilename":<String_value>,
    "urlrewritepolicylabel":<String_value>,
    "javascriptrewritepolicylabel":<String_value>,
    "cssrewritepolicylabel":<String_value>,
    "xmlrewritepolicylabel":<String_value>,
    "xcomponentrewritepolicylabel":<String_value>,
    "reqhdrrewritepolicylabel":<String_value>,
    "reshdrrewritepolicylabel":<String_value>,
    "regexforfindingurlinjavascript":<String_value>,
    "regexforfindingurlincss":<String_value>,
    "regexforfindingurlinxcomponent":<String_value>,
    "regexforfindingurlinxml":<String_value>,
    "regexforfindingcustomurls":<String_value>,
    "clientconsumedcookies":<String_value>,
    "requirepersistentcookie":<String_value>,
    "isdefault":<Boolean_value>,
    "description":<String_value>,
    "builtin":<String[]_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnclientlessaccessprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnclientlessaccessprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpneula

May 20, 2016

Configuration for EULA for vservers resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the eula. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpneula

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpneula":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpneula/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpneula

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpneula?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpneula?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpneula resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpneula?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpneula?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpneula resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpneula": [ {
    "name":<String_value>,

}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpneula/name_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpneula/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpneula/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpneula": [ {
      "name":<String_value>,

}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpneula?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpneula": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnformssoaction

May 20, 2016

Configuration for Form sso action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the form based single sign-on profile. Minimum length = 1
actionurl	<String>	Read-write	Root-relative URL to which the completed form is submitted. Minimum length = 1
userfield	<String>	Read-write	Name of the form field in which the user types in the user ID. Minimum length = 1
passwdfield	<String>	Read-write	Name of the form field in which the user types in the password. Minimum length = 1
ssosuccessrule	<String>	Read-write	Advanced expression that defines the criteria for SSO success. Expression such as checking for cookie in the response is a common example.
namevaluepair	<String>	Read-write	Other name-value pair attributes to send to the server, in addition to sending the user name and password. Value names are separated by an ampersand (&), such as in name1=value1&name2=value2.
responsesize	<Double>	Read-write	Maximum number of bytes to allow in the response size. Specifies the number of bytes in the response to be parsed for extracting the forms. Default value: 8096
nvtype	<String>	Read-write	How to process the name-value pair. Available settings function as follows: * STATIC - The administrator-configured values are used. * DYNAMIC - The response is parsed, the form is extracted, and then submitted. Default value: DYNAMIC Possible values = STATIC, DYNAMIC
submitmethod	<String>	Read-write	HTTP method (GET or POST) used by the single sign-on form to send the logon credentials to the logon server. Default value: GET Possible values = GET, POST
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnformssoaction":{
    "name":<String_value>,
    "actionurl":<String_value>,
    "userfield":<String_value>,
    "passwdfield":<String_value>,
    "ssosuccessrule":<String_value>,
    "namevaluepair":<String_value>,
    "responsesize":<Double_value>,
    "nvtype":<String_value>,
    "submitmethod":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnformssoaction":{
    "name":<String_value>,
    "actionurl":<String_value>,
    "userfield":<String_value>,
    "passwdfield":<String_value>,
    "ssosuccessrule":<String_value>,
    "responsesize":<Double_value>,
    "namevaluepair":<String_value>,
    "nvtype":<String_value>,
    "submitmethod":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnformssoaction":{
    "name":<String_value>,
    "responsesize":true,
    "namevaluepair":true,
    "nvtype":true,
    "submitmethod":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnformssoaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnformssoaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnformssoaction": [ {
    "name":<String_value>,
    "actionurl":<String_value>,
    "userfield":<String_value>,
    "passwdfield":<String_value>,
    "responsesize":<Double_value>,
    "namevaluepair":<String_value>,
    "nvtype":<String_value>,
    "ssosuccessrule":<String_value>,
    "submitmethod":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction/name_value<String>?**attrs=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1,property-name2

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnformssoaction": [ {
      "name":<String_value>,
      "actionurl":<String_value>,
      "userfield":<String_value>,
      "passwdfield":<String_value>,
      "responsesize":<Double_value>,
      "namevaluepair":<String_value>,
      "nvtype":<String_value>,
      "ssosuccessrule":<String_value>,
      "submitmethod":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnformssoaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnformssoaction": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_appcontroller_binding

May 20, 2016

Binding object showing the appcontroller that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
appcontroller	<String>	Read-write	Configured App Controller server.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_appcontroller_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_appcontroller_binding":{
    "appcontroller":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_appcontroller_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_appcontroller_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_appcontroller_binding?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnglobal_appcontroller_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_appcontroller_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_appcontroller_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_appcontroller_binding": [ {
    "appcontroller":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_appcontroller_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_appcontroller_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_auditnslogpolicy_binding

May 20, 2016

Binding object showing the auditnslogpolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_auditnslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_auditnslogpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_auditnslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_auditnslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_auditnslogpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_auditnslogpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_binding

Jul 07, 2017

Binding object which returns the resources bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vpnglobal_intranetip6_binding	<vpnglobal_intranetip6_binding[]>	Read-only	intranetip6 that can be bound to vpnglobal.
vpnglobal_auditnslogpolicy_binding	<vpnglobal_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to vpnglobal.
vpnglobal_authenticationpolicy_binding	<vpnglobal_authenticationpolicy_binding[]>	Read-only	authenticationpolicy that can be bound to vpnglobal.
vpnglobal_authenticationradiuspolicy_binding	<vpnglobal_authenticationradiuspolicy_binding[]>	Read-only	authenticationradiuspolicy that can be bound to vpnglobal.
vpnglobal_vpnsessionpolicy_binding	<vpnglobal_vpnsessionpolicy_binding[]>	Read-only	vpnsessionpolicy that can be bound to vpnglobal.
vpnglobal_domain_binding	<vpnglobal_domain_binding[]>	Read-only	domain that can be bound to vpnglobal.
vpnglobal_vpneula_binding	<vpnglobal_vpneula_binding[]>	Read-only	vpneula that can be bound to vpnglobal.
vpnglobal_vpnclientlessaccesspolicy_binding	<vpnglobal_vpnclientlessaccesspolicy_binding[]>	Read-only	vpnclientlessaccesspolicy that can be bound to vpnglobal.
vpnglobal_intranetip_binding	<vpnglobal_intranetip_binding[]>	Read-only	intranetip that can be bound to vpnglobal.
vpnglobal_vpnnexthopserver_binding	<vpnglobal_vpnnexthopserver_binding[]>	Read-only	vpnnexthopserver that can be bound to vpnglobal.
vpnglobal_authenticationldappolicy_binding	<vpnglobal_authenticationldappolicy_binding[]>	Read-only	authenticationldappolicy that can be bound to vpnglobal.
vpnglobal_sharefileserver_binding	<vpnglobal_sharefileserver_binding[]>	Read-only	sharefileserver that can be bound to vpnglobal.
vpnglobal_vpntrafficpolicy_binding	<vpnglobal_vpntrafficpolicy_binding[]>	Read-only	vpntrafficpolicy that can be bound to vpnglobal.
vpnglobal_authenticationlocalpolicy_binding	<vpnglobal_authenticationlocalpolicy_binding[]>	Read-only	authenticationlocalpolicy that can be bound to vpnglobal.
vpnglobal_vpnintranetapplication_binding	<vpnglobal_vpnintranetapplication_binding[]>	Read-only	vpnintranetapplication that can be bound to vpnglobal.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
vpnglobal_authenticationnegotiatepolicy_binding	<vpnglobal_authenticationnegotiatepolicy_binding[]>	Read-only	authenticationnegotiatepolicy that can be bound to vpnglobal.
vpnglobal_authenticationtacacspolicy_binding	<vpnglobal_authenticationtacacspolicy_binding[]>	Read-only	authenticationtacacspolicy that can be bound to vpnglobal.
vpnglobal_vpnportaltheme_binding	<vpnglobal_vpnportaltheme_binding[]>	Read-only	vpnportaltheme that can be bound to vpnglobal.
vpnglobal_authenticationsamlpolicy_binding	<vpnglobal_authenticationsamlpolicy_binding[]>	Read-only	authenticationsamlpolicy that can be bound to vpnglobal.
vpnglobal_staserver_binding	<vpnglobal_staserver_binding[]>	Read-only	staserver that can be bound to vpnglobal.
vpnglobal_auditsyslogpolicy_binding	<vpnglobal_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to vpnglobal.
vpnglobal_authenticationcertpolicy_binding	<vpnglobal_authenticationcertpolicy_binding[]>	Read-only	authenticationcertpolicy that can be bound to vpnglobal.
vpnglobal_vpnurl_binding	<vpnglobal_vpnurl_binding[]>	Read-only	vpnurl that can be bound to vpnglobal.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_binding

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_binding": [ {
    "vpnglobal_vpnintranetapplication_binding":<vpnglobal_vpnintranetapplication_binding[]_value>,
    "vpnglobal_authenticationsamlpolicy_binding":<vpnglobal_authenticationsamlpolicy_binding[]_value>,
    "vpnglobal_intranetip_binding":<vpnglobal_intranetip_binding[]_value>,
    "vpnglobal_authenticationradiuspolicy_binding":<vpnglobal_authenticationradiuspolicy_binding[]_value>,
    "vpnglobal_staserver_binding":<vpnglobal_staserver_binding[]_value>,
    "vpnglobal_sharefileserver_binding":<vpnglobal_sharefileserver_binding[]_value>,
    "vpnglobal_intranetip6_binding":<vpnglobal_intranetip6_binding[]_value>,
    "vpnglobal_appcontroller_binding":<vpnglobal_appcontroller_binding[]_value>,
    "vpnglobal_authenticationtacacspolicy_binding":<vpnglobal_authenticationtacacspolicy_binding[]_value>,
    "vpnglobal_authenticationpolicy_binding":<vpnglobal_authenticationpolicy_binding[]_value>,
    "vpnglobal_vpntrafficpolicy_binding":<vpnglobal_vpntrafficpolicy_binding[]_value>,
    "vpnglobal_domain_binding":<vpnglobal_domain_binding[]_value>,
    "vpnglobal_vpnnexthopserver_binding":<vpnglobal_vpnnexthopserver_binding[]_value>,
    "vpnglobal_authenticationldappolicy_binding":<vpnglobal_authenticationldappolicy_binding[]_value>,
    "vpnglobal_vpnsessionpolicy_binding":<vpnglobal_vpnsessionpolicy_binding[]_value>,
    "vpnglobal_authenticationcertpolicy_binding":<vpnglobal_authenticationcertpolicy_binding[]_value>,
    "vpnglobal_authenticationnegotiatepolicy_binding":<vpnglobal_authenticationnegotiatepolicy_binding[]_value>,
    "vpnglobal_auditsyslogpolicy_binding":<vpnglobal_auditsyslogpolicy_binding[]_value>,
    "vpnglobal_vpneula_binding":<vpnglobal_vpneula_binding[]_value>,
    "vpnglobal_vpnurl_binding":<vpnglobal_vpnurl_binding[]_value>,
    "vpnglobal_vpnportaltheme_binding":<vpnglobal_vpnportaltheme_binding[]_value>,
    "vpnglobal_vpnclientlessaccesspolicy_binding":<vpnglobal_vpnclientlessaccesspolicy_binding[]_value>,
    "vpnglobal_authenticationlocalpolicy_binding":<vpnglobal_authenticationlocalpolicy_binding[]_value>,
    "vpnglobal_auditnslogpolicy_binding":<vpnglobal_auditnslogpolicy_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnicadtlsconnection

May 20, 2016

Configuration for active ica connections resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	User name for which to display connections. Minimum length = 1
domain	<String>	Read-only	The domain name.
srcip	<String>	Read-only	The client IP address.
srcport	<Integer>	Read-only	The client port. Range 1 - 65535
destip	<String>	Read-only	The CPS server IP address.
destport	<Integer>	Read-only	The CPS server port. Range 1 - 65535
channelnumber	<Double>	Read-only	The channel number.
peid	<Double>	Read-only	Core id of the session owner.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnicadtlsconnection

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/config/vpnicadtlsconnection?**args=username:<String_value>**

Use this query-parameter to get vpnicadtlsconnection resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnicadtlsconnection?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnicadtlsconnection?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnicadtlsconnection resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnicadtlsconnection?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/vpnicadtlsconnection?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnicadtlsconnection resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnicadtlsconnection": [ {
username:<String_value>      "domain":<String_value>,
      "srcip":<String_value>,
      "srcport":<Integer_value>,
      "destip":<String_value>,
      "destport":<Integer_value>,
      "channelnumber":<Double_value>,
      "peid":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnicadtlsconnection?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnicadtlsconnection": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_auditsyslogpolicy_binding

May 20, 2016

Binding object showing the auditsyslogpolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_auditsyslogpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_auditsyslogpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_auditsyslogpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_authenticationcertpolicy_binding

May 20, 2016

Binding object showing the authenticationcertpolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_authenticationcertpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationcertpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationcertpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationcertpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationcertpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationcertpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_authenticationldappolicy_binding

May 20, 2016

Binding object showing the authenticationldappolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_authenticationldappolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_authenticationldappolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationldappolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationldappolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationldappolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationldappolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationldappolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_authenticationlocalpolicy_binding

May 20, 2016

Binding object showing the authenticationlocalpolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_authenticationlocalpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationlocalpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationlocalpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationlocalpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationlocalpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationlocalpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_authenticationnegotiatepolicy_binding

May 20, 2016

Binding object showing the authenticationnegotiatepolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationnegotiatepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationnegotiatepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationnegotiatepolicy_binding": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationnegotiatepolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationnegotiatepolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_authenticationpolicy_binding

May 20, 2016

Binding object showing the authenticationpolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_authenticationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_authenticationpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_authenticationradiuspolicy_binding

Jul 07, 2017

Binding object showing the authenticationradiuspolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_authenticationradiuspolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationradiuspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationradiuspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationradiuspolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationradiuspolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationradiuspolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_authenticationsamlpolicy_binding

Jul 07, 2017

Binding object showing the authenticationsamlpolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_authenticationsamlpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationsamlpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationsamlpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationsamlpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationsamlpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationsamlpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_domain_binding

May 20, 2016

Binding object showing the domain that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
intranetdomain	<String>	Read-write	The conflicting intranet domain name.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_domain_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_domain_binding":{
    "intranetdomain":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_domain_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_domain_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_domain_binding?**filter=property-name1:property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnglobal_domain_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_domain_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_domain_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_domain_binding": [ {
    "intranetdomain":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_domain_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_domain_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_authenticationtacacspolicy_binding

May 20, 2016

Binding object showing the authenticationtacacspolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_authenticationtacacspolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_authenticationtacacspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_authenticationtacacspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationtacacspolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_authenticationtacacspolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_authenticationtacacspolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_intranetip6_binding

May 20, 2016

Binding object showing the intranetip6 that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
intranetip6	<String>	Read-write	The intranet ip address or range.
numaddr	<Double>	Read-write	The intranet ip address or range's netmask.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_intranetip6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_intranetip6_binding":{
    "intranetip6":<String_value>,
    "numaddr":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_intranetip6_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_intranetip6_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_intranetip6_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_intranetip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_intranetip6_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_intranetip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_intranetip6_binding": [ {
      "intranetip6":<String_value>,
      "numaddr":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_intranetip6_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_intranetip6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_intranetip_binding

May 20, 2016

Binding object showing the intranetip that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
intranetip	<String>	Read-write	The intranet ip address or range.
netmask	<String>	Read-write	The intranet ip address or range's netmask.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_intranetip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_intranetip_binding":{
    "intranetip":<String_value>,
    "netmask":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_intranetip_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_intranetip_binding

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_intranetip_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_intranetip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_intranetip_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_intranetip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_intranetip_binding": [ {
      "intranetip":<String_value>,
      "netmask":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_intranetip_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_intranetip_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_sharefileserver_binding

Jul 07, 2017

Binding object showing the sharefileserver that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sharefile	<String>	Read-write	Configured Sharefile server, in the format IP:PORT / FQDN:PORT.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_sharefileserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_sharefileserver_binding":{
    "sharefile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_sharefileserver_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_sharefileserver_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_sharefileserver_binding?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnglobal_sharefileserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_sharefileserver_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_sharefileserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_sharefileserver_binding": [ {
    "sharefile":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_sharefileserver_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_sharefileserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_staserver_binding

May 20, 2016

Binding object showing the staserver that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
staserver	<String>	Read-write	Configured Secure Ticketing Authority (STA) server.
staaddresstype	<String>	Read-write	Type of the STA server address(ipv4/v6). Possible values = IPV4, IPV6
staauthid	<String>	Read-only	Authority ID of the STA Server. Authority ID is used to match incoming STA Tickets in the SOCKS/CGP protocol with the right STA Server.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_staserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_staserver_binding":{
    "staserver":<String_value>,
    "staaddresstype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_staserver_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_staserver_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_staserver_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_staserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_staserver_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_staserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_staserver_binding": [ {
    "staserver":<String_value>,
    "staaddresstype":<String_value>,
    "staauthid":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_staserver_binding?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_staserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_vpnclientlessaccesspolicy_binding

May 20, 2016

Binding object showing the vpnclientlessaccesspolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
globalbindtype	<String>	Read-write	. Possible values = SYSTEM_GLOBAL, VPN_GLOBAL, RNAT_GLOBAL
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
type	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQ_OVERRIDE, REQ_DEFAULT, RES_OVERRIDE, RES_DEFAULT
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_vpnclientlessaccesspolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpnclientlessaccesspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_vpnclientlessaccesspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "vpnglobal_vpnclientlessaccesspolicy_binding": [ {
    "priority":<Double_value>,
    "globalbindtype":<String_value>,
    "builtin":<String[]_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "type":<String_value>,
    "groupextraction":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnclientlessaccesspolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_vpnclientlessaccesspolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_vpneula_binding

May 20, 2016

Binding object showing the vpneula that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
eula	<String>	Read-write	Name of the EULA bound to vpnglobal.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_vpneula_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_vpneula_binding":{
    "eula":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpneula_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpneula_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpneula_binding?**filter=property-name1:property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnglobal_vpneula_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpneula_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_vpneula_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_vpneula_binding": [ {
    "eula":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpneula_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_vpneula_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_vpnintranetapplication_binding

May 20, 2016

Binding object showing the vpnintranetapplication that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
intranetapplication	<String>	Read-write	The intranet vpn application.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_vpnintranetapplication_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_vpnintranetapplication_binding":{
    "intranetapplication":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnglobal_vpnintranetapplication_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_vpnintranetapplication_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_vpnintranetapplication_binding": [ {
      "intranetapplication":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnintranetapplication_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_vpnintranetapplication_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_vpnnexthopserver_binding

May 20, 2016

Binding object showing the vpnnexthopserver that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
nexthopserver	<String>	Read-write	The name of the next hop server bound to vpn global.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_vpnnexthopserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_vpnnexthopserver_binding":{
    "nexthopserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnglobal_vpnnexthopserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_vpnnexthopserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_vpnnexthopserver_binding": [ {
    "nexthopserver":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnnexthopserver_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_vpnnexthopserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_vpnportaltheme_binding

May 20, 2016

Binding object showing the vpnportaltheme that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
portaltheme	<String>	Read-write	Name of the portal theme bound to vpnglobal.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_vpnportaltheme_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_vpnportaltheme_binding":{
    "portaltheme":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnportaltheme_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnportaltheme_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnportaltheme_binding?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnglobal_vpnportaltheme_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnportaltheme_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_vpnportaltheme_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_vpnportaltheme_binding": [ {
    "portaltheme":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnportaltheme_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_vpnportaltheme_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnurl

May 20, 2016

Configuration for VPN URL resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
urlname	<String>	Read-write	Name of the bookmark link. Minimum length = 1
linkname	<String>	Read-write	Description of the bookmark link. The description appears in the Access Interface. Minimum length = 1
actualurl	<String>	Read-write	Web address for the bookmark link. Minimum length = 1
vservername	<String>	Read-write	Name of the associated LB/CS vserver.
clientlessaccess	<String>	Read-write	If clientless access to the resource hosting the link is allowed, also use clientless access for the bookmarked web address in the Secure Client Access based session. Allows single sign-on and other HTTP processing on NetScaler Gateway for HTTPS resources. Default value: OFF Possible values = ON, OFF
comment	<String>	Read-write	Any comments associated with the bookmark link.
iconurl	<String>	Read-write	URL to fetch icon file for displaying this resource.
ssotype	<String>	Read-write	Single sign on type for unified gateway. Possible values = unifiedgateway, selfauth, samlauth
applicationtype	<String>	Read-write	The type of application this VPN URL represents. Possible values are CVPN/SaaS/VPN. Possible values = CVPN, VPN, SaaS
samlssoprofile	<String>	Read-write	Profile to be used for doing SAML SSO.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnurl

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnurl":{
    "urlname":<String_value>,
    "linkname":<String_value>,
    "actualurl":<String_value>,
    "vservername":<String_value>,
    "clientlessaccess":<String_value>,
    "comment":<String_value>,
    "iconurl":<String_value>,
    "ssotype":<String_value>,
    "applicationtype":<String_value>,
    "samlssoprofile":<String_value>
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnurl/urlname_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnurl

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnurl":{
    "urlname":<String_value>,
    "linkname":<String_value>,
    "actualurl":<String_value>,
    "vservername":<String_value>,
    "clientlessaccess":<String_value>,
    "comment":<String_value>,
    "iconurl":<String_value>,
    "ssotype":<String_value>,
    "applicationtype":<String_value>,
    "samlssoprofile":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnurl?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnurl":{
    "urlname":<String_value>,
    "vservername":true,
    "clientlessaccess":true,
    "comment":true,
    "iconurl":true,
    "ssotype":true,
    "applicationtype":true,
    "samlssoprofile":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnurl

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnurl?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnurl?**filter=property-name1:property-val1,property-name2:property-val2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the filtered set of vpnurl resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnurl?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnurl?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnurl resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnurl": [ {
      "urlname":<String_value>,
      "linkname":<String_value>,
      "vservername":<String_value>,
      "actualurl":<String_value>,
      "clientlessaccess":<String_value>,
      "comment":<String_value>,
      "iconurl":<String_value>,
      "ssotype":<String_value>,
      "applicationtype":<String_value>,
      "samlssoprofile":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnurl/urlname_value<String>

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnurl/urlname_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnurl/urlname_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnurl": [ {
      "urlname":<String_value>,
      "linkname":<String_value>,
      "vservername":<String_value>,
      "actualurl":<String_value>,
      "clientlessaccess":<String_value>,
      "comment":<String_value>,
      "iconurl":<String_value>,
      "ssotype":<String_value>,
      "applicationtype":<String_value>,
      "samlssoprofile":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnurl?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnurl": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_vpnsessionpolicy_binding

May 20, 2016

Binding object showing the vpnsessionpolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
builtin	<String[]>	Read-write	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and

© 1999-2017 Citrix Systems, Inc. All rights reserved.

then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_vpnsessionpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpnsessionpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_vpnsessionpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_vpnsessionpolicy_binding": [ {
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "builtin":<String[]_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnsessionpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_vpnsessionpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_vpntrafficpolicy_binding

Jul 07, 2017

Binding object showing the vpntrafficpolicy that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority of the policy.
policyname	<String>	Read-write	The name of the policy.
secondary	<Boolean>	Read-write	Bind the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only to a primary authentication server but also to a secondary authentication server. User groups are aggregated across both authentication servers. The user name must be exactly the same on both authentication servers, but the authentication servers can require different passwords.
groupextraction	<Boolean>	Read-write	Bind the Authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called it primary and/or secondary authentication has succeeded.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_vpntrafficpolicy_binding":{
    "policyname":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnglobal_vpntrafficpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_vpntrafficpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_vpntrafficpolicy_binding": [ {
    "priority":<Double_value>,
    "policyname":<String_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpntrafficpolicy_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_vpntrafficpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnglobal_vpnurl_binding

May 20, 2016

Binding object showing the vpnurl that can be bound to vpnglobal.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
urlname	<String>	Read-write	The intranet url.
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnglobal_vpnurl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnglobal_vpnurl_binding":{
    "urlname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnurl_binding

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnurl_binding

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnurl_binding?**filter=property-name1:property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnglobal_vpnurl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnurl_binding?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnglobal_vpnurl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnglobal_vpnurl_binding": [ {
    "urlname":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnglobal_vpnurl_binding?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnglobal_vpnurl_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnparameter

May 20, 2016

Configuration for VPN parameter resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
httpport	<Integer[]>	Read-write	Destination port numbers other than port 80, added as a comma-separated list. Traffic to these ports is processed as HTTP traffic, which allows functionality, such as HTTP authorization and single sign-on to a web application to work. Minimum value = 1
winsip	<String>	Read-write	WINS server IP address to add to NetScaler Gateway for name resolution. Minimum length = 1
dnsvservername	<String>	Read-write	Name of the DNS virtual server for the user session. Minimum length = 1
splitdns	<String>	Read-write	Route the DNS requests to the local DNS server configured on the user device, or NetScaler Gateway (remote), or both. Possible values = LOCAL, REMOTE, BOTH
sesstimeout	<Double>	Read-write	Number of minutes after which the session times out. Default value: 30 Minimum value = 1 Maximum value = 65535
clientsecurity	<String>	Read-write	Specify the client security check for the user device to permit a NetScaler Gateway session. The web address or IP address is not included in the expression for the client security check.
clientsecuritygroup	<String>	Read-write	The client security group that will be assigned on failure of the client security check. Users can in general be organized into Groups. In this case, the Client Security Group may have a more restrictive security policy. Minimum length = 1
clientsecuritymessage	<String>	Read-write	The client security message that will be displayed on failure of the client security check. Minimum length = 1 Maximum length = 127
clientsecuritylog	<String>	Read-write	Specifies whether or not to display all failed Client Security scans to the end user. Default value: OFF Possible values = ON, OFF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
smartgroup	<String>	Read-write	the groups that the user is a part of is configured on NetScaler Gateway. The ica (smartacccess2) uses these groups in the is_member_of expressions and the vpn session policy will have the expression to check for presence of antivirus. Minimum length = 1 Maximum length = 64
splittunnel	<String>	Read-write	Send, through the tunnel, traffic only for intranet applications that are defined in NetScaler Gateway. Route all other traffic directly to the Internet. The OFF setting routes all traffic through NetScaler Gateway. With the REVERSE setting, intranet applications define the network traffic that is not intercepted. All network traffic directed to internal IP addresses bypasses the VPN tunnel, while other traffic goes through NetScaler Gateway. Reverse split tunneling can be used to log all non-local LAN traffic. For example, if users have a home network and are logged on through the NetScaler Gateway Plug-in, network traffic destined to a printer or another device within the home network is not intercepted. Default value: OFF Possible values = ON, OFF, REVERSE
locallanaccess	<String>	Read-write	Set local LAN access. If split tunneling is OFF, and you set local LAN access to ON, the local client can route traffic to its local interface. When the local area network switch is specified, this combination of switches is useful. The client can allow local LAN access to devices that commonly have non-routable addresses, such as local printers or local file servers. Default value: OFF Possible values = ON, OFF
rfc1918	<String>	Read-write	As defined in the local area network, allow only the following local area network addresses to bypass the VPN tunnel when the local LAN access feature is enabled: * 10.*.*.*, * 172.16.*.*, * 192.168.*.*. Default value: OFF Possible values = ON, OFF
spoofiip	<String>	Read-write	Indicate whether or not the application requires IP spoofing, which routes the connection to the intranet application through the virtual adapter. Default value: ON Possible values = ON, OFF
killconnections	<String>	Read-write	Specify whether the NetScaler Gateway Plug-in should disconnect all preexisting connections, such as the connections existing before the user logged on to NetScaler Gateway, and prevent new incoming connections on the NetScaler Gateway Plug-in for Windows and MAC when the user is connected to NetScaler Gateway and split tunneling is disabled. Default value: OFF Possible values = ON, OFF
			Allow access to network resources by using a single IP address and subnet mask or a range of IP addresses. The OFF setting sets the mode to proxy, in which you configure destination and source IP addresses and port

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			this parameter to ON, in which the mode is set to transparent. If you are using the NetScaler Gateway Plug-in for Java, set this parameter to OFF. Default value: OFF Possible values = ON, OFF
windowsclienttype	<String>	Read-write	The Windows client type. Choose between two types of Windows Client\ a) Application Agent - which always runs in the task bar as a standalone application and also has a supporting service which runs permanently when installed\ b) Activex Control - ActiveX control run by Microsoft Internet Explorer. Default value: AGENT Possible values = AGENT, PLUGIN
defaultauthorizationaction	<String>	Read-write	Specify the network resources that users have access to when they log on to the internal network. The default setting for authorization is to deny access to all network resources. Citrix recommends using the default global setting and then creating authorization policies to define the network resources users can access. If you set the default authorization policy to DENY, you must explicitly authorize access to any network resource, which improves security. Default value: DENY Possible values = ALLOW, DENY
authorizationgroup	<String>	Read-write	Comma-separated list of groups in which the user is placed when none of the groups that the user is a part of is configured on NetScaler Gateway. The authorization policy can be bound to these groups to control access to the resources. Minimum length = 1
clientidletimeout	<Double>	Read-write	Time, in minutes, after which to time out the user session if NetScaler Gateway does not detect mouse or keyboard activity. Minimum value = 1 Maximum value = 9999
proxy	<String>	Read-write	Set options to apply proxy for accessing the internal resources. Available settings function as follows: * BROWSER - Proxy settings are configured only in Internet Explorer and Firefox browsers. * NS - Proxy settings are configured on the NetScaler appliance. * OFF - Proxy settings are not configured. Possible values = BROWSER, NS, OFF
allprotocolproxy	<String>	Read-write	IP address of the proxy server to use for all protocols supported by NetScaler Gateway. Minimum length = 1
httpproxy	<String>	Read-write	IP address of the proxy server to be used for HTTP access for all subsequent connections to the internal network. Minimum length = 1
ftpproxy	<String>	Read-write	IP address of the proxy server to be used for FTP access for all subsequent connections to the internal network. Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
socksproxy	<String>	Read-write	subsequent connections to the internal network. Minimum length = 1
gopherproxy	<String>	Read-write	IP address of the proxy server to be used for GOPHER access for all subsequent connections to the internal network. Minimum length = 1
sslproxy	<String>	Read-write	IP address of the proxy server to be used for SSL access for all subsequent connections to the internal network. Minimum length = 1
proxyexception	<String>	Read-write	Proxy exception string that will be configured in the browser for bypassing the previously configured proxies. Allowed only if proxy type is Browser. Minimum length = 1
proxylocalbypass	<String>	Read-write	Bypass proxy server for local addresses option in Internet Explorer and Firefox proxy server settings. Default value: DISABLED Possible values = ENABLED, DISABLED
clientcleanupprompt	<String>	Read-write	Prompt for client-side cache clean-up when a client-initiated session closes. Default value: ON Possible values = ON, OFF
forcecleanup	<String[]>	Read-write	Force cache clean-up when the user closes a session. You can specify all, none, or any combination of the client-side items. Possible values = none, all, cookie, addressbar, plugin, filesystemapplication, application, applicationdata, clientcertificate, autocomplete, cache
clientoptions	<String[]>	Read-write	Display only the configured menu options when you select the "Configure NetScaler Gateway" option in the NetScaler Gateway Plug-in's system tray icon for Windows. Possible values = none, all, services, filetransfer, configuration
clientconfiguration	<String[]>	Read-write	Display only the configured tabs when you select the "Configure NetScaler Gateway" option in the NetScaler Gateway Plug-in's system tray icon for Windows. Possible values = none, all, general, tunnel, trace, compression
sso	<String>	Read-write	Set single sign-on (SSO) for the session. When the user accesses a server, the user's logon credentials are passed to the server for authentication. Default value: OFF Possible values = ON, OFF
ssocredential	<String>	Read-write	Specify whether to use the primary or secondary authentication credentials for single sign-on to the server. Default value: PRIMARY Possible values = PRIMARY, SECONDARY
			Enable or disable the Windows Auto Logon for the session. If a VPN session is established after this setting is enabled, the user is automatically

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: OFF Possible values = ON, OFF
usemip	<String>	Read-write	Enable or disable the use of a unique IP address alias, or a mapped IP address, as the client IP address for each client session. Allow NetScaler Gateway to use the mapped IP address as an intranet IP address when all other IP addresses are not available. When IP pooling is configured and the mapped IP is used as an intranet IP address, the mapped IP address is used when an intranet IP address cannot be assigned. Default value: NS Possible values = NS, OFF
useiip	<String>	Read-write	Define IP address pool options. Available settings function as follows: * SPILLOVER - When an address pool is configured and the mapped IP is used as an intranet IP address, the mapped IP address is used when an intranet IP address cannot be assigned. * NOSPILLOVER - When intranet IP addresses are enabled and the mapped IP address is not used, the Transfer Login page appears for users who have used all available intranet IP addresses. * OFF - Address pool is not configured. Default value: NOSPILLOVER Possible values = NOSPILLOVER, SPILLOVER, OFF
clientdebug	<String>	Read-write	Set the trace level on NetScaler Gateway. Technical support technicians use these debug logs for in-depth debugging and troubleshooting purposes. Available settings function as follows: * DEBUG - Detailed debug messages are collected and written into the specified file. * STATS - Application audit level error messages and debug statistic counters are written into the specified file. * EVENTS - Application audit-level error messages are written into the specified file. * OFF - Only critical events are logged into the Windows Application Log. Default value: OFF Possible values = debug, stats, events, OFF
loginscript	<String>	Read-write	Path to the logon script that is run when a session is established. Separate multiple scripts by using comma. A "$" in the path signifies that the word following the "$" is an environment variable. Minimum length = 1
logoutscript	<String>	Read-write	Path to the logout script. Separate multiple scripts by using comma. A "$" in the path signifies that the word following the "$" is an environment variable. Minimum length = 1
homepage	<String>	Read-write	Web address of the home page that appears when users log on. Otherwise, users receive the default home page for NetScaler Gateway, which is the Access Interface.
			Enable ICA proxy to configure secure Internet access to servers running Citrix XenApp or XenDesktop by using Citrix Receiver instead of the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Default value: OFF Possible values = ON, OFF
wihome	<String>	Read-write	Web address of the Web Interface server, such as http://<ipAddress>/Citrix/XenApp, or Receiver for Web, which enumerates the virtualized resources, such as XenApp, XenDesktop, and cloud applications. This web address is used as the home page in ICA proxy mode. If Client Choices is ON, you must configure this setting. Because the user can choose between FullClient and ICAProxy, the user may see a different home page. An Internet web site may appear if the user gets the FullClient option, or a Web Interface site if the user gets the ICAProxy option. If the setting is not configured, the XenApp option does not appear as a client choice.
wihomeaddresstype	<String>	Read-write	Type of the wihome address(IPV4/V6). Possible values = IPV4, IPV6
citrixreceiverhome	<String>	Read-write	Web address for the Citrix Receiver home page. Configure NetScaler Gateway so that when users log on to the appliance, the NetScaler Gateway Plug-in opens a web browser that allows single sign-on to the Citrix Receiver home page.
wiportalmode	<String>	Read-write	Layout on the Access Interface. The COMPACT value indicates the use of small icons. Possible values = NORMAL, COMPACT
clientchoices	<String>	Read-write	Provide users with multiple logon options. With client choices, users have the option of logging on by using the NetScaler Gateway Plug-in for Windows, NetScaler Gateway Plug-in for Java, the Web Interface, or clientless access from one location. Depending on how NetScaler Gateway is configured, users are presented with up to three icons for logon choices. The most common are the NetScaler Gateway Plug-in for Windows, Web Interface, and clientless access. Default value: OFF Possible values = ON, OFF
epaclienttype	<String>	Read-write	Choose between two types of End point Windows Client a) Application Agent - which always runs in the task bar as a standalone application and also has a supporting service which runs permanently when installed b) Activex Control - ActiveX control run by Microsoft Internet Explorer. Possible values = AGENT, PLUGIN
iipdnssuffix	<String>	Read-write	An intranet IP DNS suffix. When a user logs on to NetScaler Gateway and is assigned an IP address, a DNS record for the user name and IP address combination is added to the NetScaler Gateway DNS cache. You can configure a DNS suffix to append to the user name when the DNS record is added to the cache. You can reach to the host from where the user is logged on by using the user's name, which can be easier to remember than an IP address. When the user logs off from NetScaler Gateway, the record is removed from the DNS cache. Minimum length = 1
			Force a disconnection from the NetScaler Gateway Plug-in with NetScaler

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
forcedtimeout	<Double>	Read-write	user must log on again. Minimum value = 1 Maximum value = 65535
forcedtimeoutwarning	<Double>	Read-write	Number of minutes to warn a user before the user session is disconnected. Minimum value = 1 Maximum value = 255
ntdomain	<String>	Read-write	Single sign-on domain to use for single sign-on to applications in the internal network. This setting can be overwritten by the domain that users specify at the time of logon or by the domain that the authentication server returns. Minimum length = 1 Maximum length = 32
clientlessvpnmode	<String>	Read-write	Enable clientless access for web, XenApp or XenDesktop, and FileShare resources without installing the NetScaler Gateway Plug-in. Available settings function as follows: * ON - Allow only clientless access. * OFF - Allow clientless access after users log on with the NetScaler Gateway Plug-in. * DISABLED - Do not allow clientless access. Default value: OFF Possible values = ON, OFF, DISABLED
clientlessmodeurlencoding	<String>	Read-write	When clientless access is enabled, you can choose to encode the addresses of internal web applications or to leave the address as clear text. Available settings function as follows: * OPAQUE - Use standard encoding mechanisms to make the domain and protocol part of the resource unclear to users. * TRANSPARENT - Do not encode the web address and make it visible to users. * ENCRYPT - Allow the domain and protocol to be encrypted using a session key. When the web address is encrypted, the URL is different for each user session for the same web resource. If users bookmark the encoded web address, save it in the web browser and then log off, they cannot connect to the web address when they log on and use the bookmark. If users save the encrypted bookmark in the Access Interface during their session, the bookmark works each time the user logs on. Default value: OPAQUE Possible values = TRANSPARENT, OPAQUE, ENCRYPT
clientlesspersistentcookie	<String>	Read-write	State of persistent cookies in clientless access mode. Persistent cookies are required for accessing certain features of SharePoint, such as opening and editing Microsoft Word, Excel, and PowerPoint documents hosted on the SharePoint server. A persistent cookie remains on the user device and is sent with each HTTP request. NetScaler Gateway encrypts the persistent cookie before sending it to the plug-in on the user device, and refreshes the cookie periodically as long as the session exists. The cookie becomes stale if the session ends. Available settings function as follows: * ALLOW - Enable persistent cookies. Users can open and edit Microsoft documents stored in SharePoint. * DENY - Disable persistent cookies. Users cannot open and edit Microsoft

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			* PROMPT - Prompt users to allow or deny persistent cookies during the session. Persistent cookies are not required for clientless access if users do not connect to SharePoint. Default value: DENY Possible values = ALLOW, DENY, PROMPT
emailhome	<String>	Read-write	Web address for the web-based email, such as Outlook Web Access.
allowedlogingroups	<String>	Read-write	Specify groups that have permission to log on to NetScaler Gateway. Users who do not belong to this group or groups are denied access even if they have valid credentials. Minimum length = 1 Maximum length = 511
encryptcsecexp	<String>	Read-write	Enable encryption of client security expressions. Default value: ENABLED Possible values = ENABLED, DISABLED
apptokentimeout	<Double>	Read-write	The timeout value in seconds for tokens to access XenMobile applications. Default value: 100 Minimum value = 1 Maximum value = 255
mdxtokentimeout	<Double>	Read-write	Validity of MDX Token in minutes. This token is used for mdx services to access backend and valid HEAD and GET request. Default value: 10 Minimum value = 1 Maximum value = 1440
uitheme	<String>	Read-write	Set VPN UI Theme to Green-Bubble, Caxton or Custom; default is Caxton. Possible values = DEFAULT, GREENBUBBLE, CUSTOM
securebrowse	<String>	Read-write	Allow users to connect through NetScaler Gateway to network resources from iOS and Android mobile devices with Citrix Receiver. Users do not need to establish a full VPN tunnel to access resources in the secure network. Default value: ENABLED Possible values = ENABLED, DISABLED
storefronturl	<String>	Read-write	Web address for StoreFront to be used in this session for enumeration of resources from XenApp or XenDesktop. Minimum length = 1 Maximum length = 255
kcdaccount	<String>	Read-write	The KCD account details to be used in SSO.
clientversions	<String>	Read-write	checkversion api. Minimum length = 1 Maximum length = 100

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1 Maximum length = 127
windowspluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Win. Default value: Always Possible values = Always, Essential, Never
macpluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Mac. Default value: Always Possible values = Always, Essential, Never
linuxpluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Linux. Default value: Always Possible values = Always, Essential, Never
iconwithreceiver	<String>	Read-write	Option to decide whether to show plugin icon along with receiver icon. Default value: OFF Possible values = ON, OFF
userdomains	<String>	Read-write	List of user domains specified as comma seperated value.
icasessiontimeout	<String>	Read-write	Enable or disable ica session timeout. If enabled and in case AAA session gets terminated, ICA connections associated with that will also get terminated. Default value: OFF Possible values = ON, OFF
name	<String>	Read-only	The VPN name.
clientidletimeoutwarning	<Double>	Read-only	The time after which the client gets a timeout warning, in minutes.

Operations (click to see Properties)

UPDATE | UNSET | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnparameter

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnparameter":{
    "httpport":<Integer[]_value>,
    "winsip":<String_value>,
    "dnsvservername":<String_value>,
    "splitdns":<String_value>,
    "sesstimeout":<Double_value>,
    "clientsecurity":<String_value>,
    "clientsecuritygroup":<String_value>,
    "clientsecuritymessage":<String_value>,
    "clientsecuritylog":<String_value>,
    "smartgroup":<String_value>,
    "splittunnel":<String_value>,
    "locallanaccess":<String_value>,
    "rfc1918":<String_value>,
    "spoofiip":<String_value>,
    "killconnections":<String_value>,
    "transparentinterception":<String_value>,
    "windowsclienttype":<String_value>,
    "defaultauthorizationaction":<String_value>,
    "authorizationgroup":<String_value>,
    "clientidletimeout":<Double_value>,
    "proxy":<String_value>,
    "allprotocolproxy":<String_value>,
    "httpproxy":<String_value>,
    "ftpproxy":<String_value>,
    "socksproxy":<String_value>,
    "gopherproxy":<String_value>,
    "sslproxy":<String_value>,
    "proxyexception":<String_value>,
    "proxylocalbypass":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "clientcleanupprompt":<String_value>,
    "forcecleanup":<String[]_value>,
    "clientoptions":<String[]_value>,
    "clientconfiguration":<String[]_value>,
    "sso":<String_value>,
    "ssocredential":<String_value>,
    "windowsautologon":<String_value>,
    "usemip":<String_value>,
    "useiip":<String_value>,
    "clientdebug":<String_value>,
    "loginscript":<String_value>,
    "logoutscript":<String_value>,
    "homepage":<String_value>,
    "icaproxy":<String_value>,
    "wihome":<String_value>,
    "wihomeaddresstype":<String_value>,
    "citrixreceiverhome":<String_value>,
    "wiportalmode":<String_value>,
    "clientchoices":<String_value>,
    "epaclienttype":<String_value>,
    "iipdnssuffix":<String_value>,
    "forcedtimeout":<Double_value>,
    "forcedtimeoutwarning":<Double_value>,
    "ntdomain":<String_value>,
    "clientlessvpnmode":<String_value>,
    "clientlessmodeurlencoding":<String_value>,
    "clientlesspersistentcookie":<String_value>,
    "emailhome":<String_value>,
    "allowedlogingroups":<String_value>,
    "encryptcsecexp":<String_value>,
    "apptokentimeout":<Double_value>,
    "mdxtokentimeout":<Double_value>,
    "uitheme":<String_value>,
    "securebrowse":<String_value>,
    "storefronturl":<String_value>,
    "kcdaccount":<String_value>,
    "clientversions":<String_value>,
    "rdpclientprofilename":<String_value>,
    "windowspluginupgrade":<String_value>,
    "macpluginupgrade":<String_value>,
    "linuxpluginupgrade":<String_value>,
    "iconwithreceiver":<String_value>,
    "userdomains":<String_value>,
    "icasessiontimeout":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnparameter?**action=unset**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:


```
{"vpnparameter":{
    "httpport":true,
    "winsip":true,
    "dnsvservername":true,
    "splitdns":true,
    "sesstimeout":true,
    "clientsecurity":true,
    "clientsecuritygroup":true,
    "smartgroup":true,
    "clientsecuritymessage":true,
    "clientsecuritylog":true,
    "authorizationgroup":true,
    "clientidletimeout":true,
    "allprotocolproxy":true,
    "httpproxy":true,
    "ftpproxy":true,
    "socksproxy":true,
    "gopherproxy":true,
    "sslproxy":true,
    "proxyexception":true,
    "forcecleanup":true,
    "clientoptions":true,
    "clientconfiguration":true,
    "loginscript":true,
    "logoutscript":true,
    "homepage":true,
    "proxy":true,
    "wihome":true,
    "citrixreceiverhome":true,
    "wiportalmode":true,
    "iipdnssuffix":true,
    "forcedtimeout":true,
    "forcedtimeoutwarning":true,
    "defaultauthorizationaction":true,
    "ntdomain":true,
    "clientlessvpnmode":true,
    "emailhome":true,
    "clientlessmodeurlencoding":true,
    "clientlesspersistentcookie":true,
    "allowedlogingroups":true,
    "apptokentimeout":true,
    "mdxtokentimeout":true,
    "storefronturl":true,
    "uitheme":true,
    "kcdaccount":true,
    "rdpclientprofilename":true,
    "windowspluginupgrade":true,
    "macpluginupgrade":true,
    "linuxpluginupgrade":true,
    "iconwithreceiver":true,
    "userdomains":true,
    "icasessiontimeout":true,
    "splittunnel":true,
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "locallanaccess":true,
    "rfc1918":true,
    "spoofiip":true,
    "killconnections":true,
    "transparentinterception":true,
    "windowsclienttype":true,
    "proxylocalbypass":true,
    "clientcleanupprompt":true,
    "sso":true,
    "ssocredential":true,
    "windowsautologon":true,
    "usemip":true,
    "useiip":true,
    "clientdebug":true,
    "icaproxy":true,
    "clientchoices":true,
    "epaclienttype":true,
    "encryptcsecexp":true,
    "securebrowse":true,
    "clientversions":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnparameter

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnparameter": [ {
    "name":<String_value>,
    "httpport":<Integer[]_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

"winsip":<String_value>,
"dnsvservername":<String_value>,
"splitdns":<String_value>,
"sesstimeout":<Double_value>,
"clientsecurity":<String_value>,
"clientsecuritygroup":<String_value>,
"clientsecuritymessage":<String_value>,
"clientsecuritylog":<String_value>,
"smartgroup":<String_value>,
"splittunnel":<String_value>,
"locallanaccess":<String_value>,
"rfc1918":<String_value>,
"spoofiip":<String_value>,
"killconnections":<String_value>,
"transparentinterception":<String_value>,
"windowsclienttype":<String_value>,
"defaultauthorizationaction":<String_value>,
"authorizationgroup":<String_value>,
"clientidletimeout":<Double_value>,
"clientidletimeoutwarning":<Double_value>,
"proxy":<String_value>,
"allprotocolproxy":<String_value>,
"httpproxy":<String_value>,
"ftpproxy":<String_value>,
"socksproxy":<String_value>,
"gopherproxy":<String_value>,
"sslproxy":<String_value>,
"proxyexception":<String_value>,
"proxylocalbypass":<String_value>,
"clientcleanupprompt":<String_value>,
"forcecleanup":<String[]_value>,
"clientoptions":<String[]_value>,
"clientconfiguration":<String[]_value>,
"sso":<String_value>,
"ssocredential":<String_value>,
"windowsautologon":<String_value>,
"usemip":<String_value>,
"useiip":<String_value>,
"clientdebug":<String_value>,
"loginscript":<String_value>,
"logoutscript":<String_value>,
"homepage":<String_value>,
"icaproxy":<String_value>,
"wihome":<String_value>,
"wihomeaddresstype":<String_value>,
"citrixreceiverhome":<String_value>,
"wiportalmode":<String_value>,
"clientchoices":<String_value>,
"epaclienttype":<String_value>,
"iipdnssuffix":<String_value>,
"forcedtimeout":<Double_value>,
"forcedtimeoutwarning":<Double_value>,
"ntdomain":<String_value>,
"clientlessvpnmode":<String_value>,
"clientlessmodeurlencoding":<String_value>,
"clientlesspersistentcookie":<String_value>,
"emailhome":<String_value>,
"allowedlogingroups":<String_value>,
"encryptcsecexp":<String_value>,
"apptokentimeout":<Double_value>,
"mdxtokentimeout":<Double_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "uitheme":<String_value>,
    "securebrowse":<String_value>,
    "storefronturl":<String_value>,
    "kcdaccount":<String_value>,
    "clientversions":<String_value>,
    "rdpclientprofilename":<String_value>,
    "windowspluginupgrade":<String_value>,
    "macpluginupgrade":<String_value>,
    "linuxpluginupgrade":<String_value>,
    "iconwithreceiver":<String_value>,
    "userdomains":<String_value>,
    "icasessiontimeout":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnicaconnection

May 20, 2016

Configuration for active ica connections resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
username	<String>	Read-write	User name for which to display connections. Minimum length = 1
all	<Boolean>	Read-write	Terminate all active icaconnections.
domain	<String>	Read-only	The domain name.
srcip	<String>	Read-only	The client IP address.
srcport	<Integer>	Read-only	The client port. Range 1 - 65535
destip	<String>	Read-only	The CPS server IP address.
destport	<Integer>	Read-only	The CPS server port. Range 1 - 65535
peid	<Double>	Read-only	Core id of the session owner.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

KILL | GET (ALL) | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

kill

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnicaconnection?**action=kill**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnicaconnection":{
    "username":<String_value>,
    "all":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnicaconnection

Query-parameters:

args

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/vpnicaconnection?**args=username:<String_value>**

Use this query-parameter to get vpnicaconnection resources based on additional properties.

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnicaconnection?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnicaconnection?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnicaconnection resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnicaconnection?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnicaconnection?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnicaconnection resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnicaconnection": [ {
username:<String_value>      "domain":<String_value>,
     "srcip":<String_value>,
     "srcport":<Integer_value>,
     "destip":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "destport":<Integer_value>,
    "peid":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnicaconnection?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnicaconnection": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnintranetapplication

May 20, 2016

Configuration for SSLVPN intranet application resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
intranetapplication	<String>	Read-write	Name of the intranet application. Minimum length = 1 Maximum length = 31
protocol	<String>	Read-write	Protocol used by the intranet application. If protocol is set to BOTH, TCP and UDP traffic is allowed. Possible values = TCP, UDP, ANY
destip	<String>	Read-write	Destination IP address, IP range, or host name of the intranet application. This address is the server IP address. Minimum length = 1
netmask	<String>	Read-write	Destination subnet mask for the intranet application.
iprange	<String>	Read-write	If you have multiple servers in your network, such as web, email, and file shares, configure an intranet application that includes the IP range for all the network applications. This allows users to access all the intranet applications contained in the IP address range. Minimum length = 1
hostname	<String>	Read-write	Name of the host for which to configure interception. The names are resolved during interception when users log on with the NetScaler Gateway Plug-in. Minimum length = 1
clientapplication	<String[]>	Read-write	Names of the client applications, such as PuTTY and Xshell. Minimum length = 1
spoofiip	<String>	Read-write	IP address that the intranet application will use to route the connection through the virtual adapter. Default value: ON Possible values = ON, OFF
destport	<String>	Read-write	Destination TCP or UDP port number for the intranet application. Use a hyphen to specify a range of port numbers, for example 90-95. Minimum length = 1
			Interception mode for the intranet application or resource. Correct value depends on the type of client software used to make connections. If the interception mode is

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Windows. With the PROXY setting, users connect with the NetScaler Gateway Plug-in for Java. Possible values = PROXY, TRANSPARENT
srcip	<String>	Read-write	Source IP address. Required if interception mode is set to PROXY. Default is the loopback address, 127.0.0.1. Minimum length = 1
srcport	<Integer>	Read-write	Source port for the application for which the NetScaler Gateway virtual server proxies the traffic. If users are connecting from a device that uses the NetScaler Gateway Plug-in for Java, applications must be configured manually by using the source IP address and TCP port values specified in the intranet application profile. If a port value is not set, the destination port value is used. Minimum value = 1
ipaddress	<String>	Read-only	The IP address for the application. This address is the real application server IP address.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnintranetapplication":{
    "intranetapplication":<String_value>,
    "protocol":<String_value>,
    "destip":<String_value>,
    "netmask":<String_value>,
    "iprange":<String_value>,
    "hostname":<String_value>,
    "clientapplication":<String[]_value>,
    "spoofiip":<String_value>,
    "destport":<String_value>,
    "interception":<String_value>,
    "srcip":<String_value>,
    "srcport":<Integer_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication/intranetapplication_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnintranetapplication resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnintranetapplication resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnintranetapplication": [ {
    "intranetapplication":<String_value>,
    "protocol":<String_value>,
    "destip":<String_value>,
    "netmask":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ipaddress":<String_value>,
    "hostname":<String_value>,
    "destport":<String_value>,
    "clientapplication":<String[]_value>,
    "spoofiip":<String_value>,
    "interception":<String_value>,
    "srcip":<String_value>,
    "srcport":<Integer_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication/intranetapplication_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication/intranetapplication_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication/intranetapplication_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnintranetapplication": [ {
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "intranetapplication":<String_value>,
    "protocol":<String_value>,
    "destip":<String_value>,
    "netmask":<String_value>,
    "ipaddress":<String_value>,
    "hostname":<String_value>,
    "destport":<String_value>,
    "clientapplication":<String[]_value>,
    "spoofiip":<String_value>,
    "interception":<String_value>,
    "srcip":<String_value>,
    "srcport":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnintranetapplication?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnintranetapplication": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnnexthopserver

May 20, 2016

Configuration for Next Hop Server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the NetScaler Gateway appliance in the first DMZ. Minimum length = 1 Maximum length = 32
nexthopip	<String>	Read-write	IP address of the NetScaler Gateway proxy in the second DMZ.
nexthopfqdn	<String>	Read-write	FQDN of the NetScaler Gateway proxy in the second DMZ. Minimum length = 1
resaddresstype	<String>	Read-write	Address Type (IPV4/IPv6) of DNS name of nextHopServer FQDN. Minimum length = 1 Possible values = IPV4, IPV6
nexthopport	<Integer>	Read-write	Port number of the NetScaler Gateway proxy in the second DMZ. Minimum value = 1 Maximum value = 65535
secure	<String>	Read-write	Use of a secure port, such as 443, for the double-hop configuration. Default value: OFF Possible values = ON, OFF
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnnexthopserver":{
    "name":<String_value>,
    "nexthopip":<String_value>,
    "nexthopfqdn":<String_value>,
    "resaddresstype":<String_value>,
    "nexthopport":<Integer_value>,
    "secure":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnnexthopserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnnexthopserver resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnnexthopserver": [ {
      "name":<String_value>,
      "nexthopip":<String_value>,
      "nexthopfqdn":<String_value>,
      "resaddresstype":<String_value>,
      "nexthopport":<Integer_value>,
      "secure":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnnexthopserver": [ {
    "name":<String_value>,
    "nexthopip":<String_value>,
    "nexthopfqdn":<String_value>,
    "resaddresstype":<String_value>,
    "nexthopport":<Integer_value>,
    "secure":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnnexthopserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnnexthopserver": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnportaltheme

May 20, 2016

Configuration for portaltheme resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the uitheme. Minimum length = 1
basetheme	<String>	Read-write	. Minimum length = 1 Possible values = Default, Greenbubble, X1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnportaltheme":{
    "name":<String_value>,
    "basetheme":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnportaltheme resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnportaltheme resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnportaltheme": [ {
      "name":<String_value>,
      "basetheme":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnportaltheme": [ {
      "name":<String_value>,
      "basetheme":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnportaltheme?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnportaltheme": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnsamlssoprofile

May 20, 2016

Configuration for SAML sso action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the new saml single sign-on profile. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after an SSO action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
samlsigningcertname	<String>	Read-write	Name of the signing authority as given in the SAML server's SSL certificate. Minimum length = 1
assertionconsumerserviceurl	<String>	Read-write	URL to which the assertion is to be sent. Minimum length = 1
relaystaterule	<String>	Read-write	Expression to extract relaystate to be sent along with assertion. Evaluation of this expression should return TEXT content. This is typically a target url to which user is redirected after the recipient validates SAML token.
sendpassword	<String>	Read-write	Option to send password in assertion. Default value: OFF Possible values = ON, OFF
samlissuername	<String>	Read-write	The name to be used in requests sent from Netscaler to IdP to uniquely identify Netscaler. Minimum length = 1
signaturealg	<String>	Read-write	Algorithm to be used to sign/verify SAML transactions. Default value: RSA-SHA1 Possible values = RSA-SHA1, RSA-SHA256
digestmethod	<String>	Read-write	Algorithm to be used to compute/verify digest for SAML transactions. Default value: SHA1 Possible values = SHA1, SHA256

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			entity name or uri that represents ServiceProvider.
nameidformat	<String>	Read-write	Format of Name Identifier sent in Assertion. Default value: transient Possible values = Unspecified, emailAddress, X509SubjectName, WindowsDomainQualifiedName, kerberos, entity, persistent, transient
nameidexpr	<String>	Read-write	Expression that will be evaluated to obtain NameIdentifier to be sent in assertion. Maximum length = 128
attribute1	<String>	Read-write	Name of attribute1 that needs to be sent in SAML Assertion.
attribute1expr	<String>	Read-write	Expression that will be evaluated to obtain attribute1's value to be sent in Assertion. Maximum length = 128
attribute1friendlyname	<String>	Read-write	User-Friendly Name of attribute1 that needs to be sent in SAML Assertion.
attribute1format	<String>	Read-write	Format of Attribute1 to be sent in Assertion. Possible values = URI, Basic
attribute2	<String>	Read-write	Name of attribute2 that needs to be sent in SAML Assertion.
attribute2expr	<String>	Read-write	Expression that will be evaluated to obtain attribute2's value to be sent in Assertion. Maximum length = 128
attribute2friendlyname	<String>	Read-write	User-Friendly Name of attribute2 that needs to be sent in SAML Assertion.
attribute2format	<String>	Read-write	Format of Attribute2 to be sent in Assertion. Possible values = URI, Basic
attribute3	<String>	Read-write	Name of attribute3 that needs to be sent in SAML Assertion.
attribute3expr	<String>	Read-write	Expression that will be evaluated to obtain attribute3's value to be sent in Assertion. Maximum length = 128
attribute3friendlyname	<String>	Read-write	User-Friendly Name of attribute3 that needs to be sent in SAML Assertion.
attribute3format	<String>	Read-write	Format of Attribute3 to be sent in Assertion. Possible values = URI, Basic
attribute4	<String>	Read-write	Name of attribute4 that needs to be sent in SAML Assertion.
			Expression that will be evaluated to obtain attribute4's value to be sent in

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum length = 128
attribute4friendlyname	<String>	Read-write	User-Friendly Name of attribute4 that needs to be sent in SAML Assertion.
attribute4format	<String>	Read-write	Format of Attribute4 to be sent in Assertion. Possible values = URI, Basic
attribute5	<String>	Read-write	Name of attribute5 that needs to be sent in SAML Assertion.
attribute5expr	<String>	Read-write	Expression that will be evaluated to obtain attribute5's value to be sent in Assertion. Maximum length = 128
attribute5friendlyname	<String>	Read-write	User-Friendly Name of attribute5 that needs to be sent in SAML Assertion.
attribute5format	<String>	Read-write	Format of Attribute5 to be sent in Assertion. Possible values = URI, Basic
attribute6	<String>	Read-write	Name of attribute6 that needs to be sent in SAML Assertion.
attribute6expr	<String>	Read-write	Expression that will be evaluated to obtain attribute6's value to be sent in Assertion. Maximum length = 128
attribute6friendlyname	<String>	Read-write	User-Friendly Name of attribute6 that needs to be sent in SAML Assertion.
attribute6format	<String>	Read-write	Format of Attribute6 to be sent in Assertion. Possible values = URI, Basic
attribute7	<String>	Read-write	Name of attribute7 that needs to be sent in SAML Assertion.
attribute7expr	<String>	Read-write	Expression that will be evaluated to obtain attribute7's value to be sent in Assertion. Maximum length = 128
attribute7friendlyname	<String>	Read-write	User-Friendly Name of attribute7 that needs to be sent in SAML Assertion.
attribute7format	<String>	Read-write	Format of Attribute7 to be sent in Assertion. Possible values = URI, Basic
attribute8	<String>	Read-write	Name of attribute8 that needs to be sent in SAML Assertion.
attribute8expr	<String>	Read-write	Expression that will be evaluated to obtain attribute8's value to be sent in Assertion. Maximum length = 128

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
attribute8format	<String>	Read-write	Format of Attribute8 to be sent in Assertion. Possible values = URI, Basic
attribute9	<String>	Read-write	Name of attribute9 that needs to be sent in SAML Assertion.
attribute9expr	<String>	Read-write	Expression that will be evaluated to obtain attribute9's value to be sent in Assertion. Maximum length = 128
attribute9friendlyname	<String>	Read-write	User-Friendly Name of attribute9 that needs to be sent in SAML Assertion.
attribute9format	<String>	Read-write	Format of Attribute9 to be sent in Assertion. Possible values = URI, Basic
attribute10	<String>	Read-write	Name of attribute10 that needs to be sent in SAML Assertion.
attribute10expr	<String>	Read-write	Expression that will be evaluated to obtain attribute10's value to be sent in Assertion. Maximum length = 128
attribute10friendlyname	<String>	Read-write	User-Friendly Name of attribute10 that needs to be sent in SAML Assertion.
attribute10format	<String>	Read-write	Format of Attribute10 to be sent in Assertion. Possible values = URI, Basic
attribute11	<String>	Read-write	Name of attribute11 that needs to be sent in SAML Assertion.
attribute11expr	<String>	Read-write	Expression that will be evaluated to obtain attribute11's value to be sent in Assertion. Maximum length = 128
attribute11friendlyname	<String>	Read-write	User-Friendly Name of attribute11 that needs to be sent in SAML Assertion.
attribute11format	<String>	Read-write	Format of Attribute11 to be sent in Assertion. Possible values = URI, Basic
attribute12	<String>	Read-write	Name of attribute12 that needs to be sent in SAML Assertion.
attribute12expr	<String>	Read-write	Expression that will be evaluated to obtain attribute12's value to be sent in Assertion. Maximum length = 128
attribute12friendlyname	<String>	Read-write	User-Friendly Name of attribute12 that needs to be sent in SAML Assertion.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = URI, Basic
attribute13	<String>	Read-write	Name of attribute13 that needs to be sent in SAML Assertion.
attribute13expr	<String>	Read-write	Expression that will be evaluated to obtain attribute13's value to be sent in Assertion. Maximum length = 128
attribute13friendlyname	<String>	Read-write	User-Friendly Name of attribute13 that needs to be sent in SAML Assertion.
attribute13format	<String>	Read-write	Format of Attribute13 to be sent in Assertion. Possible values = URI, Basic
attribute14	<String>	Read-write	Name of attribute14 that needs to be sent in SAML Assertion.
attribute14expr	<String>	Read-write	Expression that will be evaluated to obtain attribute14's value to be sent in Assertion. Maximum length = 128
attribute14friendlyname	<String>	Read-write	User-Friendly Name of attribute14 that needs to be sent in SAML Assertion.
attribute14format	<String>	Read-write	Format of Attribute14 to be sent in Assertion. Possible values = URI, Basic
attribute15	<String>	Read-write	Name of attribute15 that needs to be sent in SAML Assertion.
attribute15expr	<String>	Read-write	Expression that will be evaluated to obtain attribute15's value to be sent in Assertion. Maximum length = 128
attribute15friendlyname	<String>	Read-write	User-Friendly Name of attribute15 that needs to be sent in SAML Assertion.
attribute15format	<String>	Read-write	Format of Attribute15 to be sent in Assertion. Possible values = URI, Basic
attribute16	<String>	Read-write	Name of attribute16 that needs to be sent in SAML Assertion.
attribute16expr	<String>	Read-write	Expression that will be evaluated to obtain attribute16's value to be sent in Assertion. Maximum length = 128
attribute16friendlyname	<String>	Read-write	User-Friendly Name of attribute16 that needs to be sent in SAML Assertion.
attribute16format	<String>	Read-write	Format of Attribute16 to be sent in Assertion. Possible values = URI, Basic

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
encryptassertion	<String>	Read-write	Default value: OFF Possible values = ON, OFF
samlspcertname	<String>	Read-write	Name of the SSL certificate of peer/receving party using which Assertion is encrypted. Minimum length = 1
encryptionalgorithm	<String>	Read-write	Algorithm to be used to encrypt SAML assertion. Default value: AES256 Possible values = DES3, AES128, AES192, AES256
skewtime	<Double>	Read-write	This option specifies the number of minutes on either side of current time that the assertion would be valid. For example, if skewTime is 10, then assertion would be valid from (current time - 10) min to (current time + 10) min, ie 20min in all. Default value: 5
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnsamlssoprofile":{
    "name":<String_value>,
    "samlsigningcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "relaystaterule":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute1expr":<String_value>,
    "attribute1friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2":<String_value>,
    "attribute2expr":<String_value>,
    "attribute2friendlyname":<String_value>,
    "attribute2format":<String_value>,
    "attribute3":<String_value>,
    "attribute3expr":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute3format":<String_value>,
    "attribute4":<String_value>,
    "attribute4expr":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute4format":<String_value>,
    "attribute5":<String_value>,
    "attribute5expr":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute5format":<String_value>,
    "attribute6":<String_value>,
    "attribute6expr":<String_value>,
    "attribute6friendlyname":<String_value>,
    "attribute6format":<String_value>,
    "attribute7":<String_value>,
    "attribute7expr":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute7format":<String_value>,
    "attribute8":<String_value>,
    "attribute8expr":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute8format":<String_value>,
    "attribute9":<String_value>,
    "attribute9expr":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute9friendlyname":<String_value>,
    "attribute9format":<String_value>,
    "attribute10":<String_value>,
    "attribute10expr":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute10format":<String_value>,
    "attribute11":<String_value>,
    "attribute11expr":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute11format":<String_value>,
    "attribute12":<String_value>,
    "attribute12expr":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute12format":<String_value>,
    "attribute13":<String_value>,
    "attribute13expr":<String_value>,
    "attribute13friendlyname":<String_value>,
    "attribute13format":<String_value>,
    "attribute14":<String_value>,
    "attribute14expr":<String_value>,
    "attribute14friendlyname":<String_value>,
    "attribute14format":<String_value>,
    "attribute15":<String_value>,
    "attribute15expr":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute15format":<String_value>,
    "attribute16":<String_value>,
    "attribute16expr":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute16format":<String_value>,
    "encryptassertion":<String_value>,
    "samlspcertname":<String_value>,
    "encryptionalgorithm":<String_value>,
    "skewtime":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnsamlssoprofile":{
    "name":<String_value>,
    "samlsigningcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "relaystaterule":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute1expr":<String_value>,
    "attribute1friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2":<String_value>,
    "attribute2expr":<String_value>,
    "attribute2friendlyname":<String_value>,
    "attribute2format":<String_value>,
    "attribute3":<String_value>,
    "attribute3expr":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute3format":<String_value>,
    "attribute4":<String_value>,
    "attribute4expr":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute4format":<String_value>,
    "attribute5":<String_value>,
    "attribute5expr":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute5format":<String_value>,
    "attribute6":<String_value>,
    "attribute6expr":<String_value>,
    "attribute6friendlyname":<String_value>,
    "attribute6format":<String_value>,
    "attribute7":<String_value>,
    "attribute7expr":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute7format":<String_value>,
    "attribute8":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute8expr":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute8format":<String_value>,
    "attribute9":<String_value>,
    "attribute9expr":<String_value>,
    "attribute9friendlyname":<String_value>,
    "attribute9format":<String_value>,
    "attribute10":<String_value>,
    "attribute10expr":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute10format":<String_value>,
    "attribute11":<String_value>,
    "attribute11expr":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute11format":<String_value>,
    "attribute12":<String_value>,
    "attribute12expr":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute12format":<String_value>,
    "attribute13":<String_value>,
    "attribute13expr":<String_value>,
    "attribute13friendlyname":<String_value>,
    "attribute13format":<String_value>,
    "attribute14":<String_value>,
    "attribute14expr":<String_value>,
    "attribute14friendlyname":<String_value>,
    "attribute14format":<String_value>,
    "attribute15":<String_value>,
    "attribute15expr":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute15format":<String_value>,
    "attribute16":<String_value>,
    "attribute16expr":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute16format":<String_value>,
    "encryptassertion":<String_value>,
    "samlspcertname":<String_value>,
    "encryptionalgorithm":<String_value>,
    "skewtime":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{"vpnsamlssoprofile":{
    "name":<String_value>,
    "samlsigningcertname":true,
    "sendpassword":true,
    "samlissuername":true,
    "signaturealg":true,
    "digestmethod":true,
    "audience":true,
    "nameidformat":true,
    "nameidexpr":true,
    "attribute1":true,
    "attribute1friendlyname":true,
    "attribute1format":true,
    "attribute2":true,
    "attribute2friendlyname":true,
    "attribute2format":true,
    "attribute3":true,
    "attribute3friendlyname":true,
    "attribute3format":true,
    "attribute4":true,
    "attribute4friendlyname":true,
    "attribute4format":true,
    "attribute5":true,
    "attribute5friendlyname":true,
    "attribute5format":true,
    "attribute6":true,
    "attribute6friendlyname":true,
    "attribute6format":true,
    "attribute7":true,
    "attribute7friendlyname":true,
    "attribute7format":true,
    "attribute8":true,
    "attribute8friendlyname":true,
    "attribute8format":true,
    "attribute9":true,
    "attribute9friendlyname":true,
    "attribute9format":true,
    "attribute10":true,
    "attribute10friendlyname":true,
    "attribute10format":true,
    "attribute11":true,
    "attribute11friendlyname":true,
    "attribute11format":true,
    "attribute12":true,
    "attribute12friendlyname":true,
    "attribute12format":true,
    "attribute13":true,
    "attribute13friendlyname":true,
    "attribute13format":true,
    "attribute14":true,
    "attribute14friendlyname":true,
    "attribute14format":true,
    "attribute15":true,
    "attribute15friendlyname":true,
    "attribute15format":true,
    "attribute16":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "attribute16friendlyname":true,
        "attribute16format":true,
        "encryptassertion":true,
        "samlspcertname":true,
        "encryptionalgorithm":true,
        "skewtime":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnsamlssoprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnsamlssoprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsamlssoprofile": [ {
    "name":<String_value>,
    "samlsigningcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "relaystaterule":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>,
    "attribute1friendlyname":<String_value>,
    "attribute2friendlyname":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute6friendlyname":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute9friendlyname":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute13friendlyname":<String_value>,
    "attribute14friendlyname":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2format":<String_value>,
    "attribute3format":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute4format":<String_value>,
    "attribute5format":<String_value>,
    "attribute6format":<String_value>,
    "attribute7format":<String_value>,
    "attribute8format":<String_value>,
    "attribute9format":<String_value>,
    "attribute10format":<String_value>,
    "attribute11format":<String_value>,
    "attribute12format":<String_value>,
    "attribute13format":<String_value>,
    "attribute14format":<String_value>,
    "attribute15format":<String_value>,
    "attribute16format":<String_value>,
    "attribute1expr":<String_value>,
    "attribute2expr":<String_value>,
    "attribute3expr":<String_value>,
    "attribute4expr":<String_value>,
    "attribute5expr":<String_value>,
    "attribute6expr":<String_value>,
    "attribute7expr":<String_value>,
    "attribute8expr":<String_value>,
    "attribute9expr":<String_value>,
    "attribute10expr":<String_value>,
    "attribute11expr":<String_value>,
    "attribute12expr":<String_value>,
    "attribute13expr":<String_value>,
    "attribute14expr":<String_value>,
    "attribute15expr":<String_value>,
    "attribute16expr":<String_value>,
    "encryptassertion":<String_value>,
    "samlspcertname":<String_value>,
    "encryptionalgorithm":<String_value>,
    "skewtime":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsamlssoprofile": [ {
    "name":<String_value>,
    "samlsigningcertname":<String_value>,
    "assertionconsumerserviceurl":<String_value>,
    "sendpassword":<String_value>,
    "samlissuername":<String_value>,
    "relaystaterule":<String_value>,
    "signaturealg":<String_value>,
    "digestmethod":<String_value>,
    "audience":<String_value>,
    "nameidformat":<String_value>,
    "nameidexpr":<String_value>,
    "attribute1":<String_value>,
    "attribute2":<String_value>,
    "attribute3":<String_value>,
    "attribute4":<String_value>,
    "attribute5":<String_value>,
    "attribute6":<String_value>,
    "attribute7":<String_value>,
    "attribute8":<String_value>,
    "attribute9":<String_value>,
    "attribute10":<String_value>,
    "attribute11":<String_value>,
    "attribute12":<String_value>,
    "attribute13":<String_value>,
    "attribute14":<String_value>,
    "attribute15":<String_value>,
    "attribute16":<String_value>,
    "attribute1friendlyname":<String_value>,
    "attribute2friendlyname":<String_value>,
    "attribute3friendlyname":<String_value>,
    "attribute4friendlyname":<String_value>,
    "attribute5friendlyname":<String_value>,
    "attribute6friendlyname":<String_value>,
    "attribute7friendlyname":<String_value>,
    "attribute8friendlyname":<String_value>,
    "attribute9friendlyname":<String_value>,
    "attribute10friendlyname":<String_value>,
    "attribute11friendlyname":<String_value>,
    "attribute12friendlyname":<String_value>,
    "attribute13friendlyname":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "attribute14friendlyname":<String_value>,
    "attribute15friendlyname":<String_value>,
    "attribute16friendlyname":<String_value>,
    "attribute1format":<String_value>,
    "attribute2format":<String_value>,
    "attribute3format":<String_value>,
    "attribute4format":<String_value>,
    "attribute5format":<String_value>,
    "attribute6format":<String_value>,
    "attribute7format":<String_value>,
    "attribute8format":<String_value>,
    "attribute9format":<String_value>,
    "attribute10format":<String_value>,
    "attribute11format":<String_value>,
    "attribute12format":<String_value>,
    "attribute13format":<String_value>,
    "attribute14format":<String_value>,
    "attribute15format":<String_value>,
    "attribute16format":<String_value>,
    "attribute1expr":<String_value>,
    "attribute2expr":<String_value>,
    "attribute3expr":<String_value>,
    "attribute4expr":<String_value>,
    "attribute5expr":<String_value>,
    "attribute6expr":<String_value>,
    "attribute7expr":<String_value>,
    "attribute8expr":<String_value>,
    "attribute9expr":<String_value>,
    "attribute10expr":<String_value>,
    "attribute11expr":<String_value>,
    "attribute12expr":<String_value>,
    "attribute13expr":<String_value>,
    "attribute14expr":<String_value>,
    "attribute15expr":<String_value>,
    "attribute16expr":<String_value>,
    "encryptassertion":<String_value>,
    "samlspcertname":<String_value>,
    "encryptionalgorithm":<String_value>,
    "skewtime":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsamlssoprofile?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnsamlssoprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnsessionaction

May 20, 2016

Configuration for VPN session action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the NetScaler Gateway profile (action). Must begin with an ASCII alphabetic or underscore (_) character, and must consist only of ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the profile is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
useraccounting	<String>	Read-write	The name of the radiusPolicy to use for RADIUS user accounting info on the session.
httpport	<Integer[]>	Read-write	Destination port numbers other than port 80, added as a comma-separated list. Traffic to these ports is processed as HTTP traffic, which allows functionality, such as HTTP authorization and single sign-on to a web application to work. Minimum value = 1
winsip	<String>	Read-write	WINS server IP address to add to NetScaler Gateway for name resolution.
dnsvservername	<String>	Read-write	Name of the DNS virtual server for the user session. Minimum length = 1
splitdns	<String>	Read-write	Route the DNS requests to the local DNS server configured on the user device, or NetScaler Gateway (remote), or both. Possible values = LOCAL, REMOTE, BOTH
sesstimeout	<Double>	Read-write	Number of minutes after which the session times out. Minimum value = 1
clientsecurity	<String>	Read-write	Specify the client security check for the user device to permit a NetScaler Gateway session. The web address or IP address is not included in the expression for the client security check.
			The client security group that will be assigned on failure of the client security check. Users can in general be organized into Groups. In this case,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Minimum length = 1
clientsecuritymessage	<String>	Read-write	The client security message that will be displayed on failure of the client security check. Minimum length = 1 Maximum length = 127
clientsecuritylog	<String>	Read-write	Set the logging of client security checks. Possible values = ON, OFF
splittunnel	<String>	Read-write	Send, through the tunnel, traffic only for intranet applications that are defined in NetScaler Gateway. Route all other traffic directly to the Internet. The OFF setting routes all traffic through NetScaler Gateway. With the REVERSE setting, intranet applications define the network traffic that is not intercepted. All network traffic directed to internal IP addresses bypasses the VPN tunnel, while other traffic goes through NetScaler Gateway. Reverse split tunneling can be used to log all non-local LAN traffic. For example, if users have a home network and are logged on through the NetScaler Gateway Plug-in, network traffic destined to a printer or another device within the home network is not intercepted. Possible values = ON, OFF, REVERSE
locallanaccess	<String>	Read-write	Set local LAN access. If split tunneling is OFF, and you set local LAN access to ON, the local client can route traffic to its local interface. When the local area network switch is specified, this combination of switches is useful. The client can allow local LAN access to devices that commonly have non-routable addresses, such as local printers or local file servers. Possible values = ON, OFF
rfc1918	<String>	Read-write	As defined in the local area network, allow only the following local area network addresses to bypass the VPN tunnel when the local LAN access feature is enabled: * 10.*.*.*, * 172.16.*.*, * 192.168.*.*. Possible values = ON, OFF
spoofiip	<String>	Read-write	IP address that the intranet application uses to route the connection through the virtual adapter. Possible values = ON, OFF
killconnections	<String>	Read-write	Specify whether the NetScaler Gateway Plug-in should disconnect all preexisting connections, such as the connections existing before the user logged on to NetScaler Gateway, and prevent new incoming connections on the NetScaler Gateway Plug-in for Windows and MAC when the user is connected to NetScaler Gateway and split tunneling is disabled. Possible values = ON, OFF
transparentinterception	<String>	Read-write	Allow access to network resources by using a single IP address and subnet mask or a range of IP addresses. The OFF setting sets the mode to proxy, in which you configure destination and source IP addresses and port numbers. If you are using the NetScaler Gateway Plug-in for Windows, set this parameter to ON, in which the mode is set to transparent. If you are using the NetScaler Gateway Plug-in for Java, set this parameter to OFF.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
windowsclienttype	<String>	Read-write	Choose between two types of Windows Client\ a) Application Agent - which always runs in the task bar as a standalone application and also has a supporting service which runs permanently when installed\ b) Activex Control - ActiveX control run by Microsoft Internet Explorer. Possible values = AGENT, PLUGIN
defaultauthorizationaction	<String>	Read-write	Specify the network resources that users have access to when they log on to the internal network. The default setting for authorization is to deny access to all network resources. Citrix recommends using the default global setting and then creating authorization policies to define the network resources users can access. If you set the default authorization policy to DENY, you must explicitly authorize access to any network resource, which improves security. Possible values = ALLOW, DENY
authorizationgroup	<String>	Read-write	Comma-separated list of groups in which the user is placed when none of the groups that the user is a part of is configured on NetScaler Gateway. The authorization policy can be bound to these groups to control access to the resources. Minimum length = 1
smartgroup	<String>	Read-write	This is the group in which the user is placed when the sessionpolicy associated with this sessionaction succeeds. The vpn session policy will do the post auth EPA check and if the check succeeds the user is placed in the group specified with smartgroup. The is_member_of (http.req.user.is_member_of) expression can then be used with policies to check if EPA has passed on the user belonging to this smartgroup. Minimum length = 1 Maximum length = 64
clientidletimeout	<Double>	Read-write	Time, in minutes, after which to time out the user session if NetScaler Gateway does not detect mouse or keyboard activity. Minimum value = 1 Maximum value = 9999
proxy	<String>	Read-write	Set options to apply proxy for accessing the internal resources. Available settings function as follows: * BROWSER - Proxy settings are configured only in Internet Explorer and Firefox browsers. * NS - Proxy settings are configured on the NetScaler appliance. * OFF - Proxy settings are not configured. Possible values = BROWSER, NS, OFF
allprotocolproxy	<String>	Read-write	IP address of the proxy server to use for all protocols supported by NetScaler Gateway. Minimum length = 1
httpproxy	<String>	Read-write	IP address of the proxy server to be used for HTTP access for all subsequent connections to the internal network. Minimum length = 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			connections to the internal network. Minimum length = 1
socksproxy	<String>	Read-write	IP address of the proxy server to be used for SOCKS access for all subsequent connections to the internal network. Minimum length = 1
gopherproxy	<String>	Read-write	IP address of the proxy server to be used for GOPHER access for all subsequent connections to the internal network. Minimum length = 1
sslproxy	<String>	Read-write	IP address of the proxy server to be used for SSL access for all subsequent connections to the internal network. Minimum length = 1
proxyexception	<String>	Read-write	Proxy exception string that will be configured in the browser for bypassing the previously configured proxies. Allowed only if proxy type is Browser. Minimum length = 1
proxylocalbypass	<String>	Read-write	Bypass proxy server for local addresses option in Internet Explorer and Firefox proxy server settings. Possible values = ENABLED, DISABLED
clientcleanupprompt	<String>	Read-write	Prompt for client-side cache clean-up when a client-initiated session closes. Possible values = ON, OFF
forcecleanup	<String[]>	Read-write	Force cache clean-up when the user closes a session. You can specify all, none, or any combination of the client-side items. Possible values = none, all, cookie, addressbar, plugin, filesystemapplication, application, applicationdata, clientcertificate, autocomplete, cache
clientoptions	<String[]>	Read-write	Display only the configured menu options when you select the "Configure NetScaler Gateway" option in the NetScaler Gateway Plug-in system tray icon for Windows. Possible values = none, all, services, filetransfer, configuration
clientconfiguration	<String[]>	Read-write	Display only the configured tabs when you select the "Configure NetSCaler Gateway" option in the NetScaler Gateway Plug-in system tray icon for Windows. Possible values = none, all, general, tunnel, trace, compression
sso	<String>	Read-write	Set single sign-on (SSO) for the session. When the user accesses a server, the user's logon credentials are passed to the server for authentication. Possible values = ON, OFF
ssocredential	<String>	Read-write	Specify whether to use the primary or secondary authentication credentials for single sign-on to the server. Possible values = PRIMARY, SECONDARY
windowsautologon	<String>	Read-write	Enable or disable the Windows Auto Logon for the session. If a VPN session is established after this setting is enabled, the user is automatically

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = ON, OFF
usemip	<String>	Read-write	Enable or disable the use of a unique IP address alias, or a mapped IP address, as the client IP address for each client session. Allow NetScaler Gateway to use the mapped IP address as an intranet IP address when all other IP addresses are not available. When IP pooling is configured and the mapped IP is used as an intranet IP address, the mapped IP address is used when an intranet IP address cannot be assigned. Possible values = NS, OFF
useiip	<String>	Read-write	Define IP address pool options. Available settings function as follows: * SPILLOVER - When an address pool is configured and the mapped IP is used as an intranet IP address, the mapped IP address is used when an intranet IP address cannot be assigned. * NOSPILLOVER - When intranet IP addresses are enabled and the mapped IP address is not used, the Transfer Login page appears for users who have used all available intranet IP addresses. * OFF - Address pool is not configured. Possible values = NOSPILLOVER, SPILLOVER, OFF
clientdebug	<String>	Read-write	Set the trace level on NetScaler Gateway. Technical support technicians use these debug logs for in-depth debugging and troubleshooting purposes. Available settings function as follows: * DEBUG - Detailed debug messages are collected and written into the specified file. * STATS - Application audit level error messages and debug statistic counters are written into the specified file. * EVENTS - Application audit-level error messages are written into the specified file. * OFF - Only critical events are logged into the Windows Application Log. Possible values = debug, stats, events, OFF
loginscript	<String>	Read-write	Path to the logon script that is run when a session is established. Separate multiple scripts by using comma. A "$" in the path signifies that the word following the "$" is an environment variable. Minimum length = 1
logoutscript	<String>	Read-write	Path to the logout script. Separate multiple scripts by using comma. A "$" in the path signifies that the word following the "$" is an environment variable. Minimum length = 1
homepage	<String>	Read-write	Web address of the home page that appears when users log on. Otherwise, users receive the default home page for NetScaler Gateway, which is the Access Interface.
icaproxy	<String>	Read-write	Enable ICA proxy to configure secure Internet access to servers running Citrix XenApp or XenDesktop by using Citrix Receiver instead of the NetScaler Gateway Plug-in. Possible values = ON, OFF
			Web address of the Web Interface server, such as http://<ipAddress>/Citrix/XenApp, or Receiver for Web, which enumerates the virtualized resources, such as XenApp, XenDesktop, and cloud

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
wihome	<String>	Read-write	mode. If Client Choices is ON, you must configure this setting. Because the user can choose between FullClient and ICAProxy, the user may see a different home page. An Internet web site may appear if the user gets the FullClient option, or a Web Interface site if the user gets the ICAProxy option. If the setting is not configured, the XenApp option does not appear as a client choice.
wihomeaddresstype	<String>	Read-write	Type of the wihome address(IPV4/V6). Possible values = IPV4, IPV6
citrixreceiverhome	<String>	Read-write	Web address for the Citrix Receiver home page. Configure NetScaler Gateway so that when users log on to the appliance, the NetScaler Gateway Plug-in opens a web browser that allows single sign-on to the Citrix Receiver home page.
wiportalmode	<String>	Read-write	Layout on the Access Interface. The COMPACT value indicates the use of small icons. Possible values = NORMAL, COMPACT
clientchoices	<String>	Read-write	Provide users with multiple logon options. With client choices, users have the option of logging on by using the NetScaler Gateway Plug-in for Windows, NetScaler Gateway Plug-in for Java, the Web Interface, or clientless access from one location. Depending on how NetScaler Gateway is configured, users are presented with up to three icons for logon choices. The most common are the NetScaler Gateway Plug-in for Windows, Web Interface, and clientless access. Possible values = ON, OFF
epaclienttype	<String>	Read-write	Choose between two types of End point Windows Client a) Application Agent - which always runs in the task bar as a standalone application and also has a supporting service which runs permanently when installed b) Activex Control - ActiveX control run by Microsoft Internet Explorer. Possible values = AGENT, PLUGIN
iipdnssuffix	<String>	Read-write	An intranet IP DNS suffix. When a user logs on to NetScaler Gateway and is assigned an IP address, a DNS record for the user name and IP address combination is added to the NetScaler Gateway DNS cache. You can configure a DNS suffix to append to the user name when the DNS record is added to the cache. You can reach to the host from where the user is logged on by using the user's name, which can be easier to remember than an IP address. When the user logs off from NetScaler Gateway, the record is removed from the DNS cache. Minimum length = 1
forcedtimeout	<Double>	Read-write	Force a disconnection from the NetScaler Gateway Plug-in with NetScaler Gateway after a specified number of minutes. If the session closes, the user must log on again. Minimum value = 1 Maximum value = 65535
			Number of minutes to warn a user before the user session is disconnected.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Maximum value = 255
ntdomain	<String>	Read-write	Single sign-on domain to use for single sign-on to applications in the internal network. This setting can be overwritten by the domain that users specify at the time of logon or by the domain that the authentication server returns. Minimum length = 1 Maximum length = 32
clientlessvpnmode	<String>	Read-write	Enable clientless access for web, XenApp or XenDesktop, and FileShare resources without installing the NetScaler Gateway Plug-in. Available settings function as follows: * ON - Allow only clientless access. * OFF - Allow clientless access after users log on with the NetScaler Gateway Plug-in. * DISABLED - Do not allow clientless access. Possible values = ON, OFF, DISABLED
emailhome	<String>	Read-write	Web address for the web-based email, such as Outlook Web Access.
clientlessmodeurlencoding	<String>	Read-write	When clientless access is enabled, you can choose to encode the addresses of internal web applications or to leave the address as clear text. Available settings function as follows: * OPAQUE - Use standard encoding mechanisms to make the domain and protocol part of the resource unclear to users. * CLEAR - Do not encode the web address and make it visible to users. * ENCRYPT - Allow the domain and protocol to be encrypted using a session key. When the web address is encrypted, the URL is different for each user session for the same web resource. If users bookmark the encoded web address, save it in the web browser and then log off, they cannot connect to the web address when they log on and use the bookmark. If users save the encrypted bookmark in the Access Interface during their session, the bookmark works each time the user logs on. Possible values = TRANSPARENT, OPAQUE, ENCRYPT
clientlesspersistentcookie	<String>	Read-write	State of persistent cookies in clientless access mode. Persistent cookies are required for accessing certain features of SharePoint, such as opening and editing Microsoft Word, Excel, and PowerPoint documents hosted on the SharePoint server. A persistent cookie remains on the user device and is sent with each HTTP request. NetScaler Gateway encrypts the persistent cookie before sending it to the plug-in on the user device, and refreshes the cookie periodically as long as the session exists. The cookie becomes stale if the session ends. Available settings function as follows: * ALLOW - Enable persistent cookies. Users can open and edit Microsoft documents stored in SharePoint. * DENY - Disable persistent cookies. Users cannot open and edit Microsoft documents stored in SharePoint. * PROMPT - Prompt users to allow or deny persistent cookies during the session. Persistent cookies are not required for clientless access if users do not connect to SharePoint. Possible values = ALLOW, DENY, PROMPT
			Specify groups that have permission to log on to NetScaler Gateway. Users

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			have valid credentials. Minimum length = 1 Maximum length = 511
securebrowse	<String>	Read-write	Allow users to connect through NetScaler Gateway to network resources from iOS and Android mobile devices with Citrix Receiver. Users do not need to establish a full VPN tunnel to access resources in the secure network. Possible values = ENABLED, DISABLED
storefronturl	<String>	Read-write	Web address for StoreFront to be used in this session for enumeration of resources from XenApp or XenDesktop. Minimum length = 1 Maximum length = 255
kcdaccount	<String>	Read-write	The kcd account details to be used in SSO. Minimum length = 1 Maximum length = 32
rdpclientprofilename	<String>	Read-write	Name of the RDP profile associated with the vserver. Minimum length = 1 Maximum length = 31
windowspluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Win. Possible values = Always, Essential, Never
macpluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Mac. Possible values = Always, Essential, Never
linuxpluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Linux. Possible values = Always, Essential, Never
iconwithreceiver	<String>	Read-write	Option to decide whether to show plugin icon along with receiver. Possible values = ON, OFF
clientidletimeoutwarning	<Double>	Read-only	The time after which the client gets a timeout warning, in minutes.
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnsessionaction":{
    "name":<String_value>,
    "useraccounting":<String_value>,
    "httpport":<Integer[]_value>,
    "winsip":<String_value>,
    "dnsvservername":<String_value>,
    "splitdns":<String_value>,
    "sesstimeout":<Double_value>,
    "clientsecurity":<String_value>,
    "clientsecuritygroup":<String_value>,
    "clientsecuritymessage":<String_value>,
    "clientsecuritylog":<String_value>,
    "splittunnel":<String_value>,
    "locallanaccess":<String_value>,
    "rfc1918":<String_value>,
    "spoofiip":<String_value>,
    "killconnections":<String_value>,
    "transparentinterception":<String_value>,
    "windowsclienttype":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "defaultauthorizationaction":<String_value>,
    "authorizationgroup":<String_value>,
    "smartgroup":<String_value>,
    "clientidletimeout":<Double_value>,
    "proxy":<String_value>,
    "allprotocolproxy":<String_value>,
    "httpproxy":<String_value>,
    "ftpproxy":<String_value>,
    "socksproxy":<String_value>,
    "gopherproxy":<String_value>,
    "sslproxy":<String_value>,
    "proxyexception":<String_value>,
    "proxylocalbypass":<String_value>,
    "clientcleanupprompt":<String_value>,
    "forcecleanup":<String[]_value>,
    "clientoptions":<String[]_value>,
    "clientconfiguration":<String[]_value>,
    "sso":<String_value>,
    "ssocredential":<String_value>,
    "windowsautologon":<String_value>,
    "usemip":<String_value>,
    "useiip":<String_value>,
    "clientdebug":<String_value>,
    "loginscript":<String_value>,
    "logoutscript":<String_value>,
    "homepage":<String_value>,
    "icaproxy":<String_value>,
    "wihome":<String_value>,
    "wihomeaddresstype":<String_value>,
    "citrixreceiverhome":<String_value>,
    "wiportalmode":<String_value>,
    "clientchoices":<String_value>,
    "epaclienttype":<String_value>,
    "iipdnssuffix":<String_value>,
    "forcedtimeout":<Double_value>,
    "forcedtimeoutwarning":<Double_value>,
    "ntdomain":<String_value>,
    "clientlessvpnmode":<String_value>,
    "emailhome":<String_value>,
    "clientlessmodeurlencoding":<String_value>,
    "clientlesspersistentcookie":<String_value>,
    "allowedlogingroups":<String_value>,
    "securebrowse":<String_value>,
    "storefronturl":<String_value>,
    "kcdaccount":<String_value>,
    "rdpclientprofilename":<String_value>,
    "windowspluginupgrade":<String_value>,
    "macpluginupgrade":<String_value>,
    "linuxpluginupgrade":<String_value>,
    "iconwithreceiver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnsessionaction":{
    "name":<String_value>,
    "useraccounting":<String_value>,
    "httpport":<Integer[]_value>,
    "winsip":<String_value>,
    "dnsvservername":<String_value>,
    "splitdns":<String_value>,
    "sesstimeout":<Double_value>,
    "clientsecurity":<String_value>,
    "clientsecuritygroup":<String_value>,
    "clientsecuritymessage":<String_value>,
    "clientsecuritylog":<String_value>,
    "splittunnel":<String_value>,
    "locallanaccess":<String_value>,
    "rfc1918":<String_value>,
    "spoofiip":<String_value>,
    "killconnections":<String_value>,
    "transparentinterception":<String_value>,
    "windowsclienttype":<String_value>,
    "defaultauthorizationaction":<String_value>,
    "authorizationgroup":<String_value>,
    "smartgroup":<String_value>,
    "clientidletimeout":<Double_value>,
    "proxy":<String_value>,
    "allprotocolproxy":<String_value>,
    "httpproxy":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ftpproxy":<String_value>,
    "socksproxy":<String_value>,
    "gopherproxy":<String_value>,
    "sslproxy":<String_value>,
    "proxyexception":<String_value>,
    "proxylocalbypass":<String_value>,
    "clientcleanupprompt":<String_value>,
    "forcecleanup":<String[]_value>,
    "clientoptions":<String[]_value>,
    "clientconfiguration":<String[]_value>,
    "sso":<String_value>,
    "ssocredential":<String_value>,
    "windowsautologon":<String_value>,
    "usemip":<String_value>,
    "useiip":<String_value>,
    "clientdebug":<String_value>,
    "loginscript":<String_value>,
    "logoutscript":<String_value>,
    "homepage":<String_value>,
    "icaproxy":<String_value>,
    "wihome":<String_value>,
    "wihomeaddresstype":<String_value>,
    "citrixreceiverhome":<String_value>,
    "wiportalmode":<String_value>,
    "clientchoices":<String_value>,
    "epaclienttype":<String_value>,
    "iipdnssuffix":<String_value>,
    "forcedtimeout":<Double_value>,
    "forcedtimeoutwarning":<Double_value>,
    "ntdomain":<String_value>,
    "clientlessvpnmode":<String_value>,
    "emailhome":<String_value>,
    "clientlessmodeurlencoding":<String_value>,
    "clientlesspersistentcookie":<String_value>,
    "allowedlogingroups":<String_value>,
    "securebrowse":<String_value>,
    "storefronturl":<String_value>,
    "kcdaccount":<String_value>,
    "rdpclientprofilename":<String_value>,
    "windowspluginupgrade":<String_value>,
    "macpluginupgrade":<String_value>,
    "linuxpluginupgrade":<String_value>,
    "iconwithreceiver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction?**action=unset**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnsessionaction":{
    "name":<String_value>,
    "useraccounting":true,
    "httpport":true,
    "winsip":true,
    "dnsvservername":true,
    "splitdns":true,
    "sesstimeout":true,
    "clientsecurity":true,
    "clientsecuritygroup":true,
    "clientsecuritymessage":true,
    "clientsecuritylog":true,
    "splittunnel":true,
    "locallanaccess":true,
    "rfc1918":true,
    "spoofiip":true,
    "killconnections":true,
    "transparentinterception":true,
    "windowsclienttype":true,
    "defaultauthorizationaction":true,
    "authorizationgroup":true,
    "smartgroup":true,
    "clientidletimeout":true,
    "proxy":true,
    "allprotocolproxy":true,
    "httpproxy":true,
    "ftpproxy":true,
    "socksproxy":true,
    "gopherproxy":true,
    "sslproxy":true,
    "proxyexception":true,
    "proxylocalbypass":true,
    "clientcleanupprompt":true,
    "forcecleanup":true,
    "clientoptions":true,
    "clientconfiguration":true,
    "sso":true,
    "ssocredential":true,
    "windowsautologon":true,
    "usemip":true,
    "useiip":true,
    "clientdebug":true,
    "loginscript":true,
    "logoutscript":true,
    "homepage":true,
    "icaproxy":true,
    "wihome":true,
    "citrixreceiverhome":true,
    "wiportalmode":true,
    "clientchoices":true,
    "epaclienttype":true,
    "iipdnssuffix":true,
    "forcedtimeout":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
        "forcedtimeoutwarning":true,
        "ntdomain":true,
        "clientlessvpnmode":true,
        "emailhome":true,
        "clientlessmodeurlencoding":true,
        "clientlesspersistentcookie":true,
        "allowedlogingroups":true,
        "securebrowse":true,
        "storefronturl":true,
        "kcdaccount":true,
        "rdpclientprofilename":true,
        "windowspluginupgrade":true,
        "macpluginupgrade":true,
        "linuxpluginupgrade":true,
        "iconwithreceiver":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnsessionaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnsessionaction resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsessionaction": [ {
    "name":<String_value>,
    "useraccounting":<String_value>,
    "httpport":<Integer[]_value>,
    "winsip":<String_value>,
    "dnsvservername":<String_value>,
    "splitdns":<String_value>,
    "sesstimeout":<Double_value>,
    "clientsecurity":<String_value>,
    "clientsecuritygroup":<String_value>,
    "clientsecuritymessage":<String_value>,
    "clientsecuritylog":<String_value>,
    "splittunnel":<String_value>,
    "locallanaccess":<String_value>,
    "rfc1918":<String_value>,
    "spoofiip":<String_value>,
    "killconnections":<String_value>,
    "transparentinterception":<String_value>,
    "windowsclienttype":<String_value>,
    "defaultauthorizationaction":<String_value>,
    "authorizationgroup":<String_value>,
    "smartgroup":<String_value>,
    "clientidletimeout":<Double_value>,
    "clientidletimeoutwarning":<Double_value>,
    "proxy":<String_value>,
    "allprotocolproxy":<String_value>,
    "httpproxy":<String_value>,
    "ftpproxy":<String_value>,
    "socksproxy":<String_value>,
    "gopherproxy":<String_value>,
    "sslproxy":<String_value>,
    "proxyexception":<String_value>,
    "proxylocalbypass":<String_value>,
    "clientcleanupprompt":<String_value>,
    "forcecleanup":<String[]_value>,
    "clientoptions":<String[]_value>,
    "clientconfiguration":<String[]_value>,
    "sso":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ssocredential":<String_value>,
    "windowsautologon":<String_value>,
    "usemip":<String_value>,
    "useiip":<String_value>,
    "clientdebug":<String_value>,
    "loginscript":<String_value>,
    "logoutscript":<String_value>,
    "homepage":<String_value>,
    "icaproxy":<String_value>,
    "wihome":<String_value>,
    "wihomeaddresstype":<String_value>,
    "citrixreceiverhome":<String_value>,
    "wiportalmode":<String_value>,
    "clientchoices":<String_value>,
    "epaclienttype":<String_value>,
    "iipdnssuffix":<String_value>,
    "forcedtimeout":<Double_value>,
    "forcedtimeoutwarning":<Double_value>,
    "ntdomain":<String_value>,
    "clientlessvpnmode":<String_value>,
    "clientlessmodeurlencoding":<String_value>,
    "clientlesspersistentcookie":<String_value>,
    "emailhome":<String_value>,
    "allowedlogingroups":<String_value>,
    "securebrowse":<String_value>,
    "storefronturl":<String_value>,
    "kcdaccount":<String_value>,
    "rdpclientprofilename":<String_value>,
    "builtin":<String[]_value>,
    "windowspluginupgrade":<String_value>,
    "macpluginupgrade":<String_value>,
    "linuxpluginupgrade":<String_value>,
    "iconwithreceiver":<String_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsessionaction": [ {
    "name":<String_value>,
    "useraccounting":<String_value>,
    "httpport":<Integer[]_value>,
    "winsip":<String_value>,
    "dnsvservername":<String_value>,
    "splitdns":<String_value>,
    "sesstimeout":<Double_value>,
    "clientsecurity":<String_value>,
    "clientsecuritygroup":<String_value>,
    "clientsecuritymessage":<String_value>,
    "clientsecuritylog":<String_value>,
    "splittunnel":<String_value>,
    "locallanaccess":<String_value>,
    "rfc1918":<String_value>,
    "spoofiip":<String_value>,
    "killconnections":<String_value>,
    "transparentinterception":<String_value>,
    "windowsclienttype":<String_value>,
    "defaultauthorizationaction":<String_value>,
    "authorizationgroup":<String_value>,
    "smartgroup":<String_value>,
    "clientidletimeout":<Double_value>,
    "clientidletimeoutwarning":<Double_value>,
    "proxy":<String_value>,
    "allprotocolproxy":<String_value>,
    "httpproxy":<String_value>,
    "ftpproxy":<String_value>,
    "socksproxy":<String_value>,
    "gopherproxy":<String_value>,
    "sslproxy":<String_value>,
    "proxyexception":<String_value>,
    "proxylocalbypass":<String_value>,
    "clientcleanupprompt":<String_value>,
    "forcecleanup":<String[]_value>,
    "clientoptions":<String[]_value>,
    "clientconfiguration":<String[]_value>,
    "sso":<String_value>,
    "ssocredential":<String_value>,
    "windowsautologon":<String_value>,
    "usemip":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "useiip":<String_value>,
    "clientdebug":<String_value>,
    "loginscript":<String_value>,
    "logoutscript":<String_value>,
    "homepage":<String_value>,
    "icaproxy":<String_value>,
    "wihome":<String_value>,
    "wihomeaddresstype":<String_value>,
    "citrixreceiverhome":<String_value>,
    "wiportalmode":<String_value>,
    "clientchoices":<String_value>,
    "epaclienttype":<String_value>,
    "iipdnssuffix":<String_value>,
    "forcedtimeout":<Double_value>,
    "forcedtimeoutwarning":<Double_value>,
    "ntdomain":<String_value>,
    "clientlessvpnmode":<String_value>,
    "clientlessmodeurlencoding":<String_value>,
    "clientlesspersistentcookie":<String_value>,
    "emailhome":<String_value>,
    "allowedlogingroups":<String_value>,
    "securebrowse":<String_value>,
    "storefronturl":<String_value>,
    "kcdaccount":<String_value>,
    "rdpclientprofilename":<String_value>,
    "builtin":<String[]_value>,
    "windowspluginupgrade":<String_value>,
    "macpluginupgrade":<String_value>,
    "linuxpluginupgrade":<String_value>,
    "iconwithreceiver":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{ "vpnsessionaction": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnsessionpolicy

May 20, 2016

Configuration for VPN session policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the new session policy that is applied after the user logs on to NetScaler Gateway. Minimum length = 1
rule	<String>	Read-write	Expression, or name of a named expression, specifying the traffic that matches the policy. Can be written in either default or classic syntax. Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Action to be applied by the new session policy if the rule criteria are met. Minimum length = 1
builtin	<String[]>	Read-only	Indicates that a variable is a built-in (SYSTEM INTERNAL) type. Possible values = MODIFIABLE, DELETABLE, IMMUTABLE, PARTITION_ALL
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnsessionpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy/name_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnsessionpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnsessionpolicy":{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "name":<String_value>,
    "rule":true,
    "action":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnsessionpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnsessionpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsessionpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsessionpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>,
      "builtin":<String[]_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsessionpolicy": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnsessionpolicy_aaagroup_binding

May 20, 2016

Binding object showing the aaagroup that can be bound to vpnsessionpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy to display. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnsessionpolicy_aaagroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_aaagroup_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnsessionpolicy_aaagroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsessionpolicy_aaagroup_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnsessionpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnsessionpolicy_binding

May 20, 2016

Binding object which returns the resources bound to vpnsessionpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the session policy to display. Minimum length = 1
vpnsessionpolicy_aaauser_binding	<vpnsessionpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to vpnsessionpolicy.
vpnsessionpolicy_vpnglobal_binding	<vpnsessionpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to vpnsessionpolicy.
vpnsessionpolicy_vpnvserver_binding	<vpnsessionpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to vpnsessionpolicy.
vpnsessionpolicy_aaagroup_binding	<vpnsessionpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to vpnsessionpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "vpnsessionpolicy_binding": [ {
      "name":<String_value>,
      "vpnsessionpolicy_aaauser_binding":<vpnsessionpolicy_aaauser_binding[]_value>,
      "vpnsessionpolicy_vpnglobal_binding":<vpnsessionpolicy_vpnglobal_binding[]_value>,
      "vpnsessionpolicy_vpnvserver_binding":<vpnsessionpolicy_vpnvserver_binding[]_value>,
      "vpnsessionpolicy_aaagroup_binding":<vpnsessionpolicy_aaagroup_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpntrafficaction

May 20, 2016

Configuration for VPN traffic action resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the traffic action. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after a traffic action is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my action" or 'my action'). Minimum length = 1
qual	<String>	Read-write	Protocol, either HTTP or TCP, to be used with the action. Possible values = http, tcp
apptimeout	<Double>	Read-write	Maximum amount of time, in minutes, a user can stay logged on to the web application. Minimum value = 1 Maximum value = 715827
sso	<String>	Read-write	Provide single sign-on to the web application. Possible values = ON, OFF
hdx	<String>	Read-write	Provide hdx proxy to the ICA traffic. Possible values = ON, OFF
formssoaction	<String>	Read-write	Name of the form-based single sign-on profile. Form-based single sign-on allows users to log on one time to all protected applications in your network, instead of requiring them to log on separately to access each one.
fta	<String>	Read-write	Specify file type association, which is a list of file extensions that users are allowed to open. Possible values = ON, OFF
wanscaler	<String>	Read-write	Use the Repeater Plug-in to optimize network traffic. Possible values = ON, OFF
kcdaccount	<String>	Read-write	Kerberos constrained delegation account name. Default value: "Default" Minimum length = 1 Maximum length = 32

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
samlssoprofile	<String>	Read-write	Minimum length = 1
proxy	<String>	Read-write	IP address and Port of the proxy server to be used for HTTP access for this request. Minimum length = 1
userexpression	<String>	Read-write	expression that will be evaluated to obtain username for SingleSignOn. Maximum length = 256
passwdexpression	<String>	Read-write	expression that will be evaluated to obtain password for SingleSignOn. Maximum length = 256
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpntrafficaction":{
    "name":<String_value>,
    "qual":<String_value>,
    "apptimeout":<Double_value>,
    "sso":<String_value>,
    "hdx":<String_value>,
    "formssoaction":<String_value>,
    "fta":<String_value>,
    "wanscaler":<String_value>,
    "kcdaccount":<String_value>,
    "samlssoprofile":<String_value>,
    "proxy":<String_value>,
    "userexpression":<String_value>,
    "passwdexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpntrafficaction":{
    "name":<String_value>,
    "apptimeout":<Double_value>,
    "sso":<String_value>,
    "hdx":<String_value>,
    "formssoaction":<String_value>,
    "fta":<String_value>,
    "wanscaler":<String_value>,
    "kcdaccount":<String_value>,
    "samlssoprofile":<String_value>,
    "proxy":<String_value>,
    "userexpression":<String_value>,
    "passwdexpression":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpntrafficaction":{
    "name":<String_value>,
    "wanscaler":true,
    "kcdaccount":true,
    "proxy":true,
    "userexpression":true,
    "passwdexpression":true
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpntrafficaction resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpntrafficaction resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "vpntrafficaction": [ {
      "name":<String_value>,
      "qual":<String_value>,
      "apptimeout":<Double_value>,
      "sso":<String_value>,
      "formssoaction":<String_value>,
      "hdx":<String_value>,
      "fta":<String_value>,
      "wanscaler":<String_value>,
      "kcdaccount":<String_value>,
      "samlssoprofile":<String_value>,
      "proxy":<String_value>,
      "userexpression":<String_value>,
      "passwdexpression":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpntrafficaction": [ {
      "name":<String_value>,
      "qual":<String_value>,
      "apptimeout":<Double_value>,
      "sso":<String_value>,
      "formssoaction":<String_value>,
      "hdx":<String_value>,
      "fta":<String_value>,
      "wanscaler":<String_value>,
      "kcdaccount":<String_value>,
      "samlssoprofile":<String_value>,
      "proxy":<String_value>,
      "userexpression":<String_value>,
      "passwdexpression":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficaction?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpntrafficaction": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnsessionpolicy_aaauser_binding

May 20, 2016

Binding object showing the aaauser that can be bound to vpnsessionpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy to display. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnsessionpolicy_aaauser_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_aaauser_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnsessionpolicy_aaauser_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsessionpolicy_aaauser_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnsessionpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnsessionpolicy_vpnglobal_binding

May 20, 2016

Binding object showing the vpnglobal that can be bound to vpnsessionpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy to display. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnsessionpolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnsessionpolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsessionpolicy_vpnglobal_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnsessionpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnsessionpolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to vpnsessionpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
boundto	<String>	Read-write	The entity name to which policy is bound.
name	<String>	Read-write	Name of the session policy to display. Minimum length = 1
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnsessionpolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnsessionpolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnsessionpolicy_vpnvserver_binding": [ {
    "boundto":<String_value>,
    "name":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnsessionpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnsessionpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpntrafficpolicy

Jul 07, 2017

Configuration for VPN traffic policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the traffic policy. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the policy is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my policy" or 'my policy'). Minimum length = 1
rule	<String>	Read-write	Expression, or name of a named expression, against which traffic is evaluated. Written in the classic or default syntax. Note: Maximum length of a string literal in the expression is 255 characters. A longer string can be split into smaller strings of up to 255 characters each, and the smaller strings concatenated with the + operator. For example, you can create a 500-character string as follows: '"<string of 255 characters>" + "<string of 245 characters>"' The following requirements apply only to the NetScaler CLI: * If the expression includes one or more spaces, enclose the entire expression in double quotation marks. * If the expression itself includes double quotation marks, escape the quotations by using the \ character. * Alternatively, you can use single quotation marks to enclose the rule, in which case you do not have to escape the double quotation marks.
action	<String>	Read-write	Action to apply to traffic that matches the policy. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpntrafficpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpntrafficpolicy":{
    "name":<String_value>,
    "rule":<String_value>,
    "action":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Payload:

```
{"vpntrafficpolicy":{
    "name":<String_value>,
    "rule":true,
    "action":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpntrafficpolicy resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpntrafficpolicy resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpntrafficpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpntrafficpolicy": [ {
      "name":<String_value>,
      "rule":<String_value>,
      "action":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpntrafficpolicy": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpntrafficpolicy_aaagroup_binding

May 20, 2016

Binding object showing the aaagroup that can be bound to vpntrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_aaagroup_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_aaagroup_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpntrafficpolicy_aaagroup_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_aaagroup_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpntrafficpolicy_aaagroup_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpntrafficpolicy_aaagroup_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_aaagroup_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpntrafficpolicy_aaagroup_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpntrafficpolicy_aaauser_binding

Jul 07, 2017

Binding object showing the aaauser that can be bound to vpntrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_aaauser_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_aaauser_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpntrafficpolicy_aaauser_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_aaauser_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpntrafficpolicy_aaauser_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpntrafficpolicy_aaauser_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_aaauser_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpntrafficpolicy_aaauser_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpntrafficpolicy_binding

May 20, 2016

Binding object which returns the resources bound to vpntrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
vpntrafficpolicy_aaauser_binding	<vpntrafficpolicy_aaauser_binding[]>	Read-only	aaauser that can be bound to vpntrafficpolicy.
vpntrafficpolicy_vpnglobal_binding	<vpntrafficpolicy_vpnglobal_binding[]>	Read-only	vpnglobal that can be bound to vpntrafficpolicy.
vpntrafficpolicy_vpnvserver_binding	<vpntrafficpolicy_vpnvserver_binding[]>	Read-only	vpnvserver that can be bound to vpntrafficpolicy.
vpntrafficpolicy_aaagroup_binding	<vpntrafficpolicy_aaagroup_binding[]>	Read-only	aaagroup that can be bound to vpntrafficpolicy.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "vpntrafficpolicy_binding": [ {
      "name":<String_value>,
      "vpntrafficpolicy_aaagroup_binding":<vpntrafficpolicy_aaagroup_binding[]_value>,
      "vpntrafficpolicy_aaauser_binding":<vpntrafficpolicy_aaauser_binding[]_value>,
      "vpntrafficpolicy_vpnvserver_binding":<vpntrafficpolicy_vpnvserver_binding[]_value>,
      "vpntrafficpolicy_vpnglobal_binding":<vpntrafficpolicy_vpnglobal_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpntrafficpolicy_vpnglobal_binding

May 20, 2016

Binding object showing the vpnglobal that can be bound to vpntrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_vpnglobal_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_vpnglobal_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpntrafficpolicy_vpnglobal_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_vpnglobal_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpntrafficpolicy_vpnglobal_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpntrafficpolicy_vpnglobal_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_vpnglobal_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpntrafficpolicy_vpnglobal_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpntrafficpolicy_vpnvserver_binding

May 20, 2016

Binding object showing the vpnvserver that can be bound to vpntrafficpolicy.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the traffic policy for which to display detailed information. Minimum length = 1
boundto	<String>	Read-write	The entity name to which policy is bound.
priority	<Double>	Read-only	.
activepolicy	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_vpnvserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_vpnvserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpntrafficpolicy_vpnvserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_vpnvserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpntrafficpolicy_vpnvserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpntrafficpolicy_vpnvserver_binding": [ {
    "name":<String_value>,
    "boundto":<String_value>,
    "priority":<Double_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "activepolicy":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpntrafficpolicy_vpnvserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpntrafficpolicy_vpnvserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver

May 20, 2016

Configuration for VPN virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name for the NetScaler Gateway virtual server. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Can be changed after the virtual server is created. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my server" or 'my server'). Minimum length = 1
servicetype	<String>	Read-write	Protocol used by the NetScaler Gateway virtual server. Default value: SSL Possible values = SSL
ipv46	<String>	Read-write	IPv4 or IPv6 address of the NetScaler Gateway virtual server. Usually a public IP address. User devices send connection requests to this IP address. Minimum length = 1
range	<Double>	Read-write	Range of NetScaler Gateway virtual server IP addresses. The consecutively numbered range of IP addresses begins with the address specified by the IP Address parameter. In the configuration utility, select Network VServer to enter a range. Default value: 1 Minimum value = 1
port	<Integer>	Read-write	TCP port on which the virtual server listens. Range 1 - 65535
state	<String>	Read-write	State of the virtual server. If the virtual server is disabled, requests are not processed. Default value: ENABLED Possible values = ENABLED, DISABLED
authentication	<String>	Read-write	Require authentication for users connecting to NetScaler Gateway. Default value: ON Possible values = ON, OFF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
doublehop	<String>	Read-write	double-hop deployment provides an extra layer of security for the internal network by using three firewalls to divide the DMZ into two stages. Such a deployment can have one appliance in the DMZ and one appliance in the secure network. Default value: DISABLED Possible values = ENABLED, DISABLED
maxaaausers	<Double>	Read-write	Maximum number of concurrent user sessions allowed on this virtual server. The actual number of users allowed to log on to this virtual server depends on the total number of user licenses.
icaonly	<String>	Read-write	- When set to ON, it implies Basic mode where the user can log on using either Citrix Receiver or a browser and get access to the published apps configured at the XenApp/XenDEsktop environment pointed out by the WIHome parameter. Users are not allowed to connect using the NetScaler Gateway Plug-in and end point scans cannot be configured. Number of users that can log in and access the apps are not limited by the license in this mode. - When set to OFF, it implies Smart Access mode where the user can log on using either Citrix Receiver or a browser or a NetScaler Gateway Plug-in. The admin can configure end point scans to be run on the client systems and then use the results to control access to the published apps. In this mode, the client can connect to the gateway in other client modes namely VPN and CVPN. Number of users that can log in and access the resources are limited by the CCU licenses in this mode. Default value: OFF Possible values = ON, OFF
icaproxysessionmigration	<String>	Read-write	This option determines if an existing ICA Proxy session is transferred when the user logs on from another device. Default value: OFF Possible values = ON, OFF
dtls	<String>	Read-write	This option starts/stops the turn service on the vserver. Default value: OFF Possible values = ON, OFF
loginonce	<String>	Read-write	This option enables/disables seamless SSO for this Vserver. Default value: OFF Possible values = ON, OFF
advancedepa	<String>	Read-write	This option tells whether advanced EPA is enabled on this virtual server. Default value: OFF Possible values = ON, OFF
devicecert	<String>	Read-write	Indicates whether device certificate check as a part of EPA is on or off. Default value: OFF Possible values = ON, OFF
certkeynames	<String>	Read-write	Name of the certificate key that was bound to the corresponding SSL virtual server as the Certificate Authority for the device certificate. Minimum length = 1 Maximum length = 127

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
downstateflush	<String>	Read-write	Close existing connections when the virtual server is marked DOWN, which means the server might have timed out. Disconnecting existing connections frees resources and in certain cases speeds recovery of overloaded load balancing setups. Enable this setting on servers in which the connections can safely be closed when they are marked DOWN. Do not enable DOWN state flush on servers that must complete their transactions. Default value: ENABLED Possible values = ENABLED, DISABLED
listenpolicy	<String>	Read-write	String specifying the listen policy for the NetScaler Gateway virtual server. Can be either a named expression or a default syntax expression. The NetScaler Gateway virtual server processes only the traffic for which the expression evaluates to true. Default value: "none"
listenpriority	<Double>	Read-write	Integer specifying the priority of the listen policy. A higher number specifies a lower priority. If a request matches the listen policies of more than one virtual server, the virtual server whose listen policy has the highest priority (the lowest priority number) accepts the request. Default value: 101 Minimum value = 0 Maximum value = 100
tcpprofilename	<String>	Read-write	Name of the TCP profile to assign to this virtual server. Minimum length = 1 Maximum length = 127
httpprofilename	<String>	Read-write	Name of the HTTP profile to assign to this virtual server. Default value: "nshttp_default_strict_validation" Minimum length = 1 Maximum length = 127
comment	<String>	Read-write	Any comments associated with the virtual server.
appflowlog	<String>	Read-write	Log AppFlow records that contain standard NetFlow or IPFIX information, such as time stamps for the beginning and end of a flow, packet count, and byte count. Also log records that contain application-level information, such as HTTP web addresses, HTTP request methods and response status codes, server response time, and latency. Default value: DISABLED Possible values = ENABLED, DISABLED
icmpvsrresponse	<String>	Read-write	Criterion for responding to PING requests sent to this virtual server. If this parameter is set to ACTIVE, respond only if the virtual server is available. With the PASSIVE setting, respond even if the virtual server is not available. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
			A host route is injected according to the setting on the virtual servers. * If set to PASSIVE on all the virtual servers that share the IP address, the appliance always injects the hostroute. * If set to ACTIVE on all the virtual servers that share the IP address, the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			* If set to ACTIVE on some virtual servers and PASSIVE on the others, the appliance injects even if one virtual server set to ACTIVE is UP. Default value: PASSIVE Possible values = PASSIVE, ACTIVE
netprofile	<String>	Read-write	The name of the network profile. Minimum length = 1 Maximum length = 127
cginfrahomepageredirect	<String>	Read-write	When client requests ShareFile resources and NetScaler Gateway detects that the user is unauthenticated or the user session has expired, disabling this option takes the user to the originally requested ShareFile resource after authentication (instead of taking the user to the default VPN home page). Default value: ENABLED Possible values = ENABLED, DISABLED
maxloginattempts	<Double>	Read-write	Maximum number of logon attempts. Minimum value = 1 Maximum value = 255
failedlogintimeout	<Double>	Read-write	Number of minutes an account will be locked if user exceeds maximum permissible attempts. Minimum value = 1
l2conn	<String>	Read-write	Use Layer 2 parameters (channel number, MAC address, and VLAN ID) in addition to the 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) that is used to identify a connection. Allows multiple TCP and non-TCP connections with the same 4-tuple to coexist on the NetScaler appliance. Possible values = ON, OFF
deploymenttype	<String>	Read-write	. Default value: 5 Possible values = NONE, ICA_WEBINTERFACE, ICA_STOREFRONT, MOBILITY, WIONNS
rdpserverprofilename	<String>	Read-write	Name of the RDP server profile associated with the vserver. Minimum length = 1 Maximum length = 31
windowsepapluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Win. Possible values = Always, Essential, Never
linuxepapluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Linux. Possible values = Always, Essential, Never
macepapluginupgrade	<String>	Read-write	Option to set plugin upgrade behaviour for Mac. Possible values = Always, Essential, Never
userdomains	<String>	Read-write	List of user domains specified as comma seperated value.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			offload authentication to AAA vserver for multi-factor(nFactor) authentication.
newname	<String>	Read-write	New name for the NetScaler Gateway virtual server. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. The following requirement applies only to the NetScaler CLI: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my server" or 'my server'). Minimum length = 1
ip	<String>	Read-only	The Virtual IP address of the VPN virtual server.
value	<String>	Read-only	Indicates whether or not the certificate is bound or if SSL offload is disabled. Possible values = Certkey not bound, SSL feature disabled
type	<String>	Read-only	The type of virtual server; for example, CONTENT based or ADDRESS based. Possible values = CONTENT, ADDRESS
curstate	<String>	Read-only	The current state of the virtual server, as UP, DOWN, BUSY, and so on. Possible values = UP, DOWN, UNKNOWN, BUSY, OUT OF SERVICE, GOING OUT OF SERVICE, DOWN WHEN GOING OUT OF SERVICE, NS_EMPTY_STR, Unknown, DISABLED
status	<Integer>	Read-only	Whether or not this virtual server responds to ARPs and whether or not round-robin selection is temporarily in effect.
cachetype	<String>	Read-only	Virtual server cache type. The options are: TRANSPARENT, REVERSE, and FORWARD. Possible values = TRANSPARENT, REVERSE, FORWARD
redirect	<String>	Read-only	The cache redirect policy. The valid redirect policies are: l. CACHE - Directs all requests to the cache. 2. POLICY - Applies cache redirection policy to determine whether the request should be directed to the cache or origin. This is the default setting. 3. ORIGIN - Directs all requests to the origin server. Possible values = CACHE, POLICY, ORIGIN
			This argument is used only when configuring content switching on the specified virtual server. This is applicable only if both the URL and RULE-based policies have been configured on the same virtual server. It specifies the type of policy (URL or RULE) that takes precedence on the content switching virtual server. The default setting is RULE. l URL - In this case, the incoming request is matched against the URL-based policies before the rule-based policies. l RULE - In this case, the incoming request is matched against the rule-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			For all URL-based policies, the precedence hierarchy is: 1. Domain and exact URL 2. Domain, prefix, and suffix 3. Domain and suffix 4. Domain and prefix 5. Domain only 6. Exact URL 7. Prefix and suffix 8. Suffix only 9. Prefix only 10. Default. Possible values = RULE, URL
redirecturl	<String>	Read-only	The URL where traffic is redirected if the virtual server in system becomes unavailable. WARNING! Make sure that the domain you specify in the URL does not match the domain specified in the -d domainName argument of the ###add cs policy### command. If the same domain is specified in both arguments, the request will be continuously redirected to the same unavailable virtual server in the system. If so, the user may not get the requested content.
curaaausers	<Double>	Read-only	The number of current users logged on to this virtual server.
curtotalusers	<Double>	Read-only	The total number of current users connected through this virtual server.
domain	<String>	Read-only	The domain name of the server for which a service needs to be added. If the IP address has been specified, the domain name does not need to be specified.
rule	<String>	Read-only	The name of the rule, or expression, if any, that policy for the VPN server is to use. Rules are combinations of expressions. Expressions are simple conditions, such as a test for equality, applied to operands, such as a URL string or an IP address. Expression syntax is described in the Installation and Configuration Guide. The default rule is ns_true. Minimum length = 1
policy	<String>	Read-only	The name of the policy, if any, bound to the VPN virtual server.
servicename	<String>	Read-only	The name of the service, if any, to which the virtual server policy is bound.
weight	<Double>	Read-only	Weight for this service, if any. This weight is used when the system performs load balancing, giving greater priority to a specific service. It is useful when the services bound to a virtual server are of different capacity.
cachevserver	<String>	Read-only	The name of the default target cache virtual server, if any, to which requests are redirected.
backupvserver	<String>	Read-only	The name of the backup VPN virtual server for this VPN virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cltimeout	<Double>	Read-only	The idle time, if any, in seconds after which the client connection is terminated.
somethod	<String>	Read-only	VPN client applications are allocated from a block of intranet IP addresses. That block may be exhausted after a certain number of connections. This switch specifies the method used to determine whether or not a new connection will spill over, or exhaust, the allocated block of intranet IP addresses for that application. Possible values are CONNECTION or DYNAMICCONNECTION. CONNECTION means that a static integer value is the hard limit for the spillover threshold. The spillover threshold is described below. DYNAMICCONNECTION means that the spillover threshold is set according to the maximum number of connections defined for the VPN virtual server. Possible values = CONNECTION, DYNAMICCONNECTION, BANDWIDTH, HEALTH, NONE
sothreshold	<Double>	Read-only	VPN client applications are allocated from a block of intranet IP addresses. That block may be exhausted after a certain number of connections. The value of this option is the number of client connections after which the mapped IP address is used as the client source IP address instead of an address from the allocated block of intranet IP addresses.
sopersistence	<String>	Read-only	Whether or not cookie-based site persistance is enabled for this VPN vserver. Possible values are 'ConnectionProxy', HTTPRedirect, or NONE. Possible values = ENABLED, DISABLED
sopersistencetimeout	<Double>	Read-only	The timeout, if any, for cookie-based site persistance of this VPN vserver.
usemip	<String>	Read-only	Deprecated. See 'map' below. Possible values = ENABLED, DISABLED
map	<String>	Read-only	Whether or not mapped IP addresses are ON or OFF. Mapped IP addresses are source IP addresses for the virtual servers running on the NetScaler. Mapped IP addresses are used by the system to connect to the backend servers. Possible values = ON, OFF
bindpoint	<String>	Read-only	Bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
gotopriorityexpression	<String>	Read-only	Next priority expression.
disableprimaryondown	<String>	Read-only	Tells whether traffic will continue reaching backup virtual servers even after the primary virtual server comes UP from DOWN state. Possible values = ENABLED, DISABLED
			Binds the authentication policy as the secondary policy to use in a two-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
groupextraction	<Boolean>	Read-only	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
epaprofileoptional	<Boolean>	Read-only	Mark the EPA profile optional for preauthentication EPA profile. User would be shown a logon page even if the EPA profile fails to evaluate.
ngname	<String>	Read-only	Node group devno to which this authentication virtual sever belongs.
csvserver	<String>	Read-only	Name of the CS vserver to which the VPN vserver is bound.
response	<String>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | ENABLE | DISABLE | RENAME | CHECK | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnvserver":{
    "name":<String_value>,
    "servicetype":<String_value>,
    "ipv46":<String_value>,
    "range":<Double_value>,
    "port":<Integer_value>,
    "state":<String_value>,
    "authentication":<String_value>,
    "doublehop":<String_value>,
    "maxaaausers":<Double_value>,
    "icaonly":<String_value>,
    "icaproxysessionmigration":<String_value>,
    "dtls":<String_value>,
    "loginonce":<String_value>,
    "advancedepa":<String_value>,
    "devicecert":<String_value>,
    "certkeynames":<String_value>,
    "downstateflush":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "netprofile":<String_value>,
    "cginfrahomepageredirect":<String_value>,
    "maxloginattempts":<Double_value>,
    "failedlogintimeout":<Double_value>,
    "l2conn":<String_value>,
    "deploymenttype":<String_value>,
    "rdpserverprofilename":<String_value>,
    "windowsepapluginupgrade":<String_value>,
    "linuxepapluginupgrade":<String_value>,
    "macepapluginupgrade":<String_value>,
    "userdomains":<String_value>,
    "authnprofile":<String_value>
}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnvserver":{
    "name":<String_value>,
    "ipv46":<String_value>,
    "authentication":<String_value>,
    "doublehop":<String_value>,
    "icaonly":<String_value>,
    "icaproxysessionmigration":<String_value>,
    "dtls":<String_value>,
    "loginonce":<String_value>,
    "advancedepa":<String_value>,
    "devicecert":<String_value>,
    "certkeynames":<String_value>,
    "maxaaausers":<Double_value>,
    "downstateflush":<String_value>,
    "listenpolicy":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "netprofile":<String_value>,
    "cginfrahomepageredirect":<String_value>,
    "maxloginattempts":<Double_value>,
    "rdpserverprofilename":<String_value>,
    "failedlogintimeout":<Double_value>,
    "l2conn":<String_value>,
    "windowsepapluginupgrade":<String_value>,
    "macepapluginupgrade":<String_value>,
    "linuxepapluginupgrade":<String_value>,
    "userdomains":<String_value>,
    "authnprofile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnvserver":{
    "name":<String_value>,
    "authentication":true,
    "doublehop":true,
    "icaonly":true,
    "icaproxysessionmigration":true,
    "dtls":true,
    "loginonce":true,
    "advancedepa":true,
    "devicecert":true,
    "certkeynames":true,
    "maxaaausers":true,
    "downstateflush":true,
    "listenpolicy":true,
    "listenpriority":true,
    "tcpprofilename":true,
    "httpprofilename":true,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "comment":true,
    "appflowlog":true,
    "icmpvsrresponse":true,
    "rhistate":true,
    "netprofile":true,
    "cginfrahomepageredirect":true,
    "maxloginattempts":true,
    "rdpserverprofilename":true,
    "l2conn":true,
    "windowsepapluginupgrade":true,
    "macepapluginupgrade":true,
    "linuxepapluginupgrade":true,
    "userdomains":true,
    "authnprofile":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

enable

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver?**action=enable**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

disable

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver?**action=disable**

HTTP Method: POST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

rename

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver?**action=rename**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnvserver":{
    "name":<String_value>,
    "newname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

check

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver?**action=check**

HTTP Method: POST

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnvserver":{
    "name":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "vpnvserver": [ {
    "name":<String_value>,
    "response":<String_value>

}]}
```

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnvserver resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver": [ {
    "name":<String_value>,
    "ip":<String_value>,
    "ipv46":<String_value>,
    "value":<String_value>,
    "port":<Integer_value>,
    "range":<Double_value>,
    "servicetype":<String_value>,
    "type":<String_value>,
    "curstate":<String_value>,
    "status":<Integer_value>,
    "cachetype":<String_value>,
    "redirect":<String_value>,
    "precedence":<String_value>,
    "redirecturl":<String_value>,
    "authentication":<String_value>,
    "doublehop":<String_value>,
    "icaonly":<String_value>,
    "icaproxysessionmigration":<String_value>,
    "dtls":<String_value>,
    "loginonce":<String_value>,
    "advancedepa":<String_value>,
    "devicecert":<String_value>,
    "certkeynames":<String_value>,
    "maxaaausers":<Double_value>,
    "curaaausers":<Double_value>,
    "curtotalusers":<Double_value>,
    "domain":<String_value>,
    "rule":<String_value>,
    "policyname":<String_value>,
    "policy":<String_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "cachevserver":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "backupvserver":<String_value>,
    "priority":<Double_value>,
    "clttimeout":<Double_value>,
    "somethod":<String_value>,
    "sothreshold":<Double_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "usemip":<String_value>,
    "map":<String_value>,
    "downstateflush":<String_value>,
    "bindpoint":<String_value>,
    "gotopriorityexpression":<String_value>,
    "disableprimaryondown":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "netprofile":<String_value>,
    "cginfrahomepageredirect":<String_value>,
    "maxloginattempts":<Double_value>,
    "failedlogintimeout":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "deploymenttype":<String_value>,
    "windowsepapluginupgrade":<String_value>,
    "linuxepapluginupgrade":<String_value>,
    "macepapluginupgrade":<String_value>,
    "epaprofileoptional":<Boolean_value>,
    "rdpserverprofilename":<String_value>,
    "ngname":<String_value>,
    "state":<String_value>,
    "l2conn":<String_value>,
    "userdomains":<String_value>,
    "csvserver":<String_value>,
    "authnprofile":<String_value>
}]}
```


get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver": [ {
    "name":<String_value>,
    "ip":<String_value>,
    "ipv46":<String_value>,
    "value":<String_value>,
    "port":<Integer_value>,
    "range":<Double_value>,
    "servicetype":<String_value>,
    "type":<String_value>,
    "curstate":<String_value>,
    "status":<Integer_value>,
    "cachetype":<String_value>,
    "redirect":<String_value>,
    "precedence":<String_value>,
    "redirecturl":<String_value>,
    "authentication":<String_value>,
    "doublehop":<String_value>,
    "icaonly":<String_value>,
    "icaproxysessionmigration":<String_value>,
    "dtls":<String_value>,
    "loginonce":<String_value>,
    "advancedepa":<String_value>,
    "devicecert":<String_value>,
    "certkeynames":<String_value>,
    "maxaaausers":<Double_value>,
    "curaaausers":<Double_value>,
    "curtotalusers":<Double_value>,
    "domain":<String_value>,
    "rule":<String_value>,
    "policyname":<String_value>,
    "policy":<String_value>,
    "servicename":<String_value>,
    "weight":<Double_value>,
    "cachevserver":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "backupvserver":<String_value>,
    "priority":<Double_value>,
    "clttimeout":<Double_value>,
    "somethod":<String_value>,
    "sothreshold":<Double_value>,
    "sopersistence":<String_value>,
    "sopersistencetimeout":<Double_value>,
    "usemip":<String_value>,
    "map":<String_value>,
    "downstateflush":<String_value>,
    "bindpoint":<String_value>,
    "gotopriorityexpression":<String_value>,
    "disableprimaryondown":<String_value>,
    "listenpolicy":<String_value>,
    "listenpriority":<Double_value>,
    "tcpprofilename":<String_value>,
    "httpprofilename":<String_value>,
    "comment":<String_value>,
    "appflowlog":<String_value>,
    "icmpvsrresponse":<String_value>,
    "rhistate":<String_value>,
    "netprofile":<String_value>,
    "cginfrahomepageredirect":<String_value>,
    "maxloginattempts":<Double_value>,
    "failedlogintimeout":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "deploymenttype":<String_value>,
    "windowsepapluginupgrade":<String_value>,
    "linuxepapluginupgrade":<String_value>,
    "macepapluginupgrade":<String_value>,
    "epaprofileoptional":<Boolean_value>,
    "rdpserverprofilename":<String_value>,
    "ngname":<String_value>,
    "state":<String_value>,
    "l2conn":<String_value>,
    "userdomains":<String_value>,
    "csvserver":<String_value>,
    "authnprofile":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnvserver": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_binding

Jul 07, 2017

Binding object which returns the resources bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the NetScaler Gateway virtual server for which to show detailed information. Minimum length = 1
vpnvserver_intranetip6_binding	<vpnvserver_intranetip6_binding[]>	Read-only	intranetip6 that can be bound to vpnvserver.
vpnvserver_authenticationradiuspolicy_binding	<vpnvserver_authenticationradiuspolicy_binding[]>	Read-only	authenticationradiuspolicy that can be bound to vpnvserver.
vpnvserver_vpnsessionpolicy_binding	<vpnvserver_vpnsessionpolicy_binding[]>	Read-only	vpnsessionpolicy that can be bound to vpnvserver.
vpnvserver_vpnnexthopserver_binding	<vpnvserver_vpnnexthopserver_binding[]>	Read-only	vpnnexthopserver that can be bound to vpnvserver.
vpnvserver_authenticationldappolicy_binding	<vpnvserver_authenticationldappolicy_binding[]>	Read-only	authenticationldappolicy that can be bound to vpnvserver.
vpnvserver_authenticationsamlidppolicy_binding	<vpnvserver_authenticationsamlidppolicy_binding[]>	Read-only	authenticationsamlidppolicy that can be bound to vpnvserver.
vpnvserver_authenticationwebauthpolicy_binding	<vpnvserver_authenticationwebauthpolicy_binding[]>	Read-only	authenticationwebauthpolicy that can be bound to vpnvserver.
vpnvserver_sharefileserver_binding	<vpnvserver_sharefileserver_binding[]>	Read-only	sharefileserver that can be bound to vpnvserver.
vpnvserver_authenticationlocalpolicy_binding	<vpnvserver_authenticationlocalpolicy_binding[]>	Read-only	authenticationlocalpolicy that can be bound to vpnvserver.
vpnvserver_vpnintranetapplication_binding	<vpnvserver_vpnintranetapplication_binding[]>	Read-only	vpnintranetapplication that can be bound to vpnvserver.
vpnvserver_appcontroller_binding	<vpnvserver_appcontroller_binding[]>	Read-only	appcontroller that can be bound to vpnvserver.
vpnvserver_rewritepolicy_binding	<vpnvserver_rewritepolicy_binding[]>	Read-only	rewritepolicy that can be bound to vpnvserver.
vpnvserver_vpnportaltheme_binding	<vpnvserver_vpnportaltheme_binding[]>	Read-only	vpnportaltheme that can be bound to vpnvserver.
vpnvserver_cspolicy_binding	<vpnvserver_cspolicy_binding[]>	Read-only	cspolicy that can be bound to vpnvserver.
vpnvserver_auditsyslogpolicy_binding	<vpnvserver_auditsyslogpolicy_binding[]>	Read-only	auditsyslogpolicy that can be bound to vpnvserver.
vpnvserver_authenticationcertpolicy_binding	<vpnvserver_authenticationcertpolicy_binding[]>	Read-only	authenticationcertpolicy that can be bound to vpnvserver.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
vpnvserver_vpnepaprofile_binding	<vpnvserver_vpnepaprofile_binding[]>	Read-only	vpnepaprofile that can be bound to vpnvserver.
vpnvserver_vpnurl_binding	<vpnvserver_vpnurl_binding[]>	Read-only	vpnurl that can be bound to vpnvserver.
vpnvserver_auditnslogpolicy_binding	<vpnvserver_auditnslogpolicy_binding[]>	Read-only	auditnslogpolicy that can be bound to vpnvserver.
vpnvserver_authenticationpolicy_binding	<vpnvserver_authenticationpolicy_binding[]>	Read-only	authenticationpolicy that can be bound to vpnvserver.
vpnvserver_vpneula_binding	<vpnvserver_vpneula_binding[]>	Read-only	vpneula that can be bound to vpnvserver.
vpnvserver_icapolicy_binding	<vpnvserver_icapolicy_binding[]>	Read-only	icapolicy that can be bound to vpnvserver.
vpnvserver_vpnclientlessaccesspolicy_binding	<vpnvserver_vpnclientlessaccesspolicy_binding[]>	Read-only	vpnclientlessaccesspolicy that can be bound to vpnvserver.
vpnvserver_intranetip_binding	<vpnvserver_intranetip_binding[]>	Read-only	intranetip that can be bound to vpnvserver.
vpnvserver_appflowpolicy_binding	<vpnvserver_appflowpolicy_binding[]>	Read-only	appflowpolicy that can be bound to vpnvserver.
vpnvserver_responderpolicy_binding	<vpnvserver_responderpolicy_binding[]>	Read-only	responderpolicy that can be bound to vpnvserver.
vpnvserver_vpntrafficpolicy_binding	<vpnvserver_vpntrafficpolicy_binding[]>	Read-only	vpntrafficpolicy that can be bound to vpnvserver.
vpnvserver_authenticationnegotiatepolicy_binding	<vpnvserver_authenticationnegotiatepolicy_binding[]>	Read-only	authenticationnegotiatepolicy that can be bound to vpnvserver.
vpnvserver_aaapreauthenticationpolicy_binding	<vpnvserver_aaapreauthenticationpolicy_binding[]>	Read-only	aaapreauthenticationpolicy that can be bound to vpnvserver.
vpnvserver_feopolicy_binding	<vpnvserver_feopolicy_binding[]>	Read-only	feopolicy that can be bound to vpnvserver.
vpnvserver_authenticationtacacspolicy_binding	<vpnvserver_authenticationtacacspolicy_binding[]>	Read-only	authenticationtacacspolicy that can be bound to vpnvserver.
vpnvserver_cachepolicy_binding	<vpnvserver_cachepolicy_binding[]>	Read-only	cachepolicy that can be bound to vpnvserver.
vpnvserver_authenticationsamlpolicy_binding	<vpnvserver_authenticationsamlpolicy_binding[]>	Read-only	authenticationsamlpolicy that can be bound to vpnvserver.
vpnvserver_authenticationdfapolicy_binding	<vpnvserver_authenticationdfapolicy_binding[]>	Read-only	authenticationdfapolicy that can be bound to vpnvserver.
vpnvserver_staserver_binding	<vpnvserver_staserver_binding[]>	Read-only	staserver that can be bound to vpnvserver.
vpnvserver_authenticationloginschemapolicy_binding	<vpnvserver_authenticationloginschemapolicy_binding[]>	Read-only	authenticationloginschemapolicy that can be bound to vpnvserver.

Operations (click to see Properties)

GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_binding/name_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Payload:

```
{ "vpnvserver_binding": [ {
    "name":<String_value>,
    "vpnvserver_authenticationloginschemapolicy_binding":<vpnvserver_authenticationloginschemapolicy_binding[]_value>,
    "vpnvserver_authenticationsamlpolicy_binding":<vpnvserver_authenticationsamlpolicy_binding[]_value>,
    "vpnvserver_cspolicy_binding":<vpnvserver_cspolicy_binding[]_value>,
    "vpnvserver_icapolicy_binding":<vpnvserver_icapolicy_binding[]_value>,
    "vpnvserver_vpnportaltheme_binding":<vpnvserver_vpnportaltheme_binding[]_value>,
    "vpnvserver_appflowpolicy_binding":<vpnvserver_appflowpolicy_binding[]_value>,
    "vpnvserver_intranetip_binding":<vpnvserver_intranetip_binding[]_value>,
    "vpnvserver_vpntrafficpolicy_binding":<vpnvserver_vpntrafficpolicy_binding[]_value>,
    "vpnvserver_vpnepaprofile_binding":<vpnvserver_vpnepaprofile_binding[]_value>,
    "vpnvserver_authenticationtacacspolicy_binding":<vpnvserver_authenticationtacacspolicy_binding[]_value>,
    "vpnvserver_vpnintranetapplication_binding":<vpnvserver_vpnintranetapplication_binding[]_value>,
    "vpnvserver_authenticationradiuspolicy_binding":<vpnvserver_authenticationradiuspolicy_binding[]_value>,
    "vpnvserver_authenticationpolicy_binding":<vpnvserver_authenticationpolicy_binding[]_value>,
    "vpnvserver_aaapreauthenticationpolicy_binding":<vpnvserver_aaapreauthenticationpolicy_binding[]_value>,
    "vpnvserver_feopolicy_binding":<vpnvserver_feopolicy_binding[]_value>,
    "vpnvserver_authenticationlocalpolicy_binding":<vpnvserver_authenticationlocalpolicy_binding[]_value>,
    "vpnvserver_intranetip6_binding":<vpnvserver_intranetip6_binding[]_value>,
    "vpnvserver_vpnsessionpolicy_binding":<vpnvserver_vpnsessionpolicy_binding[]_value>,
    "vpnvserver_staserver_binding":<vpnvserver_staserver_binding[]_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "vpnvserver_authenticationcertpolicy_binding":<vpnvserver_authenticationcertpolicy_binding[]_value>,
    "vpnvserver_authenticationsamlidppolicy_binding":<vpnvserver_authenticationsamlidppolicy_binding[]_value>,
    "vpnvserver_vpnnexthopserver_binding":<vpnvserver_vpnnexthopserver_binding[]_value>,
    "vpnvserver_auditnslogpolicy_binding":<vpnvserver_auditnslogpolicy_binding[]_value>,
    "vpnvserver_auditsyslogpolicy_binding":<vpnvserver_auditsyslogpolicy_binding[]_value>,
    "vpnvserver_sharefileserver_binding":<vpnvserver_sharefileserver_binding[]_value>,
    "vpnvserver_responderpolicy_binding":<vpnvserver_responderpolicy_binding[]_value>,
    "vpnvserver_vpnurl_binding":<vpnvserver_vpnurl_binding[]_value>,
    "vpnvserver_cachepolicy_binding":<vpnvserver_cachepolicy_binding[]_value>,
    "vpnvserver_authenticationldappolicy_binding":<vpnvserver_authenticationldappolicy_binding[]_value>,
    "vpnvserver_authenticationwebauthpolicy_binding":<vpnvserver_authenticationwebauthpolicy_binding[]_value>,
    "vpnvserver_rewritepolicy_binding":<vpnvserver_rewritepolicy_binding[]_value>,
    "vpnvserver_authenticationnegotiatepolicy_binding":<vpnvserver_authenticationnegotiatepolicy_binding[]_value>,
    "vpnvserver_appcontroller_binding":<vpnvserver_appcontroller_binding[]_value>,
    "vpnvserver_vpnclientlessaccesspolicy_binding":<vpnvserver_vpnclientlessaccesspolicy_binding[]_value>,
    "vpnvserver_authenticationdfapolicy_binding":<vpnvserver_authenticationdfapolicy_binding[]_value>,
    "vpnvserver_vpneula_binding":<vpnvserver_vpneula_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_vpneula_binding

May 20, 2016

Binding object showing the vpneula that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
eula	<String>	Read-write	Name of the EULA bound to VPN vserver.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_vpneula_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_vpneula_binding":{
    "name":<String_value>,
    "eula":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpneula_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_vpneula_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_vpneula_binding": [ {
    "eula":<String_value>,
    "name":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpneula_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_vpneula_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_aaapreauthenticationpolicy_binding

May 20, 2016

Binding object showing the aaapreauthenticationpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

acttype **Name**	<Double> **Data Type**	Read-only **Permissions**	. **Description**
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_aaapreauthenticationpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_aaapreauthenticationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_aaapreauthenticationpolicy_binding resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_aaapreauthenticationpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_aaapreauthenticationpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_aaapreauthenticationpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_appcontroller_binding

May 20, 2016

Binding object showing the appcontroller that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
appcontroller	<String>	Read-write	Configured App Controller server in XenMobile deployment.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_appcontroller_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_appcontroller_binding":{
    "name":<String_value>,
    "appcontroller":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_appcontroller_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_appcontroller_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_appcontroller_binding": [ {
    "appcontroller":<String_value>,
    "name":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_appcontroller_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_appcontroller_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_appflowpolicy_binding

May 20, 2016

Binding object showing the appflowpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_appflowpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_appflowpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_appflowpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_appflowpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_appflowpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_appflowpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_appflowpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_auditnslogpolicy_binding

May 20, 2016

Binding object showing the auditnslogpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_auditnslogpolicy_binding

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_auditnslogpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnvserver_auditnslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_auditnslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_auditnslogpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_auditnslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_auditnslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_auditsyslogpolicy_binding

May 20, 2016

Binding object showing the auditsyslogpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_auditsyslogpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnvserver_auditsyslogpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_auditsyslogpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_auditsyslogpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_auditsyslogpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_auditsyslogpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnepaprofile

May 20, 2016

Configuration for Epa profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	name of device profile. Minimum length = 1
filename	<String>	Read-write	filename of the deviceprofile data xml. Minimum length = 1
data	<String>	Read-write	deviceprofile data xml. Minimum length = 1
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"vpnepaprofile":{
    "name":<String_value>,
    "filename":<String_value>,
    "data":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of vpnepaprofile resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnepaprofile resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnepaprofile": [ {
      "name":<String_value>,
      "data":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile/name_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile/name_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile/name_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnepaprofile": [ {
      "name":<String_value>,
      "data":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnepaprofile?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "vpnepaprofile": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationcertpolicy_binding

May 20, 2016

Binding object showing the authenticationcertpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
			Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
Name	**Data Type**	**Permissions**	**Description**
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
"vpnvserver_authenticationcertpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationcertpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationcertpolicy_binding resources in chunks.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationcertpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationcertpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"vpnvserver_authenticationcertpolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationdfapolicy_binding

May 20, 2016

Binding object showing the authenticationdfapolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
			Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{
"vpnvserver_authenticationdfapolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationdfapolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get the vpnvserver_authenticationdfapolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationdfapolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationdfapolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

{"vpnvserver_authenticationdfapolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationldappolicy_binding

Jul 07, 2017

Binding object showing the authenticationldappolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
			Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
Name	**Data Type**	**Permissions**	**Description**
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationldappolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
"vpnvserver_authenticationldappolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationldappolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationldappolicy_binding resources in chunks.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationldappolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationldappolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"vpnvserver_authenticationldappolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationlocalpolicy_binding

May 20, 2016

Binding object showing the authenticationlocalpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
Name	**Data Type**	**Permissions**	**Description**
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
"vpnvserver_authenticationlocalpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationlocalpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationlocalpolicy_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationlocalpolicy_binding": [ {
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationlocalpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"vpnvserver_authenticationlocalpolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationloginschemapolicy_binding

May 20, 2016

Binding object showing the authenticationloginschemapolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual

© 1999-2017 Citrix Systems, Inc. All rights reserved.

operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_authenticationloginschemapolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationloginschemapolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationloginschemapolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationloginschemapolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationloginschemapolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_authenticationloginschemapolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationnegotiatepolicy_binding

May 20, 2016

Binding object showing the authenticationnegotiatepolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationnegotiatepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationnegotiatepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationnegotiatepolicy_binding": [ {
      "priority":<Double_value>,
      "policy":<String_value>,
      "name":<String_value>,
      "secondary":<Boolean_value>,
      "acttype":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationnegotiatepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_authenticationnegotiatepolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationpolicy_binding

May 20, 2016

Binding object showing the authenticationpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
bindpoint	<String>	Read-write	you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_authenticationpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_authenticationpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationradiuspolicy_binding

Jul 07, 2017

Binding object showing the authenticationradiuspolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

acttype	<Double>	Read-only	.
Name	**Data Type**	**Permissions**	**Description**
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_authenticationradiuspolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationradiuspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationradiuspolicy_binding resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationradiuspolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationradiuspolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_authenticationradiuspolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationsamlidppolicy_binding

Jul 07, 2017

Binding object showing the authenticationsamlidppolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_authenticationsamlidppolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationsamlidppolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationsamlidppolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationsamlidppolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationsamlidppolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_authenticationsamlidppolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationsamlpolicy_binding

Jul 07, 2017

Binding object showing the authenticationsamlpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
Name	**Data Type**	**Permissions**	**Description**
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
"vpnvserver_authenticationsamlpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationsamlpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationsamlpolicy_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationsamlpolicy_binding": [ {
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationsamlpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

{"vpnvserver_authenticationsamlpolicy_binding": [{ "__count": "#no"}] }

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationtacacspolicy_binding

May 20, 2016

Binding object showing the authenticationtacacspolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

acttype **Name**	<Double> **Data Type**	Read-only **Permissions**	. **Description**
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_authenticationtacacspolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationtacacspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationtacacspolicy_binding resources in chunks.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationtacacspolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationtacacspolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_authenticationtacacspolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_authenticationwebauthpolicy_binding

May 20, 2016

Binding object showing the authenticationwebauthpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Read only	Description
Name	<Double>	Read only	Description
Name			

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_authenticationwebauthpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_authenticationwebauthpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_authenticationwebauthpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_authenticationwebauthpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_authenticationwebauthpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_authenticationwebauthpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_cachepolicy_binding

May 20, 2016

Binding object showing the cachepolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_cachepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_cachepolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_cachepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_cachepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_cachepolicy_binding": [ {
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_cachepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_cachepolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_cspolicy_binding

May 20, 2016

Binding object showing the cspolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_cspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_cspolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_cspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_cspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_cspolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_cspolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_cspolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_feopolicy_binding

May 20, 2016

Binding object showing the feopolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	Name of a policy to bind to the virtual server (for example, the name of an authentication, session, or endpoint analysis policy). Minimum length = 1
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_feopolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_feopolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_feopolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_feopolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_feopolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_feopolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_feopolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_icapolicy_binding

Jul 07, 2017

Binding object showing the icapolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_icapolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_icapolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_icapolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_icapolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_icapolicy_binding": [ {
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_icapolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_icapolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_intranetip6_binding

May 20, 2016

Binding object showing the intranetip6 that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
intranetip6	<String>	Read-write	The network id for the range of intranet IP6 addresses or individual intranet ip to be bound to the vserver.
numaddr	<Double>	Read-write	The number of ipv6 addresses.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_intranetip6_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_intranetip6_binding":{
    "name":<String_value>,
    "intranetip6":<String_value>,
    "numaddr":<Double_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_intranetip6_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_intranetip6_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_intranetip6_binding": [ {
    "intranetip6":<String_value>,
    "numaddr":<Double_value>,
    "name":<String_value>,
    "acttype":<Double_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_intranetip6_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_intranetip6_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_intranetip_binding

May 20, 2016

Binding object showing the intranetip that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
intranetip	<String>	Read-write	The network ID for the range of intranet IP addresses or individual intranet IP addresses to be bound to the virtual server.
netmask	<String>	Read-write	The netmask of the intranet IP address or range.
map	<String>	Read-only	Whether or not mapped IP addresses are ON or OFF. Mapped IP addresses are source IP addresses for the virtual servers running on the NetScaler. Mapped IP addresses are used by the system to connect to the backend servers. Possible values = ON, OFF
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_intranetip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_intranetip_binding":{
    "name":<String_value>,
    "intranetip":<String_value>,
    "netmask":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_intranetip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_intranetip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_intranetip_binding": [ {
    "name":<String_value>,
    "intranetip":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "netmask":<String_value>,
    "map":<String_value>,
    "acttype":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_intranetip_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_intranetip_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_responderpolicy_binding

Jul 07, 2017

Binding object showing the responderpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_responderpolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_responderpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_responderpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_responderpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_responderpolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_responderpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_responderpolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_rewritepolicy_binding

May 20, 2016

Binding object showing the rewritepolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_rewritepolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_rewritepolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_rewritepolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_rewritepolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_rewritepolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_rewritepolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_rewritepolicy_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_sharefileserver_binding

Jul 07, 2017

Binding object showing the sharefileserver that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
sharefile	<String>	Read-write	Configured ShareFile server in XenMobile deployment. Format IP:PORT / FQDN:PORT.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_sharefileserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_sharefileserver_binding":{
    "name":<String_value>,
    "sharefile":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_sharefileserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_sharefileserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_sharefileserver_binding": [ {
    "name":<String_value>,
    "sharefile":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_sharefileserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_sharefileserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_staserver_binding

Jul 07, 2017

Binding object showing the staserver that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
staaddresstype	<String>	Read-write	Type of the STA server address(ipv4/v6). Possible values = IPV4, IPV6
staserver	<String>	Read-write	Configured Secure Ticketing Authority (STA) server.
acttype	<Double>	Read-only	.
staauthid	<String>	Read-only	Authority ID of the STA Server. Authority ID is used to match incoming STA tickets in the SOCKS/CGP protocol with the right STA server.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_staserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_staserver_binding":{
    "name":<String_value>,
    "staserver":<String_value>,
    "staaddresstype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_staserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_staserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_staserver_binding": [ {
    "name":<String_value>,
    "staaddresstype":<String_value>,
    "staserver":<String_value>,
    "acttype":<Double_value>,
    "staauthid":<String_value>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_staserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_staserver_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_vpnclientlessaccesspolicy_binding

May 20, 2016

Binding object showing the vpnclientlessaccesspolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Next priority expression.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same for both authentication methods, but they can require different passwords.
bindpoint	<String>	Read-write	Bindpoint to which the policy is bound. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_vpnclientlessaccesspolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnclientlessaccesspolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_vpnclientlessaccesspolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "vpnvserver_vpnclientlessaccesspolicy_binding": [ {
    "priority":<Double_value>,
    "gotopriorityexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnclientlessaccesspolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"vpnvserver_vpnclientlessaccesspolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_vpnepaprofile_binding

May 20, 2016

Binding object showing the vpnepaprofile that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
epaprofile	<String>	Read-write	Advanced EPA profile to bind.
epaprofileoptional	<Boolean>	Read-write	Mark the EPA profile optional for preauthentication EPA profile. User would be shown a logon page even if the EPA profile fails to evaluate.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_vpnepaprofile_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_vpnepaprofile_binding":{
    "name":<String_value>,
    "epaprofile":<String_value>,
    "epaprofileoptional":<Boolean_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnepaprofile_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_vpnepaprofile_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_vpnepaprofile_binding": [ {
    "name":<String_value>,
    "epaprofile":<String_value>,
    "epaprofileoptional":<Boolean_value>,
    "acttype":<Double_value>
}]}
```

count

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnepaprofile_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_vpnepaprofile_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_vpnintranetapplication_binding

May 20, 2016

Binding object showing the vpnintranetapplication that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
intranetapplication	<String>	Read-write	The intranet VPN application.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_vpnintranetapplication_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_vpnintranetapplication_binding":{
    "name":<String_value>,
    "intranetapplication":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnintranetapplication_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_vpnintranetapplication_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_vpnintranetapplication_binding": [ {
    "name":<String_value>,
    "intranetapplication":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnintranetapplication_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_vpnintranetapplication_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_vpnnexthopserver_binding

May 20, 2016

Binding object showing the vpnnexthopserver that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
nexthopserver	<String>	Read-write	The name of the next hop server bound to the VPN virtual server.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_vpnnexthopserver_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_vpnnexthopserver_binding":{
    "name":<String_value>,
    "nexthopserver":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnnexthopserver_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_vpnnexthopserver_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_vpnnexthopserver_binding": [ {
    "name":<String_value>,
    "nexthopserver":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnnexthopserver_binding/name_value<String>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_vpnnexthopserver_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_vpnportaltheme_binding

Jul 07, 2017

Binding object showing the vpnportaltheme that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
portaltheme	<String>	Read-write	Name of the portal theme bound to VPN vserver.
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_vpnportaltheme_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_vpnportaltheme_binding":{
    "name":<String_value>,
    "portaltheme":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnportaltheme_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_vpnportaltheme_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_vpnportaltheme_binding": [ {
    "name":<String_value>,
    "portaltheme":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnportaltheme_binding/name_value<String>?**count=yes**

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_vpnportaltheme_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_vpnsessionpolicy_binding

Jul 07, 2017

Binding object showing the vpnsessionpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_vpnsessionpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnvserver_vpnsessionpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_vpnsessionpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_vpnsessionpolicy_binding": [ {
    "priority":<Double_value>,
    "gotoprioryexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnsessionpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_vpnsessionpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_vpntrafficpolicy_binding

May 20, 2016

Binding object showing the vpntrafficpolicy that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
priority	<Double>	Read-write	The priority, if any, of the VPN virtual server policy.
gotopriorityexpression	<String>	Read-write	Expression or other value specifying the next policy to evaluate if the current policy evaluates to TRUE. Specify one of the following values: * NEXT - Evaluate the policy with the next higher priority number. * END - End policy evaluation. * USE_INVOCATION_RESULT - Applicable if this policy invokes another policy label. If the final goto in the invoked policy label has a value of END, the evaluation stops. If the final goto is anything other than END, the current policy label performs a NEXT. * A default syntax or classic expression that evaluates to a number. If you specify an expression, the number to which it evaluates determines the next policy to evaluate, as follows: * If the expression evaluates to a higher numbered priority, the policy with that priority is evaluated next. * If the expression evaluates to the priority of the current policy, the policy with the next higher numbered priority is evaluated next. * If the expression evaluates to a number that is larger than the largest numbered priority, policy evaluation ends. An UNDEF event is triggered if: * The expression is invalid. * The expression evaluates to a priority number that is numerically lower than the current policy's priority. * The expression evaluates to a priority number that is between the current policy's priority number (say, 30) and the highest priority number (say, 100), but does not match any configured priority number (for example, the expression evaluates to the number 85). This example assumes that the priority number increments by 10 for every successive policy, and therefore a priority number of 85 does not exist in the policy label.
policy	<String>	Read-write	The name of the policy, if any, bound to the VPN virtual server.
groupextraction	<Boolean>	Read-write	Binds the authentication policy to a tertiary chain which will be used only for group extraction. The user will not authenticate against this server, and this will only be called if primary and/or secondary authentication has succeeded.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
secondary	<Boolean>	Read-write	Binds the authentication policy as the secondary policy to use in a two-factor configuration. A user must then authenticate not only via a primary authentication method but also via a secondary authentication method. User groups are aggregated across both. The user name must be exactly the same

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
bindpoint	<String>	Read-write	Bind point to which to bind the policy. Applies only to rewrite and cache policies. If you do not set this parameter, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression. Possible values = REQUEST, RESPONSE, ICA_REQUEST, OTHERTCP_REQUEST
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding

HTTP Method: PUT

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_vpntrafficpolicy_binding":{
    "name":<String_value>,
    "policy":<String_value>,
    "priority":<Double_value>,
    "secondary":<Boolean_value>,
    "groupextraction":<Boolean_value>,
    "gotopriorityexpression":<String_value>,
    "bindpoint":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>?**filter=property-**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

name1:property-value1,property-name2:property-value2

Use this query-parameter to get the filtered set of vpnvserver_vpntrafficpolicy_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_vpntrafficpolicy_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_vpntrafficpolicy_binding": [ {
    "priority":<Double_value>,
    "gotoprioritiyexpression":<String_value>,
    "policy":<String_value>,
    "groupextraction":<Boolean_value>,
    "name":<String_value>,
    "secondary":<Boolean_value>,
    "bindpoint":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpntrafficpolicy_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_vpntrafficpolicy_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver_vpnurl_binding

May 20, 2016

Binding object showing the vpnurl that can be bound to vpnvserver.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
urlname	<String>	Read-write	The intranet URL.
name	<String>	Read-write	Name of the virtual server. Minimum length = 1
acttype	<Double>	Read-only	.
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/vpnvserver_vpnurl_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"vpnvserver_vpnurl_binding":{
    "name":<String_value>,
    "urlname":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of vpnvserver_vpnurl_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the vpnvserver_vpnurl_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver_vpnurl_binding": [ {
    "urlname":<String_value>,
    "name":<String_value>,
    "acttype":<Double_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/vpnvserver_vpnurl_binding/name_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{"vpnvserver_vpnurl_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Wf

May 20, 2016

Web Front configuration.

Name	Description
wfpackage	Configuration for Web Front
wfsite	Configuration for WF site

© 1999-2017 Citrix Systems, Inc. All rights reserved.

wfpackage

May 20, 2016

Configuration for Web Front resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
jre	<String>	Read-write	Complete path to the JRE tar file. You can use OpenJDK7 package for FreeBSD 8.x/amd64.The Java package can be downloaded from http://ftp.freebsd.org/pub/FreeBSD/releases/amd64/amd64/8.4-RELEASE/packages/java/openjdk-7.17.02_2.tbz or http://www.freebsdfoundation.org/cgi-bin/download?download=diablo-jdk-freebsd6.amd64.1.7.17.07.02.tbz. Minimum length = 1 Maximum length = 255
wf	<String>	Read-write	Complete path to the WebFront tar file for installing the WebFront on the NetScaler appliance. This file includes Apache Tomcat Web server. The file name has the following format: nswf-<version number>.tar (for example, nswf-1.5.tar). Minimum length = 1 Maximum length = 255

Operations (click to see Properties)

INSTALL | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Install

URL: http://<netscaler-ip-address>/nitro/v1/config/wfpackage?**action=Install**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"wfpackage":{
    "jre":<String_value>,
    "wf":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/wfpackage

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "wfpackage": [ {
      "jre":<String_value>,
      "wf":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

wfsite

May 20, 2016

Configuration for WF site resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sitename	<String>	Read-write	Name of the WebFront site being created on the NetScaler appliance. Minimum length = 1 Maximum length = 255
storefronturl	<String>	Read-write	FQDN or IP of Windows StoreFront server where the store is configured. Minimum length = 1 Maximum length = 255
storename	<String>	Read-write	Name of the store present in StoreFront. Minimum length = 1 Maximum length = 255
html5receiver	<String>	Read-write	Specifies whether or not to use HTML5 receiver for launching apps for the WF site. Default value: FALLBACK Possible values = ALWAYS, FALLBACK, OFF
workspacecontrol	<String>	Read-write	Specifies whether of not to use workspace control for the WF site. Default value: ON Possible values = ON, OFF
displayroamingaccounts	<String>	Read-write	Specifies whether or not to display the accounts selection screen during First Time Use of Receiver . Default value: ON Possible values = ON, OFF
xframeoptions	<String>	Read-write	The value to be sent in the X-Frame-Options header. WARNING: Setting this option to ALLOW could leave the end user vulnerable to Click Jacking attacks. Default value: DENY Possible values = ALLOW, DENY
state	<String>	Read-only	State of wf site. Default value: UP Possible values = UP, INITIALIZING, DOWN-HostUnknown, DOWN-ReqTimeout, DOWN-Wrong Store, DOWN-SF Error, DOWN-SSL Error, DOWN-Conn Reset, DOWN
__count	<Double>	Read-only	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

Name	Data Type	Permissions	Description

ADD | DELETE | UPDATE | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/wfsite

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"wfsite":{
    "sitename":<String_value>,
    "storefronturl":<String_value>,
    "storename":<String_value>,
    "html5receiver":<String_value>,
    "workspacecontrol":<String_value>,
    "displayroamingaccounts":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "xframeoptions":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/wfsite/sitename_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/wfsite

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"wfsite":{
    "sitename":<String_value>,
    "storefronturl":<String_value>,
    "storename":<String_value>,
    "html5receiver":<String_value>,
    "workspacecontrol":<String_value>,
    "displayroamingaccounts":<String_value>,
    "xframeoptions":<String_value>
}}
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/wfsite

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/wfsite?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/wfsite?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of wfsite resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/wfsite?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/wfsite?**pagesize=#no&pageno=#no**

Use this query-parameter to get the wfsite resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "wfsite": [ {
    "sitename":<String_value>,
    "storefronturl":<String_value>,
    "storename":<String_value>,
    "html5receiver":<String_value>,
    "workspacecontrol":<String_value>,
    "displayroamingaccounts":<String_value>,
    "xframeoptions":<String_value>,
    "state":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/wfsite/sitename_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/wfsite/sitename_value<String>?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/wfsite/sitename_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:

```
{ "wfsite": [ {
    "sitename":<String_value>,
    "storefronturl":<String_value>,
    "storename":<String_value>,
    "html5receiver":<String_value>,
    "workspacecontrol":<String_value>,
    "displayroamingaccounts":<String_value>,
    "xframeoptions":<String_value>,
    "state":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/wfsite?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "wfsite": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

WebInterface

May 20, 2016

Web Interface configuration.

Name	Description
wipackage	Configuration for Web Interface
wisite	Configuration for WI site
wisite_accessmethod_binding	Binding object showing the accessmethod that can be bound to wisite
wisite_binding	Binding object showing the resources that can be bound to wisite
wisite_farmname_binding	Binding object showing the farmname that can be bound to wisite
wisite_translationinternalip_binding	Binding object showing the translationinternalip that can be bound to wisite

© 1999-2017 Citrix Systems, Inc. All rights reserved.

wipackage

May 20, 2016

Configuration for Web Interface resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
jre	<String>	Read-write	Complete path to the JRE tar file. You can use OpenJDK7 package for FreeBSD 8.x/amd64.The Java package can be downloaded from http://ftp.freebsd.org/pub/FreeBSD/releases/amd64/amd64/8.4-RELEASE/packages/java/openjdk-7.17.02_2.tbz. Default value: "file://tmp/diablo-jdk-freebsd6.amd64.1.6.0.07.02.tbz" Minimum length = 1 Maximum length = 255
wi	<String>	Read-write	Complete path to the Web Interface tar file for installing the Web Interface on the NetScaler appliance. This file includes Apache Tomcat Web server. The file name has the following format: nswi-<version number>.tgz (for example, nswi-1.5.tgz). Default value: "http://citrix.com/downloads/nswi-1.7.tgz" Minimum length = 1 Maximum length = 255
maxsites	<String>	Read-write	Maximum number of Web Interface sites that can be created on the NetScaler appliance; changes the amount of RAM reserved for Web Interface usage; changing its value results in restart of Tomcat server and invalidates any existing Web Interface sessions. Possible values = 3, 25, 50, 100, 200, 500

Operations (click to see Properties)

INSTALL | GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

Install

URL: http://<netscaler-ip-address>/nitro/v1/config/wipackage?**action=Install**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"wipackage":{
    "jre":<String_value>,
    "wi":<String_value>,
    "maxsites":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/wipackage

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "wipackage": [ {
      "jre":<String_value>,
      "wi":<String_value>,
      "maxsites":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

wisite

May 20, 2016

Configuration for WI site resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sitepath	<String>	Read-write	Path to the Web Interface site being created on the NetScaler appliance. Minimum length = 1 Maximum length = 250
agurl	<String>	Read-write	Call back URL of the Gateway. Minimum length = 1 Maximum length = 255
staurl	<String>	Read-write	URL of the Secure Ticket Authority (STA) server. Minimum length = 1 Maximum length = 255
secondstaurl	<String>	Read-write	URL of the second Secure Ticket Authority (STA) server. Minimum length = 1 Maximum length = 255
sessionreliability	<String>	Read-write	Enable session reliability through Access Gateway. Default value: OFF Possible values = ON, OFF
usetwotickets	<String>	Read-write	Request tickets issued by two separate Secure Ticket Authorities (STA) when a resource is accessed. Default value: OFF Possible values = ON, OFF
authenticationpoint	<String>	Read-write	Authentication point for the Web Interface site. Possible values = WebInterface, AccessGateway
agauthenticationmethod	<String>	Read-write	Method for authenticating a Web Interface site if you have specified Web Interface as the authentication point. Available settings function as follows: * Explicit - Users must provide a user name and password to log on to the Web Interface. * Anonymous - Users can log on to the Web Interface without providing a user name and password. They have access to resources published for anonymous users. Possible values = Explicit, SmartCard
			The method of authentication to be used at Web Interface.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			Possible values = Explicit, Anonymous
defaultcustomtextlocale	<String>	Read-write	Default language for the Web Interface site. Default value: English Possible values = German, English, Spanish, French, Japanese, Korean, Russian, Chinese_simplified, Chinese_traditional
websessiontimeout	<Double>	Read-write	Time-out, in minutes, for idle Web Interface browser sessions. If a client's session is idle for a time that exceeds the time-out value, the NetScaler appliance terminates the connection. Default value: 20 Minimum value = 1 Maximum value = 1440
defaultaccessmethod	<String>	Read-write	Default access method for clients accessing the Web Interface site. Note: Before you configure an access method based on the client IP address, you must enable USIP mode on the Web Interface service to make the client?s IP address available with the Web Interface. Depending on whether the Web Interface site is configured to use an HTTP or HTTPS virtual server or to use access gateway, you can send clients or access gateway the IP address, or the alternate address, of a XenApp or XenDesktop server. Or, you can send the IP address translated from a mapping entry, which defines mapping of an internal address and port to an external address and port. Note: In the NetScaler command line, mapping entries can be created by using the bind wi site command. Possible values = Direct, Alternate, Translated, GatewayDirect, GatewayAlternate, GatewayTranslated
logintitle	<String>	Read-write	A custom login page title for the Web Interface site. Default value: "Welcome to Web Interface on NetScaler" Minimum length = 1 Maximum length = 255
appwelcomemessage	<String>	Read-write	Specifies localized text to appear at the top of the main content area of the Applications screen. LanguageCode is en, de, es, fr, ja, or any other supported language identifier. Minimum length = 1 Maximum length = 255
welcomemessage	<String>	Read-write	Localized welcome message that appears on the welcome area of the login screen. Minimum length = 1 Maximum length = 255
footertext	<String>	Read-write	Localized text that appears in the footer area of all pages. Minimum length = 1 Maximum length = 255
loginsysmessage	<String>	Read-write	Localized text that appears at the bottom of the main content area of the login screen. Minimum length = 1 Maximum length = 255

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
preloginbutton	<String>	Read-write	Localized text that appears as the name of the pre-login message confirmation button. Minimum length = 1 Maximum length = 255
preloginmessage	<String>	Read-write	Localized text that appears on the pre-login message page. Minimum length = 1 Maximum length = 2048
prelogintitle	<String>	Read-write	Localized text that appears as the title of the pre-login message page. Minimum length = 1 Maximum length = 255
domainselection	<String>	Read-write	Domain names listed on the login screen for explicit authentication. Minimum length = 1 Maximum length = 255
sitetype	<String>	Read-write	Type of access to the Web Interface site. Available settings function as follows: * XenApp/XenDesktop web site - Configures the Web Interface site for access by a web browser. * XenApp/XenDesktop services site - Configures the Web Interface site for access by the XenApp plug-in. Default value: XenAppWeb Possible values = XenAppWeb, XenAppServices
userinterfacebranding	<String>	Read-write	Specifies whether the site is focused towards users accessing applications or desktops. Setting the parameter to Desktops changes the functionality of the site to improve the experience for XenDesktop users. Citrix recommends using this setting for any deployment that includes XenDesktop. Default value: Applications Possible values = Desktops, Applications
publishedresourcetype	<String>	Read-write	Method for accessing the published XenApp and XenDesktop resources. Available settings function as follows: * Online - Allows applications to be launched on the XenApp and XenDesktop servers. * Offline - Allows streaming of applications to the client. * DualMode - Allows both online and offline modes. Default value: Online Possible values = Online, Offline, DualMode
kioskmode	<String>	Read-write	User settings do not persist from one session to another. Default value: OFF Possible values = ON, OFF
showsearch	<String>	Read-write	Enables search option on XenApp websites. Default value: OFF Possible values = ON, OFF
showrefresh	<String>	Read-write	Provides the Refresh button on the applications screen. Default value: OFF Possible values = ON, OFF

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
wiuserinterfacemodes	<String>	Read-write	* Simple - Only the login fields for the selected authentication method are displayed. * Advanced - Displays the navigation bar, which provides access to the pre-login messages and preferences screens. Default value: SIMPLE Possible values = SIMPLE, ADVANCED
userinterfacelayouts	<String>	Read-write	Specifies whether or not to use the compact user interface. Default value: AUTO Possible values = AUTO, NORMAL, COMPACT
restrictdomains	<String>	Read-write	The RestrictDomains setting is used to enable/disable domain restrictions. If domain restriction is enabled, the LoginDomains list is used for validating the login domain. It is applied to all the authentication methods except Anonymous for XenApp Web and XenApp Services sites. Default value: OFF Possible values = ON, OFF
logindomains	<String>	Read-write	List of NetBIOS domain names], Domain names to use for access restriction. Only takes effect when used in conjunction with the RestrictDomains setting. Minimum length = 1 Maximum length = 255
hidedomainfield	<String>	Read-write	The HideDomainField setting is used to control whether the domain field is displayed on the logon screen. Default value: OFF Possible values = ON, OFF
agcallbackurl	<String>	Read-write	Callback AGURL to which Web Interface contacts. . Minimum length = 1 Maximum length = 255
__count	<Double>	Read-only	count parameter

Operations (click to see Properties)

ADD | DELETE | UPDATE | UNSET | GET (ALL) | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"wisite":{
    "sitepath":<String_value>,
    "agurl":<String_value>,
    "staurl":<String_value>,
    "secondstaurl":<String_value>,
    "sessionreliability":<String_value>,
    "usetwotickets":<String_value>,
    "authenticationpoint":<String_value>,
    "agauthenticationmethod":<String_value>,
    "wiauthenticationmethods":<String[]_value>,
    "defaultcustomtextlocale":<String_value>,
    "websessiontimeout":<Double_value>,
    "defaultaccessmethod":<String_value>,
    "logintitle":<String_value>,
    "appwelcomemessage":<String_value>,
    "welcomemessage":<String_value>,
    "footertext":<String_value>,
    "loginsysmessage":<String_value>,
    "preloginbutton":<String_value>,
    "preloginmessage":<String_value>,
    "prelogintitle":<String_value>,
    "domainselection":<String_value>,
    "sitetype":<String_value>,
    "userinterfacebranding":<String_value>,
    "publishedresourcetype":<String_value>,
    "kioskmode":<String_value>,
    "showsearch":<String_value>,
    "showrefresh":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "wiuserinterfacemodes":<String_value>,
    "userinterfacelayouts":<String_value>,
    "restrictdomains":<String_value>,
    "logindomains":<String_value>,
    "hidedomainfield":<String_value>,
    "agcallbackurl":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite/sitepath_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

update

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"wisite":{
    "sitepath":<String_value>,
    "agurl":<String_value>,
    "staurl":<String_value>,
    "sessionreliability":<String_value>,
    "usetwotickets":<String_value>,
    "secondstaurl":<String_value>,
    "wiauthenticationmethods":<String[]_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "defaultaccessmethod":<String_value>,
    "defaultcustomtextlocale":<String_value>,
    "websessiontimeout":<Double_value>,
    "logintitle":<String_value>,
    "appwelcomemessage":<String_value>,
    "welcomemessage":<String_value>,
    "footertext":<String_value>,
    "loginsysmessage":<String_value>,
    "preloginbutton":<String_value>,
    "preloginmessage":<String_value>,
    "prelogintitle":<String_value>,
    "domainselection":<String_value>,
    "userinterfacebranding":<String_value>,
    "authenticationpoint":<String_value>,
    "agauthenticationmethod":<String_value>,
    "publishedresourcetype":<String_value>,
    "kioskmode":<String_value>,
    "showsearch":<String_value>,
    "showrefresh":<String_value>,
    "wiuserinterfacemodes":<String_value>,
    "userinterfacelayouts":<String_value>,
    "restrictdomains":<String_value>,
    "logindomains":<String_value>,
    "hidedomainfield":<String_value>,
    "agcallbackurl":<String_value>
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

unset

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite?**action=unset**

HTTP Method: POST

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{"wisite":{
    "sitepath":<String_value>,
    "appwelcomemessage":true,
    "welcomemessage":true,
    "footertext":true,
    "loginsysmessage":true,
    "preloginbutton":true,
    "preloginmessage":true,
    "prelogintitle":true,
    "userinterfacebranding":true,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "logindomains":true
}}
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/wisite?**attrs=property-name1,property-name2**

Use this query parameter to specify the resource details that you want to retrieve.

filter

http://<netscaler-ip-address>/nitro/v1/config/wisite?**filter=property-name1:property-val1,property-name2:property-val2**

Use this query-parameter to get the filtered set of wisite resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

view

http://<netscaler-ip-address>/nitro/v1/config/wisite?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

pagination

http://<netscaler-ip-address>/nitro/v1/config/wisite?**pagesize=#no&pageno=#no**

Use this query-parameter to get the wisite resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "wisite": [ {
    "sitepath":<String_value>,
    "agurl":<String_value>,
    "staurl":<String_value>,
    "wiauthenticationmethods":<String[]_value>,
    "logintitle":<String_value>,
    "appwelcomemessage":<String_value>,
    "welcomemessage":<String_value>,
    "footertext":<String_value>,
    "loginsysmessage":<String_value>,
    "preloginbutton":<String_value>,
    "preloginmessage":<String_value>,
    "prelogintitle":<String_value>,
    "domainselection":<String_value>,
    "defaultcustomtextlocale":<String_value>,
    "websessiontimeout":<Double_value>,
    "sitetype":<String_value>,
    "userinterfacebranding":<String_value>,
    "showsearch":<String_value>,
    "showrefresh":<String_value>,
    "wiuserinterfacemodes":<String_value>,
    "userinterfacelayouts":<String_value>,
    "publishedresourcetype":<String_value>,
    "defaultaccessmethod":<String_value>,
    "agauthenticationmethod":<String_value>,
    "sessionreliability":<String_value>,
    "usetwotickets":<String_value>,
    "secondstaurl":<String_value>,
    "authenticationpoint":<String_value>,
    "kioskmode":<String_value>,
    "restrictdomains":<String_value>,
    "logindomains":<String_value>,
    "hidedomainfield":<String_value>,
    "agcallbackurl":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite/sitepath_value<String>

Query-parameters:

attrs

http://<netscaler-ip-address>/nitro/v1/config/wisite/sitepath_value<String>?**attrs=property-name1,property-name2**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query parameter to specify the resource details that you want to retrieve.

view

http://<netscaler-ip-address>/nitro/v1/config/wisite/sitepath_value<String>?**view=summary**

Note: By default, the retrieved results are displayed in detail view (?view=detail).

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "wisite": [ {
    "sitepath":<String_value>,
    "agurl":<String_value>,
    "staurl":<String_value>,
    "wiauthenticationmethods":<String[]_value>,
    "logintitle":<String_value>,
    "appwelcomemessage":<String_value>,
    "welcomemessage":<String_value>,
    "footertext":<String_value>,
    "loginsysmessage":<String_value>,
    "preloginbutton":<String_value>,
    "preloginmessage":<String_value>,
    "prelogintitle":<String_value>,
    "domainselection":<String_value>,
    "defaultcustomtextlocale":<String_value>,
    "websessiontimeout":<Double_value>,
    "sitetype":<String_value>,
    "userinterfacebranding":<String_value>,
    "showsearch":<String_value>,
    "showrefresh":<String_value>,
    "wiuserinterfacemodes":<String_value>,
    "userinterfacelayouts":<String_value>,
    "publishedresourcetype":<String_value>,
    "defaultaccessmethod":<String_value>,
    "agauthenticationmethod":<String_value>,
    "sessionreliability":<String_value>,
    "usetwotickets":<String_value>,
    "secondstaurl":<String_value>,
    "authenticationpoint":<String_value>,
    "kioskmode":<String_value>,
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "restrictdomains":<String_value>,
    "logindomains":<String_value>,
    "hidedomainfield":<String_value>,
    "agcallbackurl":<String_value>
}]}
```

count

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "wisite": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

wisite_accessmethod_binding

May 20, 2016

Binding object showing the accessmethod that can be bound to wisite.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clientipaddress	<String>	Read-write	IPv4 or network address of the client for which you want to associate an access method. Default value: 0
clientnetmask	<String>	Read-write	Subnet mask associated with the IPv4 or network address specified by the Client IP Address parameter. Default value: 0
accessmethod	<String>	Read-write	Secure access method to be applied to the IPv4 or network address of the client specified by the Client IP Address parameter. Depending on whether the Web Interface site is configured to use an HTTP or HTTPS virtual server or to use access gateway, you can send clients or access gateway the IP address, or the alternate address, of a XenApp or XenDesktop server. Or, you can send the IP address translated from a mapping entry, which defines mapping of an internal address and port to an external address and port. Possible values = Direct, Alternate, Translated, GatewayDirect, GatewayAlternate, GatewayTranslated
sitepath	<String>	Read-write	Path to the Web Interface site. Minimum length = 1 Maximum length = 250
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/wisite_accessmethod_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"wisite_accessmethod_binding":{
    "sitepath":<String_value>,
    "accessmethod":<String_value>,
    "clientipaddress":<String_value>,
    "clientnetmask":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>

HTTP Method: DELETE

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of wisite_accessmethod_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the wisite_accessmethod_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "wisite_accessmethod_binding": [ {
    "clientipaddress":<String_value>,
    "clientnetmask":<String_value>,
    "accessmethod":<String_value>,
    "sitepath":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite_accessmethod_binding/sitepath_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"wisite_accessmethod_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

wisite_binding

May 20, 2016

Binding object which returns the resources bound to wisite.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sitepath	<String>	Read-write	Path of a Web Interface site whose details you want the NetScaler appliance to display. Minimum length = 1 Maximum length = 250
wisite_translationinternalip_binding	<wisite_translationinternalip_binding[]>	Read-only	translationinternalip that can be bound to wisite.
wisite_accessmethod_binding	<wisite_accessmethod_binding[]>	Read-only	accessmethod that can be bound to wisite.
wisite_farmname_binding	<wisite_farmname_binding[]>	Read-only	farmname that can be bound to wisite.

Operations (click to see Properties)

GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite_binding/sitepath_value<String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Payload:

```
{ "wisite_binding": [ {
      "sitepath":<String_value>,
      "wisite_translationinternalip_binding":<wisite_translationinternalip_binding[]_value>,
      "wisite_accessmethod_binding":<wisite_accessmethod_binding[]_value>,
      "wisite_farmname_binding":<wisite_farmname_binding[]_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

wisite_farmname_binding

Jul 07, 2017

Binding object showing the farmname that can be bound to wisite.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sitepath	<String>	Read-write	Path to the Web Interface site. Minimum length = 1 Maximum length = 250
groups	<String>	Read-write	Active Directory groups that are permitted to enumerate resources from server farms. Including a setting for this parameter activates the user roaming feature. A maximum of 512 user groups can be specified for each farm defined with the Farm<n> parameter. The groups must be comma separated.
xmlport	<Double>	Read-write	Port number at which to contact the XML service.
transport	<String>	Read-write	Transport protocol to use for transferring data, related to the Web Interface site, between the NetScaler appliance and the XML service. Possible values = HTTP, HTTPS, SSLRELAY
sslrelayport	<Double>	Read-write	TCP port at which the XenApp or XenDesktop servers listenfor SSL Relay traffic from the NetScaler appliance. This parameter is required if you have set SSL Relay as the transport protocol. Web Interface uses root certificates when authenticating a server running SSL Relay. Make sure that all the servers running SSL Relay are configured to listen on the same port.
farmname	<String>	Read-write	Name for the logical representation of a XenApp or XenDesktop farm to be bound to the Web Interface site. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters.
loadbalance	<String>	Read-write	Use all the XML servers (load balancing mode) or only one server (failover mode). Possible values = ON, OFF
recoveryfarm	<String>	Read-write	Binded farm is set as a recovery farm. Default value: OFF Possible values = ON, OFF
xmlserveraddresses	<String>	Read-write	Comma-separated IP addresses or host names of XenApp or XenDesktop servers providing XML services.
__count	<Double>	Read-write	count parameter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/wisite_farmname_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"wisite_farmname_binding":{
    "sitepath":<String_value>,
    "farmname":<String_value>,
    "xmlserveraddresses":<String_value>,
    "groups":<String_value>,
    "recoveryfarm":<String_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "xmlport":<Double_value>,
    "transport":<String_value>,
    "sslrelayport":<Double_value>,
    "loadbalance":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

---

get

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of wisite_farmname_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the wisite_farmname_binding resources in chunks.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "wisite_farmname_binding": [ {
    "sitepath":<String_value>,
    "groups":<String_value>,
    "xmlport":<Double_value>,
    "transport":<String_value>,
    "sslrelayport":<Double_value>,
    "farmname":<String_value>,
    "loadbalance":<String_value>,
    "recoveryfarm":<String_value>,
    "xmlserveraddresses":<String_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite_farmname_binding/sitepath_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Payload:


```
{"wisite_farmname_binding": [ { "__count": "#no"} ] }
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

wisite_translationinternalip_binding

May 20, 2016

Binding object showing the translationinternalip that can be bound to wisite.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sitepath	<String>	Read-write	Path to the Web Interface site. Minimum length = 1 Maximum length = 250
accesstype	<String>	Read-write	Type of access to the XenApp or XenDesktop server. Available settings function as follows: * User Device - Clients can use the translated address of the mapping entry to connect to the XenApp or XenDesktop server. * Gateway - Access Gateway can use the translated address of the mapping entry to connect to the XenApp or XenDesktop server. * User Device and Gateway - Both clients and Access Gateway can use the translated address of the mapping entry to connect to the XenApp or XenDesktop server. Default value: UserDevice Possible values = UserDevice, Gateway, UserDeviceAndGateway
translationexternalport	<Integer>	Read-write	External port number associated with the server's port number. Range 1 - 65535
translationinternalip	<String>	Read-write	IP address of the server for which you want to associate an external IP address. (Clients access the server through the associated external address and port.). Default value: 0
translationexternalip	<String>	Read-write	External IP address associated with server's IP address.
translationinternalport	<Integer>	Read-write	Port number of the server for which you want to associate an external port. (Clients access the server through the associated external address and port.). Range 1 - 65535
__count	<Double>	Read-write	count parameter

Operations (click to see Properties)

ADD: | DELETE: | GET | COUNT

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query

© 1999-2017 Citrix Systems, Inc. All rights reserved.

parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ## Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

add:

URL: http://<netscaler-ip-address/nitro/v1/config/wisite_translationinternalip_binding

HTTP Method: PUT

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Content-Type:application/json
```

Request Payload:

```
{
"wisite_translationinternalip_binding":{
    "sitepath":<String_value>,
    "translationinternalip":<String_value>,
    "translationinternalport":<Integer_value>,
    "translationexternalip":<String_value>,
    "translationexternalport":<Integer_value>,
    "accesstype":<String_value>
}}
```

Response:

HTTP Status Code on Success: 201 Created
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

delete:

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>

HTTP Method: DELETE

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

get

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>

Query-parameters:

filter

http://<netscaler-ip-address>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>?**filter=property-name1:property-value1,property-name2:property-value2**

Use this query-parameter to get the filtered set of wisite_translationinternalip_binding resources configured on NetScaler.Filtering can be done on any of the properties of the resource.

pagination

http://<netscaler-ip-address>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>?**pagesize=#no&pageno=#no**

Use this query-parameter to get the wisite_translationinternalip_binding resources in chunks.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "wisite_translationinternalip_binding": [ {
      "sitepath":<String_value>,
      "accesstype":<String_value>,
      "translationexternalport":<Integer_value>,
      "translationinternalip":<String_value>,
      "translationexternalip":<String_value>,
      "translationinternalport":<Integer_value>
}]}
```

---

count

URL: http://<netscaler-ip-address>/nitro/v1/config/wisite_translationinternalip_binding/sitepath_value<String>?**count=yes**

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors).
The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{"wisite_translationinternalip_binding": [ { "__count": "#no"} ] }
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Statistics

Jul 06, 2017

The statistics API is arranged in a logical groups of objects. Choose an appropriate group to find more details.

Name	Description
AAA	Authentication, authorization, and accounting service statistics.
Appflow	AppFlow statistics.
Application Firewall	Application Firewall statistics.
Appqoe	Application Level Quality of Experience statistics
Audit	Audit statistics.
Authentication	Authentication statistics.
Authorization	Authorization statistics. Authorization services check which resources users are authorized to access, and grant permissions accordingly.
Autoscale	AutoScale statistics.
Basic	Basic system statistics.
Ca	Content Accelerator statistics. The Content Accelerator feature is used to cache static and dynamic web application data.
Integrated Caching	Integrated Caching statistics. The Integrated Caching feature is used to cache static and dynamic web application data.
Cluster	Cluster commands.
Compression	Compression statistics. The system's feature for compressing HTTP responses to compression-aware browsers.
Cache Redirection	Cache redirection statistics. The cache redirection feature can transparently redirect cacheable HTTP requests to a cache and send non-cacheable or dynamic HTTP requests to the origin server. A cache stores or caches frequently requested web content and serves such web content to a client on behalf of the origin servers, alleviating the load on the origin server farm.
	Content Switching statistics. Content Switching feature that enables you to direct traffic to servers on the basis of

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Content Switching	
Name	**Description**
Domain Name Service	Domain Name Service(DNS) statistics.
HTTP DoS Protection	HTTP DOS protection statistics.
Front-end-optimization	Front end optimization statistics. The system's feature to optimize Web content for performance.
Global Server Load Balancing	Global Server Load Balancing (GSLB) statistics. GSLB feature ensures that client requests are directed to a best performing site available in a global enterprise and distributed Internet environment.
High Availability	High-availability commands.
Ica	ICA statistics.
Ipsec	IPSEC statistics.
Load Balancing	Load Balancing statistics. The load balancing methods manage the selection of an appropriate physical server in a server farm.
LLDP	Link Layer Discovery Protocol.
Lsn	Large Scale NAT commands
Mediaclassification	Media Classification feature is used to show the stats of different media types that are being served by the netscaler
Network	Networking commands.
NS	System/Global level statistics.
Priority Queuing	Priority Queuing statistics. Priority Queuing ensures that key customer web sites get the timely responses they need. System provides priority-based queuing, which provides different levels of service to customers.
Protocol	Protocol statistics.
Qos	QoS statistics.
Responder	Responder statistics.
Rewrite	Rewrite statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Description
Sure Connect	under heavy load for the original URL. The original content will be served to the client, automatically, after an estimated delay period.
SNMP	SNMP(Simple Network Management Protocol) statistics.
Spillover	Spillover policies and actions.
SSL	SSL statistics
Stream	Stream statistics.
System	System statistics.
Traffic Management	TM session/policy statistics.
Transform	URL Transformation statistics.
SSL VPN	Virtual Private Network statistics.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AAA

May 20, 2016

Authentication, authorization, and accounting service statistics.

Name	Description
aaa	Statistics for aaa.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

aaa

May 20, 2016

Statistics for aaa.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
aaaauthsuccess	<Double>	Read-only	Count of authentication successes.
aaaauthsuccessrate	<Double>	Read-only	Rate (/s) counter for aaaauthsuccess
aaaauthfail	<Double>	Read-only	Count of authentication failures.
aaaauthfailrate	<Double>	Read-only	Rate (/s) counter for aaaauthfail
aaaauthonlyhttpsuccess	<Double>	Read-only	Count of HTTP connections that succeeded authorization.
aaaauthonlyhttpsuccessrate	<Double>	Read-only	Rate (/s) counter for aaaauthonlyhttpsuccess
aaaauthonlyhttpfail	<Double>	Read-only	Count of HTTP connections that failed authorization.
aaaauthonlyhttpfailrate	<Double>	Read-only	Rate (/s) counter for aaaauthonlyhttpfail
aaaauthnonhttpsuccess	<Double>	Read-only	Count of non HTTP connections that succeeded authorization.
aaaauthnonhttpsuccessrate	<Double>	Read-only	Rate (/s) counter for aaaauthnonhttpsuccess
aaaauthnonhttpfail	<Double>	Read-only	Count of non HTTP connections that failed authorization.
aaaauthnonhttpfailrate	<Double>	Read-only	Rate (/s) counter for aaaauthnonhttpfail
aaacursessions	<Double>	Read-only	Count of current AAA sessions.
aaacursessionsrate	<Double>	Read-only	Rate (/s) counter for aaacursessions
aaatotsessions	<Double>	Read-only	Count of all AAA sessions.
aaasessionsrate	<Double>	Read-only	Rate (/s) counter for aaatotsessions
aaatotsessiontimeout	<Double>	Read-only	Count of AAA sessions that have timed out.
aaasessiontimeoutrate	<Double>	Read-only	Rate (/s) counter for aaatotsessiontimeout
aaacuricasessions	<Double>	Read-only	Count of current ICA only sessions.
aaacuricasessionsrate	<Double>	Read-only	Rate (/s) counter for aaacuricasessions
aaacuricaonlyconn	<Double>	Read-only	Count of current ICA only connections.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
aaacuricaonlyconnrate	<Double>	Read-only	Rate (/s) counter for aaacuricaonlyconn
aaacuricaconn	<Double>	Read-only	Count of current ICA connections.
aaacuricaconnrate	<Double>	Read-only	Rate (/s) counter for aaacuricaconn
aaacurtmsessions	<Double>	Read-only	Count of current AAATM sessions.
aaacurtmsessionsrate	<Double>	Read-only	Rate (/s) counter for aaacurtmsessions
aaatottmsessions	<Double>	Read-only	Count of all AAATM sessions.
aaatmsessionsrate	<Double>	Read-only	Rate (/s) counter for aaatottmsessions

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/aaa

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/aaa?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get aaa resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "aaa": [ {
    "aaatotsessiontimeout":<Double_value>,
    "aaaauthfail":<Double_value>,
    "aaaauthsuccess":<Double_value>,
    "aaatottmsessions":<Double_value>,
    "aaaauthonlyhttpfailrate":<Double_value>,
    "aaaauthnonhttpsuccess":<Double_value>,
    "aaacuricaonlyconnrate":<Double_value>,
    "aaaauthonlyhttpsuccessrate":<Double_value>,
    "aaacurtmsessions":<Double_value>,
    "aaacuricasessions":<Double_value>,
    "aaatotsessions":<Double_value>,
    "aaaauthonlyhttpfail":<Double_value>,
    "aaacuricaonlyconn":<Double_value>,
    "aaatmsessionsrate":<Double_value>,
    "aaaauthfailrate":<Double_value>,
    "aaacursessionsrate":<Double_value>,
    "aaaauthnonhttpsuccessrate":<Double_value>,
    "aaacuricaconn":<Double_value>,
    "aaaauthnonhttpfailrate":<Double_value>,
    "aaaauthnonhttpfail":<Double_value>,
    "aaacuricaconnrate":<Double_value>,
    "aaaauthsuccessrate":<Double_value>,
    "aaaauthonlyhttpsuccess":<Double_value>,
    "aaacurtmsessionsrate":<Double_value>,
    "aaasessiontimeoutrate":<Double_value>,
    "aaacuricasessionsrate":<Double_value>,
    "aaacursessions":<Double_value>,
    "aaasessionsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Appflow

Jul 07, 2017

AppFlow statistics.

Name	Description
appflow	Statistics for appflow.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appflow

May 20, 2016
Statistics for appflow.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
appflowtotaltxoctets	<Double>	Read-only	The total number of AppFlow (IPFIX) octets that the NetScaler has transmitted.
appflowtxoctetsrate	<Double>	Read-only	Rate (/s) counter for appflowtotaltxoctets
appflowtotalflows	<Double>	Read-only	The total number of AppFlow (IPFIX) flows that the NetScaler has transmitted.
appflowflowsrate	<Double>	Read-only	Rate (/s) counter for appflowtotalflows
appflowtotaltxmessagess	<Double>	Read-only	The total number of AppFlow (IPFIX) messages that the NetScaler has transmitted.
appflowtxmessagessrate	<Double>	Read-only	Rate (/s) counter for appflowtotaltxmessagess
appflowtotalignoctets	<Double>	Read-only	The total number of octets that the NetScaler has ignored for AppFlow (IPFIX).
appflowignoctetsrate	<Double>	Read-only	Rate (/s) counter for appflowtotalignoctets
appflowtotalignpacketss	<Double>	Read-only	The total number of packets that the NetScaler has ignored for AppFlow (IPFIX).
appflowignpacketssrate	<Double>	Read-only	Rate (/s) counter for appflowtotalignpacketss
appflowtotalnotxoctets	<Double>	Read-only	The total number of AppFlow (IPFIX) octets that the NetScaler has not transmitted.
appflownotxoctetsrate	<Double>	Read-only	Rate (/s) counter for appflowtotalnotxoctets
appflowtotalnotxflows	<Double>	Read-only	The total number of AppFlow (IPFIX) flows that the NetScaler has not transmitted.
appflownotxflowsrate	<Double>	Read-only	Rate (/s) counter for appflowtotalnotxflows
appflowtotalnotxpackets	<Double>	Read-only	The total number of AppFlow (IPFIX) packets that the NetScaler has not transmitted.
appflownotxpacketsrate	<Double>	Read-only	Rate (/s) counter for appflowtotalnotxpackets

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/appflow

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/appflow?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get appflow resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appflow": [ {
    "appflowtxmessagessrate":<Double_value>,
    "appflowtotalnotxoctets":<Double_value>,
    "appflowtotaltxmessagess":<Double_value>,
    "appflowtotalignpacketss":<Double_value>,
    "appflowtotalflows":<Double_value>,
    "appflowtotalnotxflows":<Double_value>,
    "appflownotxflowsrate":<Double_value>,
    "appflowtotalnotxpackets":<Double_value>,
    "appflownotxoctetsrate":<Double_value>,
    "appflowignpacketssrate":<Double_value>,
    "appflowtxoctetsrate":<Double_value>,
    "appflowignoctetsrate":<Double_value>,
    "appflowtotalignoctets":<Double_value>,
    "appflowtotaltxoctets":<Double_value>,
    "appflownotxpacketsrate":<Double_value>,
    "appflowflowsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Application Firewall

May 20, 2016

Application Firewall statistics.

Name	Description
appfw	Statistics for appfw.
appfwpolicy	Statistics for application firewall policy resource.
appfwpolicylabel	Statistics for application firewall policy label resource.
appfwprofile	Statistics for application firewall profile resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfw

May 20, 2016
Statistics for appfw.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
appfirewalltotallog	<Double>	Read-only	Total number of security check log messages generated by the Application Firewall.
appfirewalllograte	<Double>	Read-only	Rate (/s) counter for appfirewalltotallog
appfirewalltotalviol	<Double>	Read-only	Total number of security check violations seen by the Application Firewall.
appfirewallviolrate	<Double>	Read-only	Rate (/s) counter for appfirewalltotalviol
appfirewallshortavgresptime	<Double>	Read-only	Average backend response time in milliseconds over the last 7 seconds
appfirewalllongavgresptime	<Double>	Read-only	Average backend response time in milliseconds since reboot
appfirewallrequests	<Double>	Read-only	HTTP/HTTPS requests sent to your protected web servers via the Application Firewall.
appfirewallrequestsrate	<Double>	Read-only	Rate (/s) counter for appfirewallrequests
appfirewallreqbytes	<Double>	Read-only	Number of bytes transfered for requests
appfirewallreqbytesrate	<Double>	Read-only	Rate (/s) counter for appfirewallreqbytes
appfirewallresponses	<Double>	Read-only	HTTP/HTTPS responses sent by your protected web servers via the Application Firewall.
appfirewallresponsesrate	<Double>	Read-only	Rate (/s) counter for appfirewallresponses
appfirewallresbytes	<Double>	Read-only	Number of bytes transfered for responses
appfirewallresbytesrate	<Double>	Read-only	Rate (/s) counter for appfirewallresbytes
appfirewallaborts	<Double>	Read-only	Incomplete HTTP/HTTPS requests aborted by the client before the Application Firewall could finish processing them.
appfirewallabortsrate	<Double>	Read-only	Rate (/s) counter for appfirewallaborts
appfirewallredirects	<Double>	Read-only	HTTP/HTTPS requests redirected by the Application Firewall to a different Web page or web server. (HTTP 302)
appfirewallredirectsrate	<Double>	Read-only	Rate (/s) counter for appfirewallredirects
appfirewalltrapsdropped	<Double>	Read-only	AppFirewall SNMP traps dropped due to time limit.
appfirewallviolstarturl	<Double>	Read-only	Number of Start URL security check violations seen by the Application Firewall.
appfirewallviolstarturlrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolstarturl

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
appfirewallviolrefererheaderrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolrefererheader
appfirewallviolbufferoverflow	<Double>	Read-only	Number of Buffer Overflow security check violations seen by the Application Firewall.
appfirewallviolbufferoverflowrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolbufferoverflow
appfirewallviolcookie	<Double>	Read-only	Number of Cookie Consistency security check violations seen by the Application Firewall.
appfirewallviolcookierate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcookie
appfirewallviolcsrftag	<Double>	Read-only	Number of Cross Site Request Forgery form tag security check violations seen by the Application Firewall.
appfirewallviolcsrftagrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcsrftag
appfirewallviolxss	<Double>	Read-only	Number of HTML Cross-Site Scripting security check violations seen by the Application Firewall.
appfirewallviolxssrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxss
appfirewallviolsql	<Double>	Read-only	Number of HTML SQL Injection security check violations seen by the Application Firewall.
appfirewallviolsqlrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsql
appfirewallviolfieldformat	<Double>	Read-only	Number of Field Format security check violations seen by the Application Firewall.
appfirewallviolfieldformatrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolfieldformat
appfirewallviolfieldconsistency	<Double>	Read-only	Number of Field Consistency security check violations seen by the Application Firewall.
appfirewallviolfieldconsistencyrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolfieldconsistency
appfirewallviolcreditcard	<Double>	Read-only	Number of Credit Card security check violations seen by the Application Firewall.
appfirewallviolcreditcardrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcreditcard
appfirewallviolsafeobject	<Double>	Read-only	Number of Safe Object security check violations seen by the Application Firewall.
appfirewallviolsafeobjectrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsafeobject
appfirewallviolsignature	<Double>	Read-only	Number of Signature violations seen by the Application Firewall.
appfirewallviolsignaturerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsignature
appfirewallviolcontenttype	<Double>	Read-only	Number of Content type security check violations seen by the Application Firewall.
appfirewallviolcontenttyperate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcontenttype
appfirewallviolwellformednessviolations	<Double>	Read-only	Number of XML Format security check violations seen by the Application Firewall.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfirewallviolwellformednessviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolwellformednessviolations
Name	**Data Type**	**Permissions**	**Description**
appfirewallviolxdosviolations	<Double>	Read-only	Number of XML Denial-of-Service security check violations seen by the Application Firewall.
appfirewallviolxdosviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxdosviolations
appfirewallviolmsgvalviolations	<Double>	Read-only	Number of XML Message Validation security check violations seen by the Application Firewall.
appfirewallviolmsgvalviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolmsgvalviolations
appfirewallviolwsiviolations	<Double>	Read-only	Number of Web Services Interoperability (WS-I) security check violations seen by the Application Firewall.
appfirewallviolwsiviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolwsiviolations
appfirewallviolxmlsqlviolations	<Double>	Read-only	Number of XML SQL Injection security check violations seen by the Application Firewall.
appfirewallviolxmlsqlviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlsqlviolations
appfirewallviolxmlxssviolations	<Double>	Read-only	Number of XML Cross-Site Scripting (XSS) security check violations seen by the Application Firewall.
appfirewallviolxmlxssviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlxssviolations
appfirewallviolxmlattachmentviolations	<Double>	Read-only	Number of XML Attachment security check violations seen by the Application Firewall.
appfirewallviolxmlattachmentviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlattachmentviolations
appfirewallviolxmlsoapfaultviolations	<Double>	Read-only	Number of requests returning soap:fault from the backend server
appfirewallviolxmlsoapfaultviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlsoapfaultviolations
appfirewallviolxmlgenviolations	<Double>	Read-only	Number of requests returning XML generic error from the backend server
appfirewallviolxmlgenviolationsrate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlgenviolations
appfirewalllogstarturl	<Double>	Read-only	Number of Start URL security check log messages generated by the Application Firewall.
appfirewalllogstarturlrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogstarturl
appfirewalllogdenyurl	<Double>	Read-only	Number of Deny URL security check log messages generated by the Application Firewall.
appfirewalllogdenyurlrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogdenyurl
appfirewalllogrefererheader	<Double>	Read-only	Number of Referer Header security check log messages generated by the Application Firewall.
appfirewalllogrefererheaderrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogrefererheader
appfirewalllogbufferoverflow	<Double>	Read-only	Number of Buffer Overflow security check log messages generated by the Application Firewall.
appfirewalllogbufferoverflowrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogbufferoverflow
appfirewalllogcookie	<Double>	Read-only	Number of Cookie Consistency security check log messages generated by the Application Firewall.
appfirewalllogcookierate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcookie

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfirewalllogcsrftag	<Double>	Read-only	Number of Cross Site Request Forgery form tag security check log messages generated by the Application Firewall.
Name	**Data Type**	**Permissions**	**Description**
appfirewalllogcsrftagrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcsrftag
appfirewalllogxss	<Double>	Read-only	Number of HTML Cross-Site Scripting security check log messages generated by the Application Firewall.
appfirewalllogxssrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogxss
appfirewalllogtransformxss	<Double>	Read-only	Number of HTML Cross-Site Scripting security check transform log messages generated by the Application Firewall.
appfirewalllogtransformxssrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogtransformxss
appfirewalllogsql	<Double>	Read-only	Number of HTML SQL Injection security check log messages generated by the Application Firewall.
appfirewalllogsqlrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogsql
appfirewalllogtransformsql	<Double>	Read-only	Number of HTML SQL Injection security check transform log messages generated by the Application Firewall.
appfirewalllogtransformsqlrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogtransformsql
appfirewalllogfieldformat	<Double>	Read-only	Number of Field Format security check log messages generated by the Application Firewall.
appfirewalllogfieldformatrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogfieldformat
appfirewalllogfieldconsistency	<Double>	Read-only	Number of Field Consistency security check log messages generated by the Application Firewall.
appfirewalllogfieldconsistencyrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogfieldconsistency
appfirewalllogcreditcard	<Double>	Read-only	Number of Credit Card security check log messages generated by the Application Firewall.
appfirewalllogcreditcardrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcreditcard
appfirewalllogsafeobject	<Double>	Read-only	Number of Safe Object security check log messages generated by the Application Firewall.
appfirewalllogsafeobjectrate	<Double>	Read-only	Rate (/s) counter for appfirewalllogsafeobject
appfirewallsignaturelogs	<Double>	Read-only	Number of Signature logs generated by the Application Firewall.
appfirewallsignaturelogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallsignaturelogs
appfirewalllogcontenttype	<Double>	Read-only	Number of content type security check log messages generated by the Application Firewall.
appfirewalllogcontenttyperate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcontenttype
appfirewallwellformednesslogs	<Double>	Read-only	Number of XML Format security check log messages generated by the Application Firewall.
appfirewallwellformednesslogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallwellformednesslogs
appfirewallxdoslogs	<Double>	Read-only	Number of XML Denial-of-Service security check log messages generated by the Application Firewall.
appfirewallxdoslogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxdoslogs
appfirewallmsgvallogs	<Double>	Read-only	Number of XML Message Validation security check log messages generated by the Application Firewall.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
appfirewallmsgvallogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallmsgvallogs
appfirewallwsilogs	<Double>	Read-only	Number of Web Services Interoperability (WS-I) security check log messages generated by the Application Firewall.
appfirewallwsilogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallwsilogs
appfirewallxmlsqllogs	<Double>	Read-only	Number of XML SQL Injection security check log messages generated by the Application Firewall.
appfirewallxmlsqllogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlsqllogs
appfirewallxmlxsslogs	<Double>	Read-only	Number of XML Cross-Site Scripting (XSS) security check log messages generated by the Application Firewall.
appfirewallxmlxsslogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlxsslogs
appfirewallxmlattachmentlogs	<Double>	Read-only	Number of XML Attachment security check log messages generated by the Application Firewall.
appfirewallxmlattachmentlogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlattachmentlogs
appfirewallxmlsoapfaultlogs	<Double>	Read-only	Number of requests generating soap:fault log messages
appfirewallxmlsoapfaultlogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlsoapfaultlogs
appfirewallxmlgenlogs	<Double>	Read-only	Number of requests generating XML generic error log messages
appfirewallxmlgenlogsrate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlgenlogs
appfirewallret4xx	<Double>	Read-only	Number of requests returning HTTP 4xx from the backend server
appfirewallret4xxrate	<Double>	Read-only	Rate (/s) counter for appfirewallret4xx
appfirewallret5xx	<Double>	Read-only	Number of requests returning HTTP 5xx from the backend server
appfirewallret5xxrate	<Double>	Read-only	Rate (/s) counter for appfirewallret5xx

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/appfw

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/appfw?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get appfw resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfw": [ {
    "appfirewallredirectsrate":<Double_value>,
    "appfirewallviolcookierate":<Double_value>,
    "appfirewallsignaturelogs":<Double_value>,
    "appfirewalllogdenyurl":<Double_value>,
    "appfirewalllogsqlrate":<Double_value>,
    "appfirewallviolfieldconsistency":<Double_value>,
    "appfirewallviolbufferoverflowrate":<Double_value>,
    "appfirewalllogtransformsql":<Double_value>,
    "appfirewallviolxssrate":<Double_value>,
    "appfirewalllogtransformsqlrate":<Double_value>,
    "appfirewalltotalviol":<Double_value>,
    "appfirewallwellformednesslogs":<Double_value>,
    "appfirewallviolsafeobjectrate":<Double_value>,
    "appfirewallresponsesrate":<Double_value>,
    "appfirewallviolxmlxssviolations":<Double_value>,
    "appfirewallresponses":<Double_value>,
    "appfirewallviolsignature":<Double_value>,
    "appfirewalllogbufferoverflow":<Double_value>,
    "appfirewallviolxmlgenviolationsrate":<Double_value>,
    "appfirewallviolrate":<Double_value>,
    "appfirewallabortsrate":<Double_value>,
    "appfirewallviolxmlsoapfaultviolationsrate":<Double_value>,
    "appfirewallrequestsrate":<Double_value>,
    "appfirewallwsilogs":<Double_value>,
    "appfirewallredirects":<Double_value>,
    "appfirewallsignaturelogsrate":<Double_value>,
    "appfirewallxdoslogsrate":<Double_value>,
    "appfirewalllogtransformxss":<Double_value>,
    "appfirewallxmlattachmentlogsrate":<Double_value>,
    "appfirewalllogrefererheaderrate":<Double_value>,
    "appfirewalllogcookie":<Double_value>,
    "appfirewallviolcreditcardrate":<Double_value>,
    "appfirewallret5xx":<Double_value>,
    "appfirewallviolmsgvalviolations":<Double_value>,
    "appfirewalllongavgresptime":<Double_value>,
    "appfirewalllogstarturl":<Double_value>,
    "appfirewallviolxmlgenviolations":<Double_value>,
    "appfirewallviolxmlsoapfaultviolations":<Double_value>,
    "appfirewallviolsafeobject":<Double_value>,
    "appfirewalllogxssrate":<Double_value>,
    "appfirewallviolsql":<Double_value>,
    "appfirewallviolxdosviolationsrate":<Double_value>,
    "appfirewalllogcreditcardrate":<Double_value>,
    "appfirewalllogsql":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "appfirewalllogdenyurlrate":<Double_value>,
    "appfirewallxmlsoapfaultlogs":<Double_value>,
    "appfirewalllogcontenttyperate":<Double_value>,
    "appfirewallviolxmlattachmentviolationsrate":<Double_value>,
    "appfirewalllogbufferoverflowrate":<Double_value>,
    "appfirewallret5xxrate":<Double_value>,
    "appfirewalltotallog":<Double_value>,
    "appfirewalllogcookierate":<Double_value>,
    "appfirewalllogcsrftag":<Double_value>,
    "appfirewallviolxss":<Double_value>,
    "appfirewallviolwellformednessviolations":<Double_value>,
    "appfirewallreqbytes":<Double_value>,
    "appfirewallxmlattachmentlogs":<Double_value>,
    "appfirewalllogcsrftagrate":<Double_value>,
    "appfirewallviolmsgvalviolationsrate":<Double_value>,
    "appfirewallviolcontenttyperate":<Double_value>,
    "appfirewalllograte":<Double_value>,
    "appfirewallxmlsqllogsrate":<Double_value>,
    "appfirewallxmlgenlogs":<Double_value>,
    "appfirewallreqbytesrate":<Double_value>,
    "appfirewalllogsafeobject":<Double_value>,
    "appfirewallviolcreditcard":<Double_value>,
    "appfirewalllogrefererheader":<Double_value>,
    "appfirewallviolxmlsqlviolationsrate":<Double_value>,
    "appfirewallaborts":<Double_value>,
    "appfirewallviolcsrftagrate":<Double_value>,
    "appfirewalllogstarturlrate":<Double_value>,
    "appfirewallviolcookie":<Double_value>,
    "appfirewallxmlxsslogs":<Double_value>,
    "appfirewallmsgvallogs":<Double_value>,
    "appfirewallviolwsiviolations":<Double_value>,
    "appfirewallviolcsrftag":<Double_value>,
    "appfirewallrequests":<Double_value>,
    "appfirewallresbytes":<Double_value>,
    "appfirewallviolbufferoverflow":<Double_value>,
    "appfirewallviolsignaturerate":<Double_value>,
    "appfirewallviolxmlxssviolationsrate":<Double_value>,
    "appfirewallviolxdosviolations":<Double_value>,
    "appfirewalllogsafeobjectrate":<Double_value>,
    "appfirewallxdoslogs":<Double_value>,
    "appfirewallviolcontenttype":<Double_value>,
    "appfirewallviolfieldconsistencyrate":<Double_value>,
    "appfirewalllogtransformxssrate":<Double_value>,
    "appfirewalllogfieldformatrate":<Double_value>,
    "appfirewallwellformednesslogsrate":<Double_value>,
    "appfirewallviolwsiviolationsrate":<Double_value>,
    "appfirewalllogcreditcard":<Double_value>,
    "appfirewallviolwellformednessviolationsrate":<Double_value>,
    "appfirewallviolfieldformatrate":<Double_value>,
    "appfirewalllogfieldconsistencyrate":<Double_value>,
    "appfirewallviolldenyurl":<Double_value>,
    "appfirewallviolxmlattachmentviolations":<Double_value>,
    "appfirewallviolsqlrate":<Double_value>,
    "appfirewallviolstarturlrate":<Double_value>,
    "appfirewallviolxmlsqlviolations":<Double_value>,
    "appfirewallshortavgresptime":<Double_value>,
    "appfirewallxmlgenlogsrate":<Double_value>,
    "appfirewalllogfieldformat":<Double_value>,
    "appfirewallxmlxsslogsrate":<Double_value>,
    "appfirewallresbytesrate":<Double_value>,
    "appfirewallviolrefererheader":<Double_value>,
    "appfirewallviolrefererheaderrate":<Double_value>,
    "appfirewalllogxss":<Double_value>,
    "appfirewalltrapsdropped":<Double_value>,
    "appfirewallviolstarturl":<Double_value>,
    "appfirewallret4xxrate":<Double_value>,
    "appfirewallxmlsqllogs":<Double_value>,
    "appfirewallwsilogsrate":<Double_value>,
    "appfirewallxmlsoapfaultlogsrate":<Double_value>,
    "appfirewallviolldenyurlrate":<Double_value>,
    "appfirewalllogcontenttype":<Double_value>,
    "appfirewallmsgvallogsrate":<Double_value>,
    "appfirewallret4xx":<Double_value>,
    "appfirewallviolfieldformat":<Double_value>,
    "appfirewalllogfieldconsistency":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicy

May 20, 2016

Statistics for application firewall policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the application firewall policy. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/appfwpolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/appfwpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_**

Use this query-parameter to get appfwpolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/appfwpolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwpolicylabel

May 20, 2016

Statistics for application firewall policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the application firewall policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/appfwpolicylabel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/appfwpolicylabel?**args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats**

Use this query-parameter to get appfwpolicylabel resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicylabel": [ {
      "labelname":<String_value>,
      "pipolicylabelhitsrate":<Double_value>,
      "pipolicylabelhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/appfwpolicylabel/labelname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwpolicylabel": [ {
      "labelname":<String_value>,
      "pipolicylabelhitsrate":<Double_value>,
      "pipolicylabelhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appfwprofile

May 20, 2016

Statistics for application firewall profile resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the application firewall profile. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
appfirewallrequestsperprofile	<Double>	Read-only	HTTP/HTTPS requests sent to your protected web servers via the Application Firewall.
appfirewallrequestsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallrequestsperprofile
appfirewallreqbytesperprofile	<Double>	Read-only	Number of bytes transfered for requests
appfirewallreqbytesperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallreqbytesperprofile
appfirewallresponsesperprofile	<Double>	Read-only	HTTP/HTTPS responses sent by your protected web servers via the Application Firewall.
appfirewallresponsesperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallresponsesperprofile
appfirewallresbytesperprofile	<Double>	Read-only	Number of bytes transfered for responses
appfirewallresbytesperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallresbytesperprofile
appfirewallabortsperprofile	<Double>	Read-only	Incomplete HTTP/HTTPS requests aborted by the client before the Application Firewall could finish processing them.
appfirewallabortsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallabortsperprofile
appfirewallredirectsperprofile	<Double>	Read-only	HTTP/HTTPS requests redirected by the Application Firewall to a different Web page or web server. (HTTP 302)
appfirewallredirectsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallredirectsperprofile
appfirewalllongavgresptimeperprofile	<Double>	Read-only	Average backend response time in milliseconds since reboot
appfirewallshortavgresptimeperprofile	<Double>	Read-only	Average backend response time in milliseconds over the last 7 seconds
appfirewallviolstarturlperprofile	<Double>	Read-only	Number of Start URL security check violations seen by the Application Firewall.
appfirewallviolstarturlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolstarturlperprofile
appfirewallvioldenyurlperprofile	<Double>	Read-only	Number of Deny URL security check violations seen by the Application Firewall.
appfirewallvioldenyurlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallvioldenyurlperprofile
appfirewallviolrefererheaderperprofile	<Double>	Read-only	Number of Referer Header security check violations seen by the Application Firewall.
appfirewallviolrefererheaderperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolrefererheaderperprofile
appfirewallviolbufferoverflowperprofile	<Double>	Read-only	Number of Buffer Overflow security check violations seen by the Application Firewall.
appfirewallviolbufferoverflowperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolbufferoverflowperprofile
appfirewallviolcookieperprofile	<Double>	Read-only	Number of Cookie Consistency security check violations seen by the Application Firewall.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
appfirewallviolcsrftagperprofile	<Double>	Read-only	Number of Cross Site Request Forgery form tag security check violations seen by the Application Firewall.
appfirewallviolcsrftagperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcsrftagperprofile
appfirewallviolxssperprofile	<Double>	Read-only	Number of HTML Cross-Site Scripting security check violations seen by the Application Firewall.
appfirewallviolxssperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxssperprofile
appfirewallviolsqlperprofile	<Double>	Read-only	Number of HTML SQL Injection security check violations seen by the Application Firewall.
appfirewallviolsqlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsqlperprofile
appfirewallviolfieldformatperprofile	<Double>	Read-only	Number of Field Format security check violations seen by the Application Firewall.
appfirewallviolfieldformatperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolfieldformatperprofile
appfirewallviolfieldconsistencyperprofile	<Double>	Read-only	Number of Field Consistency security check violations seen by the Application Firewall.
appfirewallviolfieldconsistencyperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolfieldconsistencyperprofile
appfirewallviolcreditcardperprofile	<Double>	Read-only	Number of Credit Card security check violations seen by the Application Firewall.
appfirewallviolcreditcardperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcreditcardperprofile
appfirewallviolsafeobjectperprofile	<Double>	Read-only	Number of Safe Object security check violations seen by the Application Firewall.
appfirewallviolsafeobjectperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsafeobjectperprofile
appfirewallviolsignatureperprofile	<Double>	Read-only	Number of Signature violations seen by the Application Firewall.
appfirewallviolsignatureperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolsignatureperprofile
appfirewallviolcontenttypeperprofile	<Double>	Read-only	Number of Content Type security check violations seen by the Application Firewall.
appfirewallviolcontenttypeperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolcontenttypeperprofile
appfirewallviolwellformednessviolationsperprofile	<Double>	Read-only	Number of XML Format security check violations seen by the Application Firewall.
appfirewallviolwellformednessviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolwellformednessviolationsperprofile
appfirewallviolxdosviolationsperprofile	<Double>	Read-only	Number of XML Denial-of-Service security check violations seen by the Application Firewall.
appfirewallviolxdosviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxdosviolationsperprofile
appfirewallviolmsgvalviolationsperprofile	<Double>	Read-only	Number of XML Message Validation security check violations seen by the Application Firewall.
appfirewallviolmsgvalviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolmsgvalviolationsperprofile
appfirewallviolwsiviolationsperprofile	<Double>	Read-only	Number of Web Services Interoperability (WS-I) security check violations seen by the Application Firewall.
appfirewallviolwsiviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolwsiviolationsperprofile
appfirewallviolxmlsqlviolationsperprofile	<Double>	Read-only	Number of XML SQL Injection security check violations seen by the Application Firewall.
appfirewallviolxmlsqlviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlsqlviolationsperprofile
appfirewallviolxmlxssviolationsperprofile	<Double>	Read-only	Number of XML Cross-Site Scripting (XSS) security check violations seen by the Application Firewall.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
appfirewallviolxmlattachmentviolationsperprofile	<Double>	Read-only	Number of XML Attachment security check violations seen by the Application Firewall.
appfirewallviolxmlattachmentviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlattachmentviolationsperprofile
appfirewallviolxmlsoapfaultviolationsperprofile	<Double>	Read-only	Number of requests returning soap:fault from the backend server
appfirewallviolxmlsoapfaultviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlsoapfaultviolationsperprofile
appfirewallviolxmlgenericviolationsperprofile	<Double>	Read-only	Number of requests returning XML generic violation from the backend server
appfirewallviolxmlgenericviolationsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallviolxmlgenericviolationsperprofile
appfirewalltotalviolperprofile	<Double>	Read-only	Number of violations seen by the application firewall on per profile basis
appfirewallviolperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalltotalviolperprofile
appfirewalllogstarturlperprofile	<Double>	Read-only	Number of Start URL security check log messages generated by the Application Firewall.
appfirewalllogstarturlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogstarturlperprofile
appfirewalllogdenyurlperprofile	<Double>	Read-only	Number of Deny URL security check log messages generated by the Application Firewall.
appfirewalllogdenyurlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogdenyurlperprofile
appfirewalllogrefererheaderperprofile	<Double>	Read-only	Number of Referer Header security check log messages generated by the Application Firewall.
appfirewalllogrefererheaderperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogrefererheaderperprofile
appfirewalllogbufferoverflowperprofile	<Double>	Read-only	Number of Buffer Overflow security check log messages generated by the Application Firewall.
appfirewalllogbufferoverflowperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogbufferoverflowperprofile
appfirewalllogcookieperprofile	<Double>	Read-only	Number of Cookie Consistency security check log messages generated by the Application Firewall.
appfirewalllogcookieperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcookieperprofile
appfirewalllogcsrftagperprofile	<Double>	Read-only	Number of Cross Site Request Forgery form tag security check log messages generated by the Application Firewall.
appfirewalllogcsrftagperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcsrftagperprofile
appfirewalllogxssperprofile	<Double>	Read-only	Number of HTML Cross-Site Scripting security check log messages generated by the Application Firewall.
appfirewalllogxssperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogxssperprofile
appfirewalllogxformxssperprofile	<Double>	Read-only	Number of HTML Cross-Site Scripting security check transform log messages generated by the Application Firewall.
appfirewalllogxformxssperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogxformxssperprofile
appfirewalllogsqlperprofile	<Double>	Read-only	Number of HTML SQL Injection security check log messages generated by the Application Firewall.
appfirewalllogsqlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogsqlperprofile
appfirewalllogxformsqlperprofile	<Double>	Read-only	Number of HTML SQL Injection security check transform log messages generated by the Application Firewall.
appfirewalllogxformsqlperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogxformsqlperprofile
appfirewalllogfieldformatperprofile	<Double>	Read-only	Number of Field Format security check log messages generated by the Application Firewall.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
appfirewalllogfieldconsistencyperprofile	<Double>	Read-only	Number of Field Consistency security check log messages generated by the Application Firewall.
appfirewalllogfieldconsistencyperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogfieldconsistencyperprofile
appfirewalllogcreditcardperprofile	<Double>	Read-only	Number of Credit Card security check log messages generated by the Application Firewall.
appfirewalllogcreditcardperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcreditcardperprofile
appfirewallxformlogcreditcardperprofile	<Double>	Read-only	Number of Credit Card security check transform log messages generated by the Application Firewall.
appfirewallxformlogcreditcardperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxformlogcreditcardperprofile
appfirewalllogsafeobjectperprofile	<Double>	Read-only	Number of Safe Object security check log messages generated by the Application Firewall.
appfirewalllogsafeobjectperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogsafeobjectperprofile
appfirewalllogcontenttypeperprofile	<Double>	Read-only	Number of Content type security check log messages generated by the Application Firewall.
appfirewalllogcontenttypeperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewalllogcontenttypeperprofile
appfirewallwellformednesslogsperprofile	<Double>	Read-only	Number of XML Format security check log messages generated by the Application Firewall.
appfirewallwellformednesslogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallwellformednesslogsperprofile
appfirewallxdoslogsperprofile	<Double>	Read-only	Number of XML Denial-of-Service security check log messages generated by the Application Firewall.
appfirewallxdoslogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxdoslogsperprofile
appfirewallmsgvallogsperprofile	<Double>	Read-only	Number of XML Message Validation security check log messages generated by the Application Firewall.
appfirewallmsgvallogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallmsgvallogsperprofile
appfirewallwsilogsperprofile	<Double>	Read-only	Number of Web Services Interoperability (WS-I) security check log messages generated by the Application Firewall.
appfirewallwsilogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallwsilogsperprofile
appfirewallxmlsqllogsperprofile	<Double>	Read-only	Number of XML SQL Injection security check log messages generated by the Application Firewall.
appfirewallxmlsqllogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlsqllogsperprofile
appfirewallxmlxsslogsperprofile	<Double>	Read-only	Number of XML Cross-Site Scripting (XSS) security check log messages generated by the Application Firewall.
appfirewallxmlxsslogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlxsslogsperprofile
appfirewallxmlattachmentlogsperprofile	<Double>	Read-only	Number of XML Attachment security check log messages generated by the Application Firewall.
appfirewallxmlattachmentlogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlattachmentlogsperprofile
appfirewallxmlsoapfaultlogsperprofile	<Double>	Read-only	Number of requests generating soap:fault log messages
appfirewallxmlsoapfaultlogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlsoapfaultlogsperprofile
appfirewallxmlgenericlogsperprofile	<Double>	Read-only	Number of requests generating XML Generic log messages
appfirewallxmlgenericlogsperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallxmlgenericlogsperprofile
appfirewalltotallogperprofile	<Double>	Read-only	Number of log messages generated by the application firewall on per profile basis

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
appfirewallret4xxperprofile	<Double>	Read-only	Number of requests returning HTTP 4xx from the backend server
appfirewallret4xxperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallret4xxperprofile
appfirewallret5xxperprofile	<Double>	Read-only	Number of requests returning HTTP 5xx from the backend server
appfirewallret5xxperprofilerate	<Double>	Read-only	Rate (/s) counter for appfirewallret5xxperprofile

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/appfwprofile

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/appfwprofile?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String**

Use this query-parameter to get appfwprofile resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile": [ {
    "name":<String_value>,
    "appfirewallxmlxsslogsperprofilerate":<Double_value>,
    "appfirewallviolmsgvalviolationsperprofile":<Double_value>,
    "appfirewalllogcontenttypeperprofile":<Double_value>,
    "appfirewallviolperprofilerate":<Double_value>,
    "appfirewallviolstarturlperprofile":<Double_value>,
    "appfirewalltotalviolperprofile":<Double_value>,
    "appfirewallret5xxperprofile":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

"appfirewallviolcontenttypeperprofile":<Double_value>,
"appfirewallredirectsperprofile":<Double_value>,
"appfirewallresponsesperprofile":<Double_value>,
"appfirewallxmlsqllogsperprofile":<Double_value>,
"appfirewallwsilogsperprofile":<Double_value>,
"appfirewallrequestsperprofilerate":<Double_value>,
"appfirewallviolxmlsoapfaultviolationsperprofile":<Double_value>,
"appfirewalllogbufferoverflowperprofile":<Double_value>,
"appfirewallviolwsiviolationsperprofilerate":<Double_value>,
"appfirewallviolwellformednessviolationsperprofile":<Double_value>,
"appfirewallviolfieldformatperprofile":<Double_value>,
"appfirewallviolcookieperprofilerate":<Double_value>,
"appfirewallresbytesperprofilerate":<Double_value>,
"appfirewalllogxformxssperprofile":<Double_value>,
"appfirewalllogcookieperprofile":<Double_value>,
"appfirewallviolcreditcardperprofilerate":<Double_value>,
"appfirewallxmlsoapfaultlogsperprofilerate":<Double_value>,
"appfirewalllogfieldconsistencyperprofilerate":<Double_value>,
"appfirewalllogdenyurlperprofile":<Double_value>,
"appfirewallresbytesperprofile":<Double_value>,
"appfirewallviolxssperprofile":<Double_value>,
"appfirewallmsgvallogsperprofilerate":<Double_value>,
"appfirewallviolxmlsoapfaultviolationsperprofilerate":<Double_value>,
"appfirewalllogcsrftagperprofile":<Double_value>,
"appfirewallviolcontenttypeperprofilerate":<Double_value>,
"appfirewallxmlsoapfaultlogsperprofile":<Double_value>,
"appfirewallxformlogcreditcardperprofilerate":<Double_value>,
"appfirewallreqbytesperprofilerate":<Double_value>,
"appfirewallviolmsgvalviolationsperprofilerate":<Double_value>,
"appfirewallxdoslogsperprofile":<Double_value>,
"appfirewallviolfieldconsistencyperprofilerate":<Double_value>,
"appfirewalllogrefererheaderperprofilerate":<Double_value>,
"appfirewallxmlattachmentlogsperprofilerate":<Double_value>,
"appfirewallviolxmlxssviolationsperprofilerate":<Double_value>,
"appfirewallxdoslogsperprofilerate":<Double_value>,
"appfirewallxmlxsslogsperprofile":<Double_value>,
"appfirewallxmlsqllogsperprofilerate":<Double_value>,
"appfirewallrequestsperprofile":<Double_value>,
"appfirewallviolsignatureperprofile":<Double_value>,
"appfirewallviolxmlattachmentviolationsperprofilerate":<Double_value>,
"appfirewallviolsqlperprofilerate":<Double_value>,
"appfirewallviolcsrftagperprofilerate":<Double_value>,
"appfirewallviolstarturlperprofilerate":<Double_value>,
"appfirewalllogxformxssperprofilerate":<Double_value>,
"appfirewallviolxmlsqlviolationsperprofile":<Double_value>,
"appfirewalllogsqlperprofile":<Double_value>,
"appfirewallvioldenyurlperprofile":<Double_value>,
"appfirewallredirectsperprofilerate":<Double_value>,
"appfirewalllogstarturlperprofilerate":<Double_value>,
"appfirewallviolcsrftagperprofile":<Double_value>,
"appfirewallwellformednesslogsperprofile":<Double_value>,
"appfirewalllogdenyurlperprofilerate":<Double_value>,
"appfirewallviolwsiviolationsperprofile":<Double_value>,
"appfirewallshortavgresptimeperprofile":<Double_value>,
"appfirewallviolsafeobjectperprofile":<Double_value>,
"appfirewallresponsesperprofilerate":<Double_value>,
"appfirewallviolrefererheaderperprofilerate":<Double_value>,
"appfirewallvioldenyurlperprofilerate":<Double_value>,
"appfirewallabortsperprofile":<Double_value>,
"appfirewallviolcookieperprofile":<Double_value>,
"appfirewalllogcreditcardperprofile":<Double_value>,
"appfirewalllogbufferoverflowperprofilerate":<Double_value>,
"appfirewallxmlgenericlogsperprofilerate":<Double_value>,
"appfirewallret4xxperprofile":<Double_value>,
"appfirewallviolfieldformatperprofilerate":<Double_value>,
"appfirewalllogxformsqlperprofile":<Double_value>,
"appfirewallviolbufferoverflowperprofilerate":<Double_value>,
"appfirewallmsgvallogsperprofile":<Double_value>,
"appfirewallwsilogsperprofilerate":<Double_value>,
"appfirewallviolsqlperprofile":<Double_value>,
"appfirewallviolxssperprofilerate":<Double_value>,
"appfirewallxmlgenericlogsperprofile":<Double_value>,
"appfirewalllogperprofilerate":<Double_value>,
"appfirewallviolcreditcardperprofile":<Double_value>,
"appfirewallxmlattachmentlogsperprofile":<Double_value>,
"appfirewallviolfieldconsistencyperprofile":<Double_value>,
"appfirewalllogcsrftagperprofilerate":<Double_value>,
"appfirewallret4xxperprofilerate":<Double_value>,
"appfirewalllogfieldformatperprofilerate":<Double_value>,
"appfirewallviolsafeobjectperprofilerate":<Double_value>,
"appfirewalllogxssperprofile":<Double_value>,
"appfirewallviolxmlgenericviolationsperprofile":<Double_value>,
"appfirewallviolsignatureperprofilerate":<Double_value>,
"appfirewalllongavgresptimeperprofile":<Double_value>,
"appfirewalllogfieldformatperprofile":<Double_value>,
"appfirewallabortsperprofilerate":<Double_value>,
"appfirewalllogcreditcardperprofilerate":<Double_value>,
"appfirewalllogsqlperprofilerate":<Double_value>,
"appfirewallviolbufferoverflowperprofile":<Double_value>,
"appfirewalllogsafeobjectperprofilerate":<Double_value>,
"appfirewalllogfieldconsistencyperprofile":<Double_value>,
"appfirewalllogstarturlperprofile":<Double_value>,
"appfirewallviolxmlsqlviolationsperprofilerate":<Double_value>,
"appfirewalllogxssperprofilerate":<Double_value>,
"appfirewalllogcontenttypeperprofilerate":<Double_value>,
"appfirewallwellformednesslogsperprofilerate":<Double_value>,
"appfirewallreqbytesperprofile":<Double_value>,
"appfirewalllogsafeobjectperprofile":<Double_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "appfirewallviolxmlgenericviolationsperprofilerate":<Double_value>,
    "appfirewallviolxmlattachmentviolationsperprofile":<Double_value>,
    "appfirewalllogrefererheaderperprofile":<Double_value>,
    "appfirewallviolxdosviolationsperprofilerate":<Double_value>,
    "appfirewallviolwellformednessviolationsperprofilerate":<Double_value>,
    "appfirewallviolxdosviolationsperprofile":<Double_value>,
    "appfirewalllogxformsqlperprofilerate":<Double_value>,
    "appfirewallviolxmlxssviolationsperprofile":<Double_value>,
    "appfirewalltotallogperprofile":<Double_value>,
    "appfirewalllogcookieperprofilerate":<Double_value>,
    "appfirewallret5xxperprofilerate":<Double_value>,
    "appfirewallviolrefererheaderperprofile":<Double_value>,
    "appfirewallxformlogcreditcardperprofile":<Double_value>
}}}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/appfwprofile/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appfwprofile": [ {
    "name":<String_value>,
    "appfirewallxmlxsslogsperprofilerate":<Double_value>,
    "appfirewallviolmsgvalviolationsperprofile":<Double_value>,
    "appfirewalllogcontenttypeperprofile":<Double_value>,
    "appfirewallviolperprofilerate":<Double_value>,
    "appfirewallviolstarturlperprofile":<Double_value>,
    "appfirewalltotalviolperprofile":<Double_value>,
    "appfirewallret5xxperprofile":<Double_value>,
    "appfirewallviolcontenttypeperprofile":<Double_value>,
    "appfirewallredirectsperprofile":<Double_value>,
    "appfirewallresponsesperprofile":<Double_value>,
    "appfirewallxmlsqllogsperprofile":<Double_value>,
    "appfirewallwsilogsperprofile":<Double_value>,
    "appfirewallrequestsperprofilerate":<Double_value>,
    "appfirewallviolxmlsoapfaultviolationsperprofile":<Double_value>,
    "appfirewalllogbufferoverflowperprofile":<Double_value>,
    "appfirewallviolwsiviolationsperprofilerate":<Double_value>,
    "appfirewallviolwellformednessviolationsperprofile":<Double_value>,
    "appfirewallviolfieldformatperprofile":<Double_value>,
    "appfirewallviolcookieperprofilerate":<Double_value>,
    "appfirewallresbytesperprofilerate":<Double_value>,
    "appfirewalllogxformxssperprofile":<Double_value>,
    "appfirewalllogcookieperprofile":<Double_value>,
    "appfirewallviolcreditcardperprofilerate":<Double_value>,
    "appfirewallxmlsoapfaultlogsperprofilerate":<Double_value>,
    "appfirewalllogfieldconsistencyperprofilerate":<Double_value>,
    "appfirewalllogdenyurlperprofile":<Double_value>,
    "appfirewallresbytesperprofile":<Double_value>,
    "appfirewallviolxssperprofile":<Double_value>,
    "appfirewallmsgvallogsperprofilerate":<Double_value>,
    "appfirewallviolxmlsoapfaultviolationsperprofilerate":<Double_value>,
    "appfirewalllogcsrftagperprofile":<Double_value>,
    "appfirewallviolcontenttypeperprofilerate":<Double_value>,
    "appfirewallxmlsoapfaultlogsperprofile":<Double_value>,
    "appfirewallxformlogcreditcardperprofilerate":<Double_value>,
    "appfirewallreqbytesperprofilerate":<Double_value>,
    "appfirewallviolmsgvalviolationsperprofilerate":<Double_value>,
    "appfirewallxdoslogsperprofile":<Double_value>,
    "appfirewallviolfieldconsistencyperprofilerate":<Double_value>,
    "appfirewalllogrefererheaderperprofilerate":<Double_value>,
    "appfirewallxmlattachmentlogsperprofilerate":<Double_value>,
    "appfirewallviolxmlxssviolationsperprofilerate":<Double_value>,
    "appfirewallxdoslogsperprofilerate":<Double_value>,
    "appfirewallxmlxsslogsperprofile":<Double_value>,
    "appfirewallxmlsqllogsperprofilerate":<Double_value>,
    "appfirewallrequestsperprofile":<Double_value>,
    "appfirewallviolsignatureperprofile":<Double_value>,
    "appfirewallviolxmlattachmentviolationsperprofilerate":<Double_value>,
    "appfirewallviolsqlperprofilerate":<Double_value>,
    "appfirewallviolcsrftagperprofilerate":<Double_value>,
    "appfirewallviolstarturlperprofilerate":<Double_value>,
    "appfirewalllogxformxssperprofilerate":<Double_value>,
    "appfirewallviolxmlsqlviolationsperprofile":<Double_value>,
    "appfirewalllogsqlperprofile":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "appfirewallvioldenyurlperprofile":<Double_value>,
    "appfirewallredirectsperprofilerate":<Double_value>,
    "appfirewalllogstarturlperprofilerate":<Double_value>,
    "appfirewallviolcsrftagperprofile":<Double_value>,
    "appfirewallwellformednesslogsperprofile":<Double_value>,
    "appfirewalllogdenyurlperprofilerate":<Double_value>,
    "appfirewallviolwsiviolationsperprofile":<Double_value>,
    "appfirewallshortavgresptimeperprofile":<Double_value>,
    "appfirewallviolsafeobjectperprofile":<Double_value>,
    "appfirewallresponsesperprofilerate":<Double_value>,
    "appfirewallviolrefererheaderperprofilerate":<Double_value>,
    "appfirewallvioldenyurlperprofilerate":<Double_value>,
    "appfirewallabortsperprofile":<Double_value>,
    "appfirewallviolcookieperprofile":<Double_value>,
    "appfirewalllogcreditcardperprofile":<Double_value>,
    "appfirewalllogbufferoverflowperprofilerate":<Double_value>,
    "appfirewallxmlgenericlogsperprofilerate":<Double_value>,
    "appfirewallret4xxperprofile":<Double_value>,
    "appfirewallviolfieldformatperprofilerate":<Double_value>,
    "appfirewalllogxformsqlperprofile":<Double_value>,
    "appfirewallviolbufferoverflowperprofilerate":<Double_value>,
    "appfirewallmsgvallogsperprofile":<Double_value>,
    "appfirewallwsilogsperprofilerate":<Double_value>,
    "appfirewallviolsqlperprofile":<Double_value>,
    "appfirewallviolxssperprofilerate":<Double_value>,
    "appfirewallxmlgenericlogsperprofile":<Double_value>,
    "appfirewalllogperprofilerate":<Double_value>,
    "appfirewallviolcreditcardperprofile":<Double_value>,
    "appfirewallxmlattachmentlogsperprofile":<Double_value>,
    "appfirewallviolfieldconsistencyperprofile":<Double_value>,
    "appfirewalllogcsrftagperprofilerate":<Double_value>,
    "appfirewallret4xxperprofilerate":<Double_value>,
    "appfirewalllogfieldformatperprofilerate":<Double_value>,
    "appfirewallviolsafeobjectperprofilerate":<Double_value>,
    "appfirewalllogxssperprofile":<Double_value>,
    "appfirewallviolxmlgenericviolationsperprofile":<Double_value>,
    "appfirewallviolsignatureperprofilerate":<Double_value>,
    "appfirewalllongavgresptimeperprofile":<Double_value>,
    "appfirewalllogfieldformatperprofile":<Double_value>,
    "appfirewallabortsperprofilerate":<Double_value>,
    "appfirewalllogcreditcardperprofilerate":<Double_value>,
    "appfirewalllogsqlperprofilerate":<Double_value>,
    "appfirewallviolbufferoverflowperprofile":<Double_value>,
    "appfirewalllogsafeobjectperprofilerate":<Double_value>,
    "appfirewalllogfieldconsistencyperprofile":<Double_value>,
    "appfirewalllogstarturlperprofile":<Double_value>,
    "appfirewallviolxmlsqlviolationsperprofilerate":<Double_value>,
    "appfirewalllogxssperprofilerate":<Double_value>,
    "appfirewalllogcontenttypeperprofilerate":<Double_value>,
    "appfirewallwellformednesslogsperprofilerate":<Double_value>,
    "appfirewallreqbytesperprofile":<Double_value>,
    "appfirewalllogsafeobjectperprofile":<Double_value>,
    "appfirewallviolxmlgenericviolationsperprofilerate":<Double_value>,
    "appfirewallviolxmlattachmentviolationsperprofile":<Double_value>,
    "appfirewalllogrefererheaderperprofile":<Double_value>,
    "appfirewallviolxdosviolationsperprofilerate":<Double_value>,
    "appfirewallviolwellformednessviolationsperprofilerate":<Double_value>,
    "appfirewallviolxdosviolationsperprofile":<Double_value>,
    "appfirewalllogxformsqlperprofilerate":<Double_value>,
    "appfirewallviolxmlxssviolationsperprofile":<Double_value>,
    "appfirewalltotallogperprofile":<Double_value>,
    "appfirewalllogcookieperprofilerate":<Double_value>,
    "appfirewallret5xxperprofilerate":<Double_value>,
    "appfirewallviolrefererheaderperprofile":<Double_value>,
    "appfirewallxformlogcreditcardperprofile":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Appqoe

May 20, 2016

Application Level Quality of Experience statistics

Name	Description
appqoe	Statistics for appqoe.
appqoepolicy	Statistics for AppQoS policy resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

appqoe

Jul 07, 2017
Statistics for appqoe.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
totinmemrsp	<Double>	Read-only	Total in-memory responses sent from NS
inmemrsprate	<Double>	Read-only	Rate (/s) counter for totinmemrsp
totfaultycookies	<Double>	Read-only	Total faulty cookies received
faultycookiesrate	<Double>	Read-only	Rate (/s) counter for totfaultycookies
totvalidcookies	<Double>	Read-only	Total valid cookies received
validcookiesrate	<Double>	Read-only	Rate (/s) counter for totvalidcookies
tothighprireq	<Double>	Read-only	Total Requests served from higher priority queue
highprireqrate	<Double>	Read-only	Rate (/s) counter for tothighprireq
totmediumprireq	<Double>	Read-only	Total Requests served from medium priority queue
mediumprireqrate	<Double>	Read-only	Rate (/s) counter for totmediumprireq
totlowprireq	<Double>	Read-only	Total Requests served from low priority queue
lowprireqrate	<Double>	Read-only	Rate (/s) counter for totlowprireq
totlowestprireq	<Double>	Read-only	Total Requests served from surge priority queue
lowestprireqrate	<Double>	Read-only	Rate (/s) counter for totlowestprireq
totaltsvrsubfailed	<Double>	Read-only	Total number of times alternate server substitution failed
tsvrsubfailedrate	<Double>	Read-only	Rate (/s) counter for totaltsvrsubfailed
totdostrig	<Double>	Read-only	Total number of times HDOS condition triggered
dostrigrate	<Double>	Read-only	Rate (/s) counter for totdostrig
totdosqvalidcookies	<Double>	Read-only	Total DOSQ valid cookies received
dosqvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for totdosqvalidcookies
totdoshvalidcookies	<Double>	Read-only	Total DOSH valid cookies received

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
totsidvalidcookies	<Double>	Read-only	Total SID valid cookies received
sidvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for totsidvalidcookies
totonhvalidcookies	<Double>	Read-only	Total ONH valid cookies received
onhvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for totonhvalidcookies
totpriqvalidcookies	<Double>	Read-only	Total PRIQ valid cookies received
priqvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for totpriqvalidcookies
totdosqfaultycookies	<Double>	Read-only	Total DOSQ faulty cookies received
dosqfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for totdosqfaultycookies
totdoshfaultycookies	<Double>	Read-only	Total DOSH faulty cookies received
doshfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for totdoshfaultycookies
totsidfaultycookies	<Double>	Read-only	Total SID faulty cookies received
sidfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for totsidfaultycookies
totonhfaultycookies	<Double>	Read-only	Total ONH faulty cookies received
onhfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for totonhfaultycookies
totpriqfaultycookies	<Double>	Read-only	Total PRIQ faulty cookies received
priqfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for totpriqfaultycookies
totpriembedlinks	<Double>	Read-only	Total requests for valid embedded links
priembedlinksrate	<Double>	Read-only	Rate (/s) counter for totpriembedlinks
totsessreq	<Double>	Read-only	Total valid SIDQ requests within session
sessreqrate	<Double>	Read-only	Rate (/s) counter for totsessreq
totaltcntreq	<Double>	Read-only	Total requests for alternate contents
tcntreqrate	<Double>	Read-only	Rate (/s) counter for totaltcntreq
totgetinmemrsp	<Double>	Read-only	Total in-memory GET responses sent from NS
getinmemrsprate	<Double>	Read-only	Rate (/s) counter for totgetinmemrsp
totpostinmemrsp	<Double>	Read-only	Total in-memory POST responses sent from NS
postinmemrsprate	<Double>	Read-only	Rate (/s) counter for totpostinmemrsp

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
postinmemrspbytesrate	<Double>	Read-only	Rate (/s) counter for totpostinmemrspbytes

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/appqoe

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/appqoe?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get appqoe resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoe": [ {
    "totpriqvalidcookies":<Double_value>,
    "totonhvalidcookies":<Double_value>,
    "dosqvalidcookiesrate":<Double_value>,
    "sidvalidcookiesrate":<Double_value>,
    "onhvalidcookiesrate":<Double_value>,
    "totdosqvalidcookies":<Double_value>,
    "dostrigrate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "totgetinmemrsp":<Double_value>,
    "totlowprireq":<Double_value>,
    "totdoshfaultycookies":<Double_value>,
    "doshfaultycookiesrate":<Double_value>,
    "sessreqrate":<Double_value>,
    "totpostinmemrspbytes":<Double_value>,
    "totpriqfaultycookies":<Double_value>,
    "totpostinmemrsp":<Double_value>,
    "priembedlinksrate":<Double_value>,
    "mediumprireqrate":<Double_value>,
    "totdostrig":<Double_value>,
    "totvalidcookies":<Double_value>,
    "totinmemrsp":<Double_value>,
    "totsidfaultycookies":<Double_value>,
    "totsidvalidcookies":<Double_value>,
    "totdosqfaultycookies":<Double_value>,
    "tsvrsubfailedrate":<Double_value>,
    "postinmemrsprate":<Double_value>,
    "totpriembedlinks":<Double_value>,
    "totmediumprireq":<Double_value>,
    "priqvalidcookiesrate":<Double_value>,
    "totfaultycookies":<Double_value>,
    "doshvalidcookiesrate":<Double_value>,
    "sidfaultycookiesrate":<Double_value>,
    "highprireqrate":<Double_value>,
    "dosqfaultycookiesrate":<Double_value>,
    "inmemrsprate":<Double_value>,
    "getinmemrsprate":<Double_value>,
    "onhfaultycookiesrate":<Double_value>,
    "validcookiesrate":<Double_value>,
    "totaltcntreq":<Double_value>,
    "totsessreq":<Double_value>,
    "priqfaultycookiesrate":<Double_value>,
    "tcntreqrate":<Double_value>,
    "totdoshvalidcookies":<Double_value>,
    "lowprireqrate":<Double_value>,
    "tothighprireq":<Double_value>,
    "totaltsvrsubfailed":<Double_value>,
    "totlowestprireq":<Double_value>,
    "lowestprireqrate":<Double_value>,
    "totonhfaultycookies":<Double_value>,
    "faultycookiesrate":<Double_value>,
    "postinmemrspbytesrate":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

appqoepolicy

May 20, 2016

Statistics for AppQoS policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	policyName. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
qosavgserverttfb	<Double>	Read-only	Average Server Time-To-First-Byte in milliseconds calculated for this AppQoE policy.
qosavgserverttfbrate	<Double>	Read-only	Rate (/s) counter for qosavgserverttfb
qosavgserverttlb	<Double>	Read-only	Average Server Time-To-Last-Byte in milliseconds calculated for this AppQoE policy.
qosavgserverttlbrate	<Double>	Read-only	Rate (/s) counter for qosavgserverttlb
qosavgclientttlb	<Double>	Read-only	Average Client Time-To-Last-Byte in milliseconds calculated for this AppQoE policy.
qosavgclientttlbrate	<Double>	Read-only	Rate (/s) counter for qosavgclientttlb
totthroughput	<Double>	Read-only	Throughput in Kbps calculated on this AppQoE policy
throughputrate	<Double>	Read-only	Rate (/s) counter for totthroughput
totsvrlinkedto	<Double>	Read-only	Total number of server connections that were established through this AppQoE Policy
svrlinkedtorate	<Double>	Read-only	Rate (/s) counter for totsvrlinkedto
totcltrequests	<Double>	Read-only	Total number of client connections that were requested through this AppQoE Policy
cltrequestsrate	<Double>	Read-only	Rate (/s) counter for totcltrequests
totrequests	<Double>	Read-only	Total number of requests that were requested through this AppQoE policy
requestsrate	<Double>	Read-only	Rate (/s) counter for totrequests
totrequestbytes	<Double>	Read-only	Total number of requests bytes that were requested through this AppQoE policy
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totrequestbytes
totresponse	<Double>	Read-only	Total number of responses received by this AppQoE policy
responserate	<Double>	Read-only	Rate (/s) counter for totresponse
totresponsebytes	<Double>	Read-only	Total number of response bytes received by this AppQoE policy
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totresponsebytes
totjssent	<Double>	Read-only	Total number of in-memory responses sent instead of expected responses through this AppQoE policy
jssentrate	<Double>	Read-only	Rate (/s) counter for totjssent
totjsbytessent	<Double>	Read-only	Total bytes of in-memory responses sent through this AppQoE policy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/appqoepolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/appqoepolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String**

Use this query-parameter to get appqoepolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoepolicy": [ {
    "name":<String_value>,
    "qosavgclientttlbrate":<Double_value>,
    "totjssent":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totthroughput":<Double_value>,
    "requestbytesrate":<Double_value>,
    "totjsbytessent":<Double_value>,
    "svrlinkedtorate":<Double_value>,
    "totresponse":<Double_value>,
    "totresponsebytes":<Double_value>,
    "jssentrate":<Double_value>,
    "qosavgserverttfbrate":<Double_value>,
    "pipolicyhits":<Double_value>,
    "qosavgclientttlb":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "totcltrequests":<Double_value>,
    "totrequests":<Double_value>,
    "qosavgserverttlbrate":<Double_value>,
    "totrequestbytes":<Double_value>,
    "cltrequestsrate":<Double_value>,
    "throughputrate":<Double_value>,
    "totsvrlinkedto":<Double_value>,
    "qosavgserverttfb":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "jsbytessentrate":<Double_value>,
    "responserate":<Double_value>,
    "requestsrate":<Double_value>,
    "qosavgserverttlb":<Double_value>
}}}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/appqoepolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "appqoepolicy": [ {
    "name":<String_value>,
    "qosavgclientttlbrate":<Double_value>,
    "totjssent":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totthroughput":<Double_value>,
    "requestbytesrate":<Double_value>,
    "totjsbytessent":<Double_value>,
    "svrlinkedtorate":<Double_value>,
    "totresponse":<Double_value>,
    "totresponsebytes":<Double_value>,
    "jssentrate":<Double_value>,
    "qosavgserverttfbrate":<Double_value>,
    "pipolicyhits":<Double_value>,
    "qosavgclientttlb":<Double_value>,
    "totcltrequests":<Double_value>,
    "totrequests":<Double_value>,
    "qosavgserverttlbrate":<Double_value>,
    "totrequestbytes":<Double_value>,
    "cltrequestsrate":<Double_value>,
    "throughputrate":<Double_value>,
    "totsvrlinkedto":<Double_value>,
    "qosavgserverttfb":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "jsbytessentrate":<Double_value>,
    "responserate":<Double_value>,
    "requestsrate":<Double_value>,
    "qosavgserverttlb":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Audit

May 20, 2016

Audit statistics.

Name	Description
audit	Statistics for audit.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

audit

Jul 07, 2017
Statistics for audit.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
auditsyslogmsgsent	<Double>	Read-only	Syslog messages sent to the syslog server(s) over UDP.
auditsyslogmsgsentrate	<Double>	Read-only	Rate (/s) counter for auditsyslogmsgsent
auditsyslogmsggen	<Double>	Read-only	Syslog messages about to be sent to the syslog server.
auditsyslogmsggenrate	<Double>	Read-only	Rate (/s) counter for auditsyslogmsggen
auditsyslogmsgsenttcp	<Double>	Read-only	Syslog messages sent to the syslog server(s) over TCP.
auditsyslogmsgsenttcprate	<Double>	Read-only	Rate (/s) counter for auditsyslogmsgsenttcp
auditnsballocfail	<Double>	Read-only	NAT allocation failed.
auditnsballocfailrate	<Double>	Read-only	Rate (/s) counter for auditnsballocfail
auditlog32errsyslogallocnsbfail	<Double>	Read-only	Nsb allocation failed.
auditlog32errsyslogallocnsbfailrate	<Double>	Read-only	Rate (/s) counter for auditlog32errsyslogallocnsbfail
auditmemallocfail	<Double>	Read-only	Failures in allocation of Access Gateway context structure. When an Access Gateway session is established, the NetScaler creates an internal context structure , which identifies the user and the IP address from which the user has logged in.
auditmemallocfailrate	<Double>	Read-only	Rate (/s) counter for auditmemallocfail
auditportallocfail	<Double>	Read-only	Number of times the NetScaler failed to allocate a port when sending a syslog message to the syslog server(s).
auditportallocfailrate	<Double>	Read-only	Rate (/s) counter for auditportallocfail
auditnatpcblkupfail	<Double>	Read-only	NAT lookup failed.
auditnatpcblkupfailrate	<Double>	Read-only	Rate (/s) counter for auditnatpcblkupfail
auditcontextnotfound	<Double>	Read-only	Failures in finding the context structure for an Access Gateway session during attempts to send session-specific audit messages. During an Access Gateway session, audit messages related to the session are queued up in the auditlog buffer for transmission to the audit log server(s). If the session is killed before the messages are sent, the context structure allocated at session creation is removed. This structure is needed for sending the queued auditlog messages. If it is not found, this counter is incremented.
auditcontextnotfoundrate	<Double>	Read-only	Rate (/s) counter for auditcontextnotfound
nsbchainallocfail	<Double>	Read-only	Nsb Chain allocation failed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
clientconnfail	<Double>	Read-only	Failures in establishment of a connection between the NetScaler and the auditserver tool (the Netscaler's custom logging tool).
clientconnfailrate	<Double>	Read-only	Rate (/s) counter for clientconnfail
flushcmdcnt	<Double>	Read-only	Auditlog buffer flushes. In a multiprocessor NetScaler, both the main processor and the co-processor can generate auditlog messages and fill up the auditlog buffers. But only the primary processor can free up the buffers by sending auditlog messages to the auditlog server(s). The number of auditlog buffers is fixed. If the co-processor detects that all the auditlog buffers are full, it issues a flush command to the main processor.
flushcmdcntrate	<Double>	Read-only	Rate (/s) counter for flushcmdcnt
systcpconnfail	<Double>	Read-only	Failures in establishment of a connection between the NetScaler and the syslog server.
systcpconnfailrate	<Double>	Read-only	Rate (/s) counter for systcpconnfail
logunsentlbsys	<Double>	Read-only	Total auditlog messages which are not delivered to load balanced syslog servers
logunsentlbsysrate	<Double>	Read-only	Rate (/s) counter for logunsentlbsys
logsdropped	<Double>	Read-only	Total number of log messages dropped by NetScaler after max hold limit is reached
logsdroppedrate	<Double>	Read-only	Rate (/s) counter for logsdropped
logsdroppedtxminnsbs	<Double>	Read-only	Total number of log messages dropped by NetScaler when NSBQ length is less than TX min NSBs
logsdroppedtxminnsbsrate	<Double>	Read-only	Rate (/s) counter for logsdroppedtxminnsbs

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/audit

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/audit?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get audit resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "audit": [ {
    "auditportallocfailrate":<Double_value>,
    "auditnsballocfailrate":<Double_value>,
    "auditsyslogmsgsenttcprate":<Double_value>,
    "auditcontextnotfoundrate":<Double_value>,
    "logsdroppedtxminnsbs":<Double_value>,
    "clientconnfail":<Double_value>,
    "flushcmdcntrate":<Double_value>,
    "auditlog32errsyslogallocnsbfailrate":<Double_value>,
    "auditnsballocfail":<Double_value>,
    "auditsyslogmsgsentrate":<Double_value>,
    "clientconnfailrate":<Double_value>,
    "auditnatpcblkupfailrate":<Double_value>,
    "auditmemallocfailrate":<Double_value>,
    "auditlog32errsyslogallocnsbfail":<Double_value>,
    "logsdroppedrate":<Double_value>,
    "auditportallocfail":<Double_value>,
    "nsbchainallocfail":<Double_value>,
    "logunsentlbsys":<Double_value>,
    "auditnatpcblkupfail":<Double_value>,
    "systcpconnfailrate":<Double_value>,
    "systcpconnfail":<Double_value>,
    "auditmemallocfail":<Double_value>,
    "logunsentlbsysrate":<Double_value>,
    "auditsyslogmsggenrate":<Double_value>,
    "logsdropped":<Double_value>,
    "auditsyslogmsgsent":<Double_value>,
    "auditcontextnotfound":<Double_value>,
    "flushcmdcnt":<Double_value>,
    "logsdroppedtxminnsbsrate":<Double_value>,
    "auditsyslogmsggen":<Double_value>,
    "nsbchainallocfailrate":<Double_value>,
    "auditsyslogmsgsenttcp":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Authentication

Jul 07, 2017

Authentication statistics.

Name	Description
authenticationloginschemapolicy	
authenticationpolicy	Statistics for Authentication Policy resource.
authenticationpolicylabel	Statistics for authentication policy label resource.
authenticationsamlidppolicy	Statistics for AAA Saml IdentityProvider (IdP) policy resource.
authenticationvserver	Statistics for authentication virtual server resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationloginschemapolicy

May 20, 2016
Read/write properties

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the LoginSchema policy for which statistics will be displayed. If not given statistics are shown for all policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/authenticationloginschemapolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/authenticationloginschemapolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>**

Use this query-parameter to get authenticationloginschemapolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationloginschemapolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/authenticationloginschemapolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationloginschemapolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationpolicy

May 20, 2016

Statistics for Authentication Policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the advanced authentication policy for which to display statistics. If no name is specified, statistics for all advanced authentication polices are shown. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/authenticationpolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/authenticationpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:**

Use this query-parameter to get authenticationpolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/authenticationpolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationpolicylabel

May 20, 2016

Statistics for authentication policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the authentication policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/authenticationpolicylabel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/authenticationpolicylabel?**args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,cl**

Use this query-parameter to get authenticationpolicylabel resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicylabel": [ {
    "labelname":<String_value>,
    "pipolicylabelhitsrate":<Double_value>,
    "pipolicylabelhits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/authenticationpolicylabel/labelname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationpolicylabel": [ {
    "labelname":<String_value>,
    "pipolicylabelhitsrate":<Double_value>,
    "pipolicylabelhits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationsamlidppolicy

May 20, 2016

Statistics for AAA Saml IdentityProvider (IdP) policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the SAML Identity Provider (IdP) policy for which statistics will be displayed. If not given statistics are shown for all policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/authenticationsamlidppolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/authenticationsamlidppolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clea**

Use this query-parameter to get authenticationsamlidppolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlidppolicy": [ {
    "name":<String_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/authenticationsamlidppolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationsamlidppolicy": [ {
    "name":<String_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

authenticationvserver

May 20, 2016

Statistics for authentication virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the authentication virtual server. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/authenticationvserver

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/authenticationvserver?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats**

Use this query-parameter to get authenticationvserver resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver": [ {
    "name":<String_value>,
    "totalrequests":<Double_value>,
    "responsebytesrate":<Double_value>,
    "requestbytesrate":<Double_value>,
    "type":<String_value>,
    "primaryipaddress":<String_value>,
    "responsesrate":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "state":<String_value>,
    "totalresponses":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/authenticationvserver/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authenticationvserver": [ {
    "name":<String_value>,
    "totalrequests":<Double_value>,
    "responsebytesrate":<Double_value>,
    "requestbytesrate":<Double_value>,
    "type":<String_value>,
    "primaryipaddress":<String_value>,
    "responsesrate":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "state":<String_value>,
    "totalresponses":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Authorization

May 20, 2016

Authorization statistics. Authorization services check which resources users are authorized to access, and grant permissions accordingly.

Name	Description
authorizationpolicylabel	Statistics for authorization policy label resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

authorizationpolicylabel

May 20, 2016

Statistics for authorization policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the authorization policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/authorizationpolicylabel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/authorizationpolicylabel?**args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clea**

Use this query-parameter to get authorizationpolicylabel resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicylabel": [ {
      "labelname":<String_value>,
      "pipolicylabelhitsrate":<Double_value>,
      "pipolicylabelhits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/authorizationpolicylabel/labelname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "authorizationpolicylabel": [ {
      "labelname":<String_value>,
      "pipolicylabelhitsrate":<Double_value>,
      "pipolicylabelhits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Autoscale

May 20, 2016

AutoScale statistics.

Name	Description
autoscalepolicy	Statistics for Autoscale policy resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

autoscalepolicy

May 20, 2016

Statistics for Autoscale policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the autoscale policy for which statistics will be displayed. If not given statistics are shown for all autoscale policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/autoscalepolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/autoscalepolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<Stri**

Use this query-parameter to get autoscalepolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "autoscalepolicy": [ {
    "name":<String_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>,
    "pipolicyundefhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/autoscalepolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "autoscalepolicy": [ {
    "name":<String_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>,
    "pipolicyundefhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Basic

Jul 07, 2017

Basic system statistics.

Name	Description
service	Statistics for service resource.
servicegroup	Statistics for service group resource.
servicegroupmember	Statistics for service group entity resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

service

May 20, 2016
Statistics for service resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the service. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
throughput	<Double>	Read-only	Number of bytes received or sent by this service (Mbps).
throughputrate	<Double>	Read-only	Rate (/s) counter for throughput
avgsvrttfb	<Double>	Read-only	Average TTFB between the NetScaler appliance and the server.TTFB is the time interval between sending the request packet to a service and receiving the first response from the service
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
servicetype	<String>	Read-only	The service type of this service.Possible values are ADNS, DNS, MYSQL, RTSP, SSL_DIAMETER, ADNS_TCP, DNS_TCP, NNTP, SIP_UDP, SSL_TCP, ANY, FTP, RADIUS, SNMP, TCP, DHCPRA, HTTP, RDP, SSL, TFTP, DIAMETER, MSSQL, RPCSVR, SSL_BRIDGE, UDP
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
curclntconnections	<Double>	Read-only	Number of current client connections.
surgecount	<Double>	Read-only	Number of requests in the surge queue.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
svrestablishedconn	<Double>	Read-only	Number of server connections in ESTABLISHED state.
curreusepool	<Double>	Read-only	Number of requests in the idle queue/reuse pool.
maxclients	<Double>	Read-only	Maximum open connections allowed on this service.
curload	<Double>	Read-only	Load on the service that is calculated from the bound load based monitor.
curtflags	<Double>	Read-only	Current flags on the service for internal use in display handlers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
vsvrservicehits	<Double>	Read-only	Number of times that the service has been provided.
vsvrservicehitsrate	<Double>	Read-only	Rate (/s) counter for vsvrservicehits
activetransactions	<Double>	Read-only	Number of active transactions handled by this service. (Including those in the surge queue.) Active Transaction means number of transactions currently served by the server including those waiting in the SurgeQ

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/service

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/service?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value**

Use this query-parameter to get service resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "service": [ {
    "name":<String_value>,
    "svrestablishedconn":<Double_value>,
    "curcIntconnections":<Double_value>,
    "servicetype":<String_value>,
    "totalrequests":<Double_value>,
    "surgecount":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "throughput":<Double_value>,
    "throughputrate":<Double_value>,
    "curtflags":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cursrvrconnections":<Double_value>,
    "primaryipaddress":<String_value>,
    "activetransactions":<Double_value>,
    "responsesrate":<Double_value>,
    "maxclients":<Double_value>,
    "avgsvrttfb":<Double_value>,
    "curload":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "curreusepool":<Double_value>,
    "state":<String_value>,
    "vsvrservicehits":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>,
    "vsvrservicehitsrate":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/service/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "service": [ {
    "name":<String_value>,
    "svrestablishedconn":<Double_value>,
    "curcIntconnections":<Double_value>,
    "servicetype":<String_value>,
    "totalrequests":<Double_value>,
    "surgecount":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "throughput":<Double_value>,
    "throughputrate":<Double_value>,
    "curtflags":<Double_value>,
    "cursrvrconnections":<Double_value>,
    "primaryipaddress":<String_value>,
    "activetransactions":<Double_value>,
    "responsesrate":<Double_value>,
    "maxclients":<Double_value>,
    "avgsvrttfb":<Double_value>,
    "curload":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "curreusepool":<Double_value>,
    "state":<String_value>,
    "vsvrservicehits":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>,
    "vsvrservicehitsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

servicegroup

May 20, 2016

Statistics for service group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Name of the service group for which to display settings. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
servicetype	<String>	Read-only	The service type of this service.Possible values are ADNS, DNS, MYSQL, RTSP, SSL_DIAMETER, ADNS_TCP, DNS_TCP, NNTP, SIP_UDP, SSL_TCP, ANY, FTP, RADIUS, SNMP, TCP, DHCPRA, HTTP, RDP, SSL, TFTP, DIAMETER, MSSQL, RPCSVR, SSL_BRIDGE, UDP

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/servicegroup

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/servicegroup?**args=servicegroupname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clears**

Use this query-parameter to get servicegroup resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroup": [ {
    "servicegroupname":<String_value>,
    "servicetype":<String_value>,
    "state":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/servicegroup/servicegroupname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroup": [ {
    "servicegroupname":<String_value>,
    "servicetype":<String_value>,
    "state":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

servicegroupmember

May 20, 2016

Statistics for service group entity resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicegroupname	<String>	Read-write	Displays statistics for the specified service group.Name of the service group. Must begin with an ASCII alphanumeric or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at sign (@), equal sign (=), and hyphen (-) characters. CLI Users: If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my servicegroup" or 'my servicegroup'). Minimum length = 1
ip	<String>	Read-write	IP address of the service group. Mutually exclusive with the server name parameter.
servername	<String>	Read-write	Name of the server. Mutually exclusive with the IP address parameter. Minimum length = 1
port	<Integer>	Read-write	Port number of the service group member. Range 1 - 65535
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
avgsvrttfb	<Double>	Read-only	Average TTFB between the NetScaler appliance and the server.TTFB is the time interval between sending the request packet to a service and receiving the first response from the service
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
servicetype	<String>	Read-only	The service type of this service.Possible values are ADNS, DNS, MYSQL, RTSP, SSL_DIAMETER, ADNS_TCP, DNS_TCP, NNTP, SIP_UDP, SSL_TCP, ANY, FTP, RADIUS, SNMP, TCP, DHCPRA, HTTP, RDP, SSL, TFTP, DIAMETER, MSSQL, RPCSVR, SSL_BRIDGE, UDP
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
curclntconnections	<Double>	Read-only	Number of current client connections.
surgecount	<Double>	Read-only	Number of requests in the surge queue.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
svrestablishedconn	<Double>	Read-only	Number of server connections in ESTABLISHED state.
curreusepool	<Double>	Read-only	Number of requests in the idle queue/reuse pool.
maxclients	<Double>	Read-only	Maximum open connections allowed on this service.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/servicegroupmember

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/servicegroupmember?**args=servicegroupname:<String_value>,ip:<String_value>,servername:<String_value>,port:<Integer_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get servicegroupmember resources based on additional properties.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "servicegroupmember": [ {
    "servicegroupname":<String_value>,
    "svrestablishedconn":<Double_value>,
    "curclntconnections":<Double_value>,
    "servicetype":<String_value>,
    "totalrequests":<Double_value>,
    "surgecount":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "cursrvrconnections":<Double_value>,
    "primaryipaddress":<String_value>,
    "responsesrate":<Double_value>,
    "maxclients":<Double_value>,
    "avgsvrttfb":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "curreusepool":<Double_value>,
    "state":<String_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Ca

May 20, 2016

Content Accelerator statistics. The Content Accelerator feature is used to cache static and dynamic web application data.

Name	Description
ca	Statistics for ca.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ca

May 20, 2016
Statistics for ca.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
caaunindentifiedbyteshr	<Double>	Read-only	This tells the hit ratio of unindentified bytes
caiosbyteshr	<Double>	Read-only	This tells the hit ratio of IOS bytes
caotherbyteshr	<Double>	Read-only	This tells the hit ratio of Other device
calaptpdsktpbyteshr	<Double>	Read-only	This tells the hit ratio of laptop_desktop bytes
caandroidbyteshr	<Double>	Read-only	This tells the hit ratio of 3GP bytes
ca3gpvidcachebyteshr	<Double>	Read-only	This tells the hit ratio of 3GP bytes
camp4vidcachebyteshr	<Double>	Read-only	This tells the hit ratio of MP4 bytes
caflvvidcachebyteshr	<Double>	Read-only	This tells the hit ratio of FLV bytes
caaacaudiocachebyteshr	<Double>	Read-only	This tells the hit ratio AAC bytes served
caadtsaudiocachebyteshr	<Double>	Read-only	This tells the byte hit ratio of ADTS bytes
caaplelivestrmngplvidcachebyteshr	<Double>	Read-only	This tells the byte hit ratio of AppleLive Playlist bytes
caapplelivestrmngvidcachebyteshr	<Double>	Read-only	This tells the total number of AppleLive bytes
camsftsmthstrmngplvidcabyteshr	<Double>	Read-only	This tells the byte hit ratio of MicrosoftSmoothStreaming Playlist bytes
camsftsmthstrmngvidcabyteshr	<Double>	Read-only	This tells the Bytes hit ratio of MicrosoftSmoothStreaming bytes .
caunidentifiedhr	<Double>	Read-only	This tells the hit ratio of android requests
caioshr	<Double>	Read-only	This tells the hit ratio of ios requests
caotherhr	<Double>	Read-only	This tells the hit ratio of other mobile device requests
calaptopdesktphr	<Double>	Read-only	This tells the hit ratio of laptop/desktop requests
caandroidhr	<Double>	Read-only	This tells the hit ratio of android requests
ca3gpvidhr	<Double>	Read-only	This tells the total number of 3GP requests

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cacflvvidhr	<Double>	Read-only	This tells the hit ratio of FLV requests
cacaacaudiohr	<Double>	Read-only	This tells the hit ratio of AAC requests
cacadtsaudiohr	<Double>	Read-only	This tells the Hit Ratio of ADTS requests
cacapplelivestreamingplaylistvidhr	<Double>	Read-only	This tells the hit ratio of AppleLive Playlist requests
cacapplelivestreamingvidhr	<Double>	Read-only	This tells the hit ratio of AppleLive requests
cacmsftsmthstrmplvidhr	<Double>	Read-only	This tells the hit ratio of MicrosoftSmoothStreaming Playlist requests
camsftsmthstrmvidhr	<Double>	Read-only	This tells the hit ratio of MicrosoftSmoothStreaming requests
catotother	<Double>	Read-only	
catotvideo	<Double>	Read-only	
catotaudio	<Double>	Read-only	
catotobjstored	<Double>	Read-only	
calookuphitrate	<Double>	Read-only	Rate (/s) counter for catotlookuphit
catotgetobjreq	<Double>	Read-only	
catotrequests	<Double>	Read-only	
camsftsmthstrmvid	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming requests served by NS
camsftsmthstrmvidrate	<Double>	Read-only	Rate (/s) counter for camsftsmthstrmvid
cacmsftsmthstrmvid	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming requests served from cache
cacmsftsmthstrmvidrate	<Double>	Read-only	Rate (/s) counter for cacmsftsmthstrmvid
camsftsmthstrvid	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming Playlist requests served by NS
camsftsmthstrvidrate	<Double>	Read-only	Rate (/s) counter for camsftsmthstrvid
cacmsftsmthstrmplvid	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming Playlist requests served from cache
cacmsftsmthstrmplvidrate	<Double>	Read-only	Rate (/s) counter for cacmsftsmthstrmplvid
cacaplelivestrmngvid	<Double>	Read-only	This tells the total number of AppleLive requests served by NS
cacaplelivestrmngvidrate	<Double>	Read-only	Rate (/s) counter for cacaplelivestrmngvid
cacapplelivestreamingvid	<Double>	Read-only	This tells the total number of AppleLive requests served from cache
cacapplelivestreamingvidrate	<Double>	Read-only	Rate (/s) counter for cacapplelivestreamingvid

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cacapplelivestrmngvid	<Double>	Read-only	This tells the total number of AppleLive Playlist requests served by NS
cacapplelivestrmngvidrate	<Double>	Read-only	Rate (/s) counter for cacapplelivestrmngvid
cacapplelivestreamingplaylistvid	<Double>	Read-only	This tells the total number of AppleLive Playlist requests served from cache
cacapplelivestreamingplaylistvidrate	<Double>	Read-only	Rate (/s) counter for cacapplelivestreamingplaylistvid
caadtsaudio	<Double>	Read-only	This tells the total number of ADTS requests served by NS
caadtsaudiorate	<Double>	Read-only	Rate (/s) counter for caadtsaudio
cacadtsaudio	<Double>	Read-only	This tells the total number of ADTS requests served from cache
cacadtsaudiorate	<Double>	Read-only	Rate (/s) counter for cacadtsaudio
caaacaudio	<Double>	Read-only	This tells the total number of AAC requests served by NS
caaacaudiorate	<Double>	Read-only	Rate (/s) counter for caaacaudio
cacaacaudio	<Double>	Read-only	This tells the total number of AAC requests served from cache
cacaacaudiorate	<Double>	Read-only	Rate (/s) counter for cacaacaudio
caflvvid	<Double>	Read-only	This tells the total number of FLV requests served by NS
caflvvidrate	<Double>	Read-only	Rate (/s) counter for caflvvid
cacflvvid	<Double>	Read-only	This tells the total number of FLV requests served from cache
cacflvvidrate	<Double>	Read-only	Rate (/s) counter for cacflvvid
camp4vid	<Double>	Read-only	This tells the total number of MP4 requests served by NS
camp4vidrate	<Double>	Read-only	Rate (/s) counter for camp4vid
cacmp4vid	<Double>	Read-only	This tells the total number of MP4 requests served from cache
cacmp4vidrate	<Double>	Read-only	Rate (/s) counter for cacmp4vid
ca3pvid	<Double>	Read-only	This tells the total number of 3GP requests served by NS
ca3pvidrate	<Double>	Read-only	Rate (/s) counter for ca3pvid
ca3gpvid	<Double>	Read-only	This tells the hit ratio of 3GP requests served from cache
ca3gpvidrate	<Double>	Read-only	Rate (/s) counter for ca3gpvid
camsftsmthstrmvidbytes	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming bytes served by NS
camsftsmthstrmvidbytesrate	<Double>	Read-only	Rate (/s) counter for camsftsmthstrmvidbytes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

camicrosoftsmoothstreamingvidcachebytes **Name**	<Double> **Data Type**	Read-only **Permissions**	This tells the total number of MicrosoftSmoothStreaming bytes served from cache. **Description**
camicrosoftsmoothstreamingvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for camicrosoftsmoothstreamingvidcachebytes
camsftsmthstrmplvidbytes	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming Playlist bytes served by NS
camsftsmthstrmplvidbytesrate	<Double>	Read-only	Rate (/s) counter for camsftsmthstrmplvidbytes
camicrosoftsmoothstreamingplaylistvidcachebytes	<Double>	Read-only	This tells the total number of MicrosoftSmoothStreaming Playlist bytes served from cache
camicrosoftsmoothstreamingplaylistvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for camicrosoftsmoothstreamingplaylistvidcachebytes
caapplelivestreamingvidbytes	<Double>	Read-only	This tells the total number of AppleLive bytes served by NS
caapplelivestreamingvidbytesrate	<Double>	Read-only	Rate (/s) counter for caapplelivestreamingvidbytes
caapplelivestreamingvidcachebytes	<Double>	Read-only	This tells the total number of AppleLive bytes served from cache
caapplelivestreamingvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for caapplelivestreamingvidcachebytes
caapplelivestreamingplaylistvidbytes	<Double>	Read-only	This tells the total number of AppleLive Playlist bytes served by NS
caapplelivestreamingplaylistvidbytesrate	<Double>	Read-only	Rate (/s) counter for caapplelivestreamingplaylistvidbytes
caapplelivestreamingplaylistvidcachebytes	<Double>	Read-only	This tells the total number of AppleLive Playlist bytes served from cache
caapplelivestreamingplaylistvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for caapplelivestreamingplaylistvidcachebytes
caadtsaudiobytes	<Double>	Read-only	This tells the total number of ADTS bytes served by NS
caadtsaudiobytesrate	<Double>	Read-only	Rate (/s) counter for caadtsaudiobytes
caadtsaudiocachebytes	<Double>	Read-only	This tells the total number of ADTS bytes served from cache
caadtsaudiocachebytesrate	<Double>	Read-only	Rate (/s) counter for caadtsaudiocachebytes
caaacaudiobytes	<Double>	Read-only	This tells the total number of AAC bytes served by NS
caaacaudiobytesrate	<Double>	Read-only	Rate (/s) counter for caaacaudiobytes
caaacaudiocachebytes	<Double>	Read-only	This tells the total number of AAC bytes served from cache
caaacaudiocachebytesrate	<Double>	Read-only	Rate (/s) counter for caaacaudiocachebytes
caflvvidbytes	<Double>	Read-only	This tells the total number of FLV bytes served by NS
caflvvidbytesrate	<Double>	Read-only	Rate (/s) counter for caflvvidbytes
caflvvidcachebytes	<Double>	Read-only	This tells the total number of FLV bytes served from cache
caflvvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for caflvvidcachebytes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
camp4vidbytesrate	<Double>	Read-only	Rate (/s) counter for camp4vidbytes
camp4vidcachebytes	<Double>	Read-only	This tells the total number of MP4 bytes served from cache
camp4vidcachebytesrate	<Double>	Read-only	Rate (/s) counter for camp4vidcachebytes
ca3gpvidbytes	<Double>	Read-only	This tells the total number of 3GP bytes served by NS
ca3gpvidbytesrate	<Double>	Read-only	Rate (/s) counter for ca3gpvidbytes
ca3gpvidcachebytes	<Double>	Read-only	This tells the total number of 3GP bytes served from cache
ca3gpvidcachebytesrate	<Double>	Read-only	Rate (/s) counter for ca3gpvidcachebytes
caandroid	<Double>	Read-only	Total number of android requests to netscaler
caandroidrate	<Double>	Read-only	Rate (/s) counter for caandroid
calaptopdesktp	<Double>	Read-only	Total number of laptop/desktop requests to netscaler
calaptopdesktprate	<Double>	Read-only	Rate (/s) counter for calaptopdesktp
caios	<Double>	Read-only	Total number of iOs requests to netscaler
caiosrate	<Double>	Read-only	Rate (/s) counter for caios
caother	<Double>	Read-only	Total number of other mobile device requests to netscaler
caotherrate	<Double>	Read-only	Rate (/s) counter for caother
caunidentified	<Double>	Read-only	Total number of unidentified requests to netscaler
caunidentifiedrate	<Double>	Read-only	Rate (/s) counter for caunidentified
caandroidcache	<Double>	Read-only	This tells android requests served from cache
caandroidcacherate	<Double>	Read-only	Rate (/s) counter for caandroidcache
caioscache	<Double>	Read-only	This tells iOS requests served from cache
caioscacherate	<Double>	Read-only	Rate (/s) counter for caioscache
caothercache	<Double>	Read-only	This tells Other device requests served from cache
caothercacherate	<Double>	Read-only	Rate (/s) counter for caothercache
calaptopdesktpcache	<Double>	Read-only	This tells laptop/desktop requests served from cache
calaptopdesktpcacherate	<Double>	Read-only	Rate (/s) counter for calaptopdesktpcache

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
caunidentifiedcacherate	<Double>	Read-only	Rate (/s) counter for caunidentifiedcache
caandroidbytes	<Double>	Read-only	This tells the total number of Android bytes served by NS
caandroidbytesrate	<Double>	Read-only	Rate (/s) counter for caandroidbytes
caiosbytes	<Double>	Read-only	This tells the total number of IOS bytes served by NS
caiosbytesrate	<Double>	Read-only	Rate (/s) counter for caiosbytes
caotherbytes	<Double>	Read-only	This tells the total number of Other mobile device bytes served by NS
caotherbytesrate	<Double>	Read-only	Rate (/s) counter for caotherbytes
caalptopdsktpbytes	<Double>	Read-only	This tells the total number of Laptop/desktop bytes served by NS
caalptopdsktpbytesrate	<Double>	Read-only	Rate (/s) counter for caalptopdsktpbytes
caunidentifiedbytes	<Double>	Read-only	This tells the total number of unidentified device bytes served by NS
caunidentifiedbytesrate	<Double>	Read-only	Rate (/s) counter for caunidentifiedbytes
caandroididcachebytes	<Double>	Read-only	This tells the total number of Android bytes served from cache
caandroididcachebytesrate	<Double>	Read-only	Rate (/s) counter for caandroididcachebytes
caiosidcachebytes	<Double>	Read-only	This tells the total number of IOS bytes served from cache
caiosidcachebytesrate	<Double>	Read-only	Rate (/s) counter for caiosidcachebytes
caotherididcachebytes	<Double>	Read-only	This tells the total number of other device bytes served from cache
caotherididcachebytesrate	<Double>	Read-only	Rate (/s) counter for caotherididcachebytes
calaptpdektpbytes	<Double>	Read-only	This tells the total number of Laptop/desktop bytes served from cache
calaptpdektpbytesrate	<Double>	Read-only	Rate (/s) counter for calaptpdektpbytes
caaunindentifiedbytes	<Double>	Read-only	This tells the total number of unindentified device bytes served from cache
caaunindentifiedbytesrate	<Double>	Read-only	Rate (/s) counter for caaunindentifiedbytes

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/ca

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/ca?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get ca resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ca": [ {
    "caaacaudiocachebytes":<Double_value>,
    "camsftsmthstrmvidhr":<Double_value>,
    "caflvvidrate":<Double_value>,
    "cacmsftsmthstrmplvid":<Double_value>,
    "camp4vidbytesrate":<Double_value>,
    "camsftsmthstrmngplvidcabyteshr":<Double_value>,
    "caapplelivestreamingplaylistvidbytesrate":<Double_value>,
    "catotvideo":<Double_value>,
    "caaacaudiobytes":<Double_value>,
    "calookuphitrate":<Double_value>,
    "cacapplelivestreamingplaylistvid":<Double_value>,
    "cacmp4vidhr":<Double_value>,
    "cacapplelivestreamingvidhr":<Double_value>,
    "caflvvidbytesrate":<Double_value>,
    "catotobjstored":<Double_value>,
    "ca3gpvidcachebytesrate":<Double_value>,
    "caaacaudiocachebyteshr":<Double_value>,
    "caadtsaudiobytes":<Double_value>,
    "cacadtsaudio":<Double_value>,
    "caaudiorate":<Double_value>,
    "cacapplelivestreamingplaylistvidrate":<Double_value>,
    "caaplelivestrmngplvidcachebyteshr":<Double_value>,
    "caotherididcachebytes":<Double_value>,
    "calaptpdsktpbyteshr":<Double_value>,
    "caflvvidcachebytes":<Double_value>,
    "caalptopdsktpbytesrate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

"cacflvvidrate":<Double_value>,
"caaunindentifiedbyteshr":<Double_value>,
"caapplelivestreamingplaylistvidcachebytes":<Double_value>,
"catotrequests":<Double_value>,
"cacflvvidhr":<Double_value>,
"caadtsaudiocachebyteshr":<Double_value>,
"caiosbytesrate":<Double_value>,
"cacmp4vidrate":<Double_value>,
"catotlookuphit":<Double_value>,
"caadtsaudiobytesrate":<Double_value>,
"ca3pvid":<Double_value>,
"calaptopdesktpcache":<Double_value>,
"camsftsmthstrmplvidbytes":<Double_value>,
"caiosidcachebytesrate":<Double_value>,
"ca3gpvidbytesrate":<Double_value>,
"caunidentifiedcache":<Double_value>,
"caapplelivestreamingvidcachebytesrate":<Double_value>,
"caapplelivestreamingvidbytes":<Double_value>,
"caflvvidcachebyteshr":<Double_value>,
"caaunindentifiedbytesrate":<Double_value>,
"caios":<Double_value>,
"caiosrate":<Double_value>,
"caandroidbytes":<Double_value>,
"caotherrate":<Double_value>,
"caflvvidbytes":<Double_value>,
"calaptpdektpbytes":<Double_value>,
"caapplelivestreamingplaylistvidcachebytesrate":<Double_value>,
"camsftsmthstrmplvidbytesrate":<Double_value>,
"caaacaudiorate":<Double_value>,
"ca3gpvidhr":<Double_value>,
"calaptopdesktprate":<Double_value>,
"camp4vidrate":<Double_value>,
"cacadtsaudiohr":<Double_value>,
"cacapplelivestrmngvid":<Double_value>,
"caaacaudiocachebytesrate":<Double_value>,
"caapplelivestreamingvidbytesrate":<Double_value>,
"cacaplelivestrmngvidrate":<Double_value>,
"caunidentifiedhr":<Double_value>,
"cagetobjreqrate":<Double_value>,
"camp4vidcachebyteshr":<Double_value>,
"calaptopdesktphr":<Double_value>,
"camsftsmthstrmvidbytesrate":<Double_value>,
"camsftsmthstrmngvidcabyteshr":<Double_value>,
"camicrosoftsmoothstreamingplaylistvidcachebytes":<Double_value>,
"caiosbytes":<Double_value>,
"calaptopdesktp":<Double_value>,
"caothercacherate":<Double_value>,
"camp4vidcachebytesrate":<Double_value>,
"cacaacaudiohr":<Double_value>,
"caotherididcachebytesrate":<Double_value>,
"camsftsmthstrmvidbytes":<Double_value>,
"cacmsftsmthstrmplvidhr":<Double_value>,
"calaptopdesktpcacherate":<Double_value>,
"ca3gpvidbytes":<Double_value>,
"caotherbytes":<Double_value>,
"carequestsrate":<Double_value>,
"caunidentifiedbytesrate":<Double_value>,
"ca3gpvidcachebyteshr":<Double_value>,
"caadtsaudiocachebytes":<Double_value>,
"caadtsaudio":<Double_value>,
"caadtsaudiocachebytesrate":<Double_value>,
"ca3gpvidrate":<Double_value>,
"caioscache":<Double_value>,
"caaunindentifiedbytes":<Double_value>,
"cacapplelivestreamingplaylistvidhr":<Double_value>,
"caunidentified":<Double_value>,
"caandroid":<Double_value>,
"caapplelivestreamingplaylistvidbytes":<Double_value>,
"camp4vidbytes":<Double_value>,
"caunidentifiedrate":<Double_value>,
"cacaacaudio":<Double_value>,
"cacflvvid":<Double_value>,
"catotgetobjreq":<Double_value>,
"caotherbyteshr":<Double_value>,
"caioshr":<Double_value>,
"calaptpdektpbytesrate":<Double_value>,
"caandroididcachebytes":<Double_value>,
"caflvvid":<Double_value>,
"cacadtsaudiorate":<Double_value>,
"caandroidbyteshr":<Double_value>,
"caandroididcachebytesrate":<Double_value>,
"cacmp4vid":<Double_value>,
"caandroidbytesrate":<Double_value>,
"cacmsftsmthstrmplvidrate":<Double_value>,
"caadtsaudiorate":<Double_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "camp4vid":<Double_value>,
    "caunidentifiedcacherate":<Double_value>,
    "caandroidcacherate":<Double_value>,
    "cavideorate":<Double_value>,
    "camicrosoftsmoothstreamingplaylistvidcachebytesrate":<Double_value>,
    "caaacaudio":<Double_value>,
    "catotother":<Double_value>,
    "cacapplelivestreamingvidrate":<Double_value>,
    "cacapplelivestreamingvid":<Double_value>,
    "caandroidhr":<Double_value>,
    "cacmsftsmthstrmvidrate":<Double_value>,
    "caioscacherate":<Double_value>,
    "caunidentifiedbytes":<Double_value>,
    "caandroidrate":<Double_value>,
    "cacaacaudiorate":<Double_value>,
    "cacmsftsmthstrmvid":<Double_value>,
    "camicrosoftsmoothstreamingvidcachebytes":<Double_value>,
    "caother":<Double_value>,
    "caalptopdsktpbytes":<Double_value>,
    "caothercache":<Double_value>,
    "camsftsmthstrvid":<Double_value>,
    "camp4vidcachebytes":<Double_value>,
    "ca3gpvid":<Double_value>,
    "cacapplelivestrmngvidrate":<Double_value>,
    "camicrosoftsmoothstreamingvidcachebytesrate":<Double_value>,
    "catoherrate":<Double_value>,
    "caotherhr":<Double_value>,
    "camsftsmthstrmvidrate":<Double_value>,
    "camsftsmthstrvidrate":<Double_value>,
    "ca3gpvidcachebytes":<Double_value>,
    "caaacaudiobytesrate":<Double_value>,
    "camsftsmthstrmvid":<Double_value>,
    "cacaplelivestrmngvid":<Double_value>,
    "caapplelivestrmngvidcachebyteshr":<Double_value>,
    "ca3pvidrate":<Double_value>,
    "caandroidcache":<Double_value>,
    "caflvvidcachebytesrate":<Double_value>,
    "caiosidcachebytes":<Double_value>,
    "caotherbytesrate":<Double_value>,
    "caapplelivestreamingvidcachebytes":<Double_value>,
    "caiosbyteshr":<Double_value>,
    "catotaudio":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Integrated Caching

May 20, 2016

Integrated Caching statistics. The Integrated Caching feature is used to cache static and dynamic web application data.

Name	Description
cache	Statistics for cache.
cachecontentgroup	Statistics for Integrated Cache content group resource.
cachepolicy	Statistics for Integrated Cache policy resource.
cachepolicylabel	Statistics for cache policy label resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cache

May 20, 2016

Statistics for cache.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
cachemaxmemorykb	<Double>	Read-only	Largest amount of memory the NetScaler can dedicate to caching, up to 50% of available memory. A 0 value disables caching, but the caching module continues to run.
cacherecentpercentsuccessfulrevalidation	<Double>	Read-only	Recently recorded percentage of times stored content was successfully revalidated by a 304 response rather than by a full response
cacherecentpercentstoreablemiss	<Double>	Read-only	Recently recorded ratio of store-able misses to all misses expressed as percentage.
cacherecentpercentparameterizedhits	<Double>	Read-only	Recently recorded ratio of parameterized 304 hits to all parameterized hits expressed as a percentage
cacherecentpercentoriginbandwidthsaved	<Double>	Read-only	Bytes served from cache divided by total bytes served to client. This ratio can be greater than 1 because of the assumption that all compression has been done in the NetScaler.
cacherecentpercenthit	<Double>	Read-only	Recently recorded cache hit ratio expressed as percentage
cacherecentpercentbytehit	<Double>	Read-only	Recently recorded cache byte hit ratio expressed as percentage. Here we define byte hit ratio as ((number of bytes served from the cache)/(total number of bytes served to the client)). This is the standard definition of Byte Hit Ratio. If compression is turned ON in NS then this ratio doesn't mean much. This might under or over estimate the origin-to-cache bandwidth saving (depending upon whether bytes served by CMP in NetScaler are more or less than compressed bytes served from the cache). If CMP is turned OFF in NS then this ratio is same as cacheRecentPercentOriginBandwidthSaved.
cacherecentpercent304hits	<Double>	Read-only	Recently recorded ratio of 304 hits to all hits expressed as percentage
cacheutilizedmemorykb	<Double>	Read-only	Amount of memory the integrated cache is currently using.
cachemaxmemoryactivekb	<Double>	Read-only	Currently active value of maximum memory
cache64maxmemorykb	<Double>	Read-only	Largest amount of memory the NetScaler can dedicate to caching, up to 50% of available memory. A 0 value disables caching, but the caching module continues to run.
cachepercentpethits	<Double>	Read-only	Percentage of cache hits in content groups that have Poll Every Time enabled, relative to all searches of content groups with Poll Every Time enabled.
cachetotpethits	<Double>	Read-only	Number of times a cache hit was found during a search of a content group that has Poll Every Time enabled.
cachepethitsrate	<Double>	Read-only	Rate (/s) counter for cachetotpethits
cachepercentparameterized304hits	<Double>	Read-only	Percentage of parameterized 304 hits relative to all parameterized hits.
cachetotparameterizedhits	<Double>	Read-only	Parameterized requests resulting in either a 304 or non-304 hit.
cacheparameterizedhitsrate	<Double>	Read-only	Rate (/s) counter for cachetotparameterizedhits
cachepercentsuccessfulrevalidation	<Double>	Read-only	Percentage of times stored content was successfully revalidated by a 304 (Object Not Modifed) response rather than by a full response
cachepercentstoreablemiss	<Double>	Read-only	Responses that were fetched from the origin, stored in the cache, and then served to the client, as a percentage of all cache misses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cachetotfulltoconditionalrequest	<Double>	Read-only	conditional requests.
cachefulltoconditionalrequestrate	<Double>	Read-only	Rate (/s) counter for cachetotfulltoconditionalrequest
cachetotsuccessfulrevalidation	<Double>	Read-only	Total number of times stored content was successfully revalidated by a 304 Not Modified response from the origin.
cachesuccessfulrevalidationrate	<Double>	Read-only	Rate (/s) counter for cachetotsuccessfulrevalidation
cachetotrevalidationmiss	<Double>	Read-only	Responses that an intervening cache revalidated with the integrated cache before serving, as determined by a Cache-Control: Max-Age header configurable in the integrated cache
cacherevalidationmissrate	<Double>	Read-only	Rate (/s) counter for cachetotrevalidationmiss
cachetotnonstoreablemisses	<Double>	Read-only	Cache misses for which the fetched response is not stored in the cache. These responses match policies with a NOCACHE action or are affected by Poll Every Time.
cachenonstoreablemissesrate	<Double>	Read-only	Rate (/s) counter for cachetotnonstoreablemisses
cachetotstoreablemisses	<Double>	Read-only	Cache misses for which the fetched response is stored in the cache before serving it to the client. Storable misses conform to a built-in or user-defined caching policy that contains a CACHE action.
cachestoreablemissesrate	<Double>	Read-only	Rate (/s) counter for cachetotstoreablemisses
cachecompressedbytesserved	<Double>	Read-only	Number of compressed bytes served from the cache
cachecompressedbytesservedrate	<Double>	Read-only	Rate (/s) counter for cachecompressedbytesserved
cachepercentbytehit	<Double>	Read-only	Bytes served from the cache divided by total bytes served to the client. If compression is On in the NetScaler, this ratio may not reflect the bytes served by the compression module. If the compression is Off, this ratio is the same as cachePercentOriginBandwidthSaved.
cachebytesserved	<Double>	Read-only	Total number of bytes served from the integrated cache
cachebytesservedrate	<Double>	Read-only	Rate (/s) counter for cachebytesserved
cachetotresponsebytes	<Double>	Read-only	Total number of HTTP response bytes served by NetScaler from both the origin and the cache
cacheresponsebytesrate	<Double>	Read-only	Rate (/s) counter for cachetotresponsebytes
cachepercent304hits	<Double>	Read-only	304 responses as a percentage of all responses that the NetScaler served.
cachenummarker	<Double>	Read-only	Marker objects created when a response exceeds the maximum or minimum size for entries in its content group or has not yet received the minimum number of hits required for items in its content group.
cachepercentoriginbandwidthsaved	<Double>	Read-only	Percentage of origin bandwidth saved, expressed as number of bytes served from the integrated cache divided by all bytes served. The assumption is that all compression is done in the NetScaler.
cachepercenthit	<Double>	Read-only	Cache hits as percentage of the total number of requests
cachetotmisses	<Double>	Read-only	Intercepted HTTP requests requiring fetches from origin server.
cachemissesrate	<Double>	Read-only	Rate (/s) counter for cachetotmisses
cachetothits	<Double>	Read-only	Responses served from the integrated cache. These responses match a policy with a CACHE action.
cachehitsrate	<Double>	Read-only	Rate (/s) counter for cachetothits
cachetotrequests	<Double>	Read-only	Total cache hits plus total cache misses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cachenumcached	<Double>	Read-only	Responses currently in integrated cache. Includes responses fully downloaded, in the process of being downloaded, and expired or flushed but not yet removed.
cachecurhits	<Double>	Read-only	This number should be close to the number of hits being served currently.
cachecurmisses	<Double>	Read-only	Responses fetched from the origin and served from the cache. Should approximate storable misses. Does not include non-storable misses.
cachetotnon304hits	<Double>	Read-only	Total number of full (non-304) responses served from the cache. A 304 status code indicates that a response has not been modified since the last time it was served
cachenon304hitsrate	<Double>	Read-only	Rate (/s) counter for cachetotnon304hits
cachetot304hits	<Double>	Read-only	Object not modified responses served from the cache. (Status code 304 served instead of the full response.)
cache304hitsrate	<Double>	Read-only	Rate (/s) counter for cachetot304hits
cachetotsqlhits	<Double>	Read-only	sql response served from cache
cachesqlhitsrate	<Double>	Read-only	Rate (/s) counter for cachetotsqlhits
cachetotexpireatlastbyte	<Double>	Read-only	Instances of content expiring immediately after receiving the last body byte due to the Expire at Last Byte setting for the content group.
cacheexpireatlastbyterate	<Double>	Read-only	Rate (/s) counter for cachetotexpireatlastbyte
cachetotflashcachemisses	<Double>	Read-only	Number of requests to a content group with flash cache enabled that were cache misses. Flash cache distributes the response to all the clients in aqueue.
cacheflashcachemissesrate	<Double>	Read-only	Rate (/s) counter for cachetotflashcachemisses
cachetotflashcachehits	<Double>	Read-only	Number of requests to a content group with flash cache enabled that were cache hits. The flash cache setting queues requests that arrive simultaneously and distributes the response to all the clients in the queue.
cacheflashcachehitsrate	<Double>	Read-only	Rate (/s) counter for cachetotflashcachehits
cachetotparameterizedinvalidationrequests	<Double>	Read-only	Requests matching a policy with an invalidation (INVAL) action and a content group that uses an invalidation selector or parameters.
cacheparameterizedinvalidationrequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotparameterizedinvalidationrequests
cachetotnonparameterizedinvalidationrequests	<Double>	Read-only	Requests that match an invalidation policy where the invalGroups parameter is configured and expires one or more content groups.
cachenonparameterizedinvalidationrequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotnonparameterizedinvalidationrequests
cachetotinvalidationrequests	<Double>	Read-only	Requests that match an invalidation policy and result in expiration of specific cached responses or entire content groups.
cacheinvalidationrequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotinvalidationrequests
cachetotparameterizedrequests	<Double>	Read-only	Total number of requests where the content group has hit and invalidation parameters or selectors.
cacheparameterizedrequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotparameterizedrequests
cachetotparameterizednon304hits	<Double>	Read-only	Parameterized requests resulting in a full response (not status code 304: Object Not Updated) served from the cache.
cacheparameterizednon304hitsrate	<Double>	Read-only	Rate (/s) counter for cachetotparameterizednon304hits
cachetotparameterized304hits	<Double>	Read-only	Parameterized requests resulting in an object not modified (status code 304) response.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
cacheparameterized304hitsrate	<Double>	Read-only	Rate (/s) counter for cachetotparameterized304hits
cachetotpetrequests	<Double>	Read-only	Requests that triggered a search of a content group that has Poll Every Time (PET) enabled (always consult the origin server before serving cached data).
cachepetrequestsrate	<Double>	Read-only	Rate (/s) counter for cachetotpetrequests
cacheerrmemalloc	<Double>	Read-only	Total number of times the cache failed to allocate memory to store responses.
cachelargestresponsereceived	<Double>	Read-only	Size, in bytes, of largest response sent to client from the cache or the origin server.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/cache

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/cache?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get cache resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:


```
{ "cache": [ {
    "cachehitsrate":<Double_value>,
    "cachepethitsrate":<Double_value>,
    "cachetotexpireatlastbyte":<Double_value>,
    "cachenummarker":<Double_value>,
    "cachecurhits":<Double_value>,
    "cacheresponsebytesrate":<Double_value>,
    "cacheutilizedmemorykb":<Double_value>,
    "cachetotnonparameterizedinvalidationrequests":<Double_value>,
    "cachepercentpethits":<Double_value>,
    "cachebytesserved":<Double_value>,
    "cachetotparameterizednon304hits":<Double_value>,
    "cachebytesservedrate":<Double_value>,
    "cachepercent304hits":<Double_value>,
    "cachepercentbytehit":<Double_value>,
    "cachepercentstoreablemiss":<Double_value>,
    "cachetotpetrequests":<Double_value>,
    "cachecompressedbytesserved":<Double_value>,
    "cachetotresponsebytes":<Double_value>,
    "cachelargestresponsereceived":<Double_value>,
    "cachenonparameterizedinvalidationrequestsrate":<Double_value>,
    "cacherecentpercentbytehit":<Double_value>,
    "cacherecentpercentsuccessfulrevalidation":<Double_value>,
    "cachesuccessfulrevalidationrate":<Double_value>,
    "cachesqlhitsrate":<Double_value>,
    "cacherequestsrate":<Double_value>,
    "cachemissesrate":<Double_value>,
    "cachemaxmemorykb":<Double_value>,
    "cachenumcached":<Double_value>,
    "cachetothits":<Double_value>,
    "cachetotsuccessfulrevalidation":<Double_value>,
    "cachepercentparameterized304hits":<Double_value>,
    "cacheflashcachehitsrate":<Double_value>,
    "cachestoreablemissesrate":<Double_value>,
    "cachetotflashcachehits":<Double_value>,
    "cachetotsqlhits":<Double_value>,
    "cachetotrevalidationmiss":<Double_value>,
    "cachemaxmemoryactivekb":<Double_value>,
    "cachetotstoreablemisses":<Double_value>,
    "cacheparameterizedhitsrate":<Double_value>,
    "cacheparameterized304hitsrate":<Double_value>,
    "cacheexpireatlastbyterate":<Double_value>,
    "cachecurmisses":<Double_value>,
    "cacherevalidationmissrate":<Double_value>,
    "cachenon304hitsrate":<Double_value>,
    "cachepercentsuccessfulrevalidation":<Double_value>,
    "cachetotnon304hits":<Double_value>,
    "cachefulltoconditionalrequestrate":<Double_value>,
    "cacherecentpercentoriginbandwidthsaved":<Double_value>,
    "cacherecentpercentparameterizedhits":<Double_value>,
    "cachetotparameterizedinvalidationrequests":<Double_value>,
    "cachetotflashcachemisses":<Double_value>,
    "cacheparameterizedrequestsrate":<Double_value>,
    "cacheerrmemalloc":<Double_value>,
    "cacherecentpercenthit":<Double_value>,
    "cachecompressedbytesservedrate":<Double_value>,
    "cache304hitsrate":<Double_value>,
    "cachepercenthit":<Double_value>,
    "cachenonstoreablemissesrate":<Double_value>,
    "cachetotinvalidationrequests":<Double_value>,
    "cache64maxmemorykb":<Double_value>,
    "cachetot304hits":<Double_value>,
    "cachetotfulltoconditionalrequest":<Double_value>,
    "cachetotparameterizedhits":<Double_value>,
    "cachepetrequestsrate":<Double_value>,
    "cacherecentpercent304hits":<Double_value>,
    "cachetotnonstoreablemisses":<Double_value>,
    "cachetotparameterizedrequests":<Double_value>,
    "cachepercentoriginbandwidthsaved":<Double_value>,
    "cachetotrequests":<Double_value>,
    "cacheparameterizednon304hitsrate":<Double_value>,
    "cachetotmisses":<Double_value>,
    "cachetotpethits":<Double_value>,
    "cacheparameterizedinvalidationrequestsrate":<Double_value>,
    "cacherecentpercentstoreablemiss":<Double_value>,
    "cacheflashcachemissesrate":<Double_value>,
    "cacheinvalidationrequestsrate":<Double_value>,
    "cachetotparameterized304hits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachecontentgroup

May 20, 2016

Statistics for Integrated Cache content group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the cache contentgroup for which to display statistics. If you do not set this parameter, statistics are shown for all cache contentgroups. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
groupnon304hit	<Double>	Read-only	Non304 hits for ContentGroup
group304hit	<Double>	Read-only	304 hits for ContentGroup
totcell	<Double>	Read-only	Number of objects in contentgroup
totmarkercell	<Double>	Read-only	Number of marker objects in contentgroup
timesflushed	<Double>	Read-only	Number of times contentgroup is flushed
totmemory	<Double>	Read-only	current memory usage
maxmemory	<Double>	Read-only	maximum memory usage limit

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/cachecontentgroup

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/cachecontentgroup?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<**

Use this query-parameter to get cachecontentgroup resources based on additional properties.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachecontentgroup": [ {
      "name":<String_value>,
      "group304hit":<Double_value>,
      "totmarkercell":<Double_value>,
      "timesflushed":<Double_value>,
      "totmemory":<Double_value>,
      "totcell":<Double_value>,
      "groupnon304hit":<Double_value>,
      "maxmemory":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/cachecontentgroup/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachecontentgroup": [ {
      "name":<String_value>,
      "group304hit":<Double_value>,
      "totmarkercell":<Double_value>,
      "timesflushed":<Double_value>,
      "totmemory":<Double_value>,
      "totcell":<Double_value>,
      "groupnon304hit":<Double_value>,
      "maxmemory":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicy

May 20, 2016

Statistics for Integrated Cache policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the cache policy for which to display statistics. If you do not set this parameter, statistics are shown for all cache policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/cachepolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/cachepolicy?**args=policyname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<St**

Use this query-parameter to get cachepolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicy": [ {
    "policyname":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/cachepolicy/policyname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicy": [ {
    "policyname":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cachepolicylabel

Jul 07, 2017
Statistics for cache policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the cache-policy label for which to display statistics. If you do not set this parameter statistics are shown for all cache-policy labels. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/cachepolicylabel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/cachepolicylabel?**args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats**

Use this query-parameter to get cachepolicylabel resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicylabel": [ {
      "labelname":<String_value>,
      "pipolicylabelhitsrate":<Double_value>,
      "pipolicylabelhits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/cachepolicylabel/labelname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cachepolicylabel": [ {
      "labelname":<String_value>,
      "pipolicylabelhitsrate":<Double_value>,
      "pipolicylabelhits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cluster

May 20, 2016

Cluster commands.

Name	Description
clusterinstance	Statistics for cluster instance resource.
clusternode	Statistics for cluster node resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusterinstance

Jul 07, 2017
Statistics for cluster instance resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clid	<Double>	Read-write	ID of the cluster instance for which to display statistics. Minimum value = 1 Maximum value = 16
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
clnumnodes	<Double>	Read-only	Number of nodes in the cluster.
clcurstatus	<String>	Read-only	State of the cluster.
clviewleader	<String>	Read-only	NSIP address of the Configuration Coordinator of the cluster.
totsteeredpkts	<Double>	Read-only	Total number of packets steered on the cluster backplane.
clbkplanerx	<Double>	Read-only	Traffic received on backplane (in mbits)
clbkplanerxrate	<Double>	Read-only	Rate (/s) counter for clbkplanerx
clbkplanetx	<Double>	Read-only	Traffic transmitted from backplane (in mbits)
clbkplanetxrate	<Double>	Read-only	Rate (/s) counter for clbkplanetx
numdfddroppkts	<Double>	Read-only	Number of steered packets that are dropped.
totpropagationtimeout	<Double>	Read-only	Number of times the update to the client timed-out.

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/clusterinstance

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/clusterinstance?**args=clid:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String**

Use this query-parameter to get clusterinstance resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusterinstance": [ {
    "clid":<Double_value>,
    "totpropagationtimeout":<Double_value>,
    "clbkplanetx":<Double_value>,
    "clbkplanetxrate":<Double_value>,
    "totsteeredpkts":<Double_value>,
    "numdfddroppkts":<Double_value>,
    "clviewleader":<String_value>,
    "clbkplanerx":<Double_value>,
    "clcurstatus":<String_value>,
    "clnumnodes":<Double_value>,
    "clbkplanerxrate":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/clusterinstance/clid_value><Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusterinstance": [ {
    "clid":<Double_value>,
    "totpropagationtimeout":<Double_value>,
    "clbkplanetx":<Double_value>,
    "clbkplanetxrate":<Double_value>,
    "totsteeredpkts":<Double_value>,
    "numdfddroppkts":<Double_value>,
    "clviewleader":<String_value>,
    "clbkplanerx":<Double_value>,
    "clcurstatus":<String_value>,
    "clnumnodes":<Double_value>,
    "clbkplanerxrate":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

clusternode

May 20, 2016

Statistics for cluster node resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
nodeid	<Double>	Read-write	ID of the cluster node for which to display statistics. If an ID is not provided, statistics are shown for all nodes. Minimum value = 0 Maximum value = 31
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
clsyncstate	<String>	Read-only	Sync state of the cluster node.
clnodeeffectivehealth	<String>	Read-only	Health of the cluster node.
clnodeip	<String>	Read-only	NSIP address of the cluster node.
clmasterstate	<String>	Read-only	Operational state of the cluster node.
cltothbtx	<Double>	Read-only	Number of heartbeats sent. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
cltothbrx	<Double>	Read-only	Number of heartbeats received. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
nnmcurconn	<Double>	Read-only	Number of connections open for node-to-node communication.
nnmtotconntx	<Double>	Read-only	Number of node-to-node messages sent. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
nnmtotconnrx	<Double>	Read-only	Number of node-to-node messages received. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
clptpstate	<String>	Read-only	PTP state of the node. This state is Master for one node and Slave for the rest. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows UNKNOWN. When executed from the cluster IP address, shows all the statistics.
clptptx	<Double>	Read-only	Number of PTP packets transmitted by the node. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
clptprx	<Double>	Read-only	Number of PTP packets received on the node. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.
nnmerrmsend	<Double>	Read-only	Number of errors in sending node-to-node multicast/broadcast messages. When executed from the NSIP address, shows the statistics for local node only. For remote node it shows a value of 0. When executed from the cluster IP address, shows all the statistics.

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/clusternode

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/clusternode?**args=nodeid:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<Strin**

Use this query-parameter to get clusternode resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternode": [ {
    "nodeid":<Double_value>,
    "clnodeip":<String_value>,
    "clsyncstate":<String_value>,
    "nnmcurconn":<Double_value>,
    "nnmerrmsend":<Double_value>,
    "clnodeeffectivehealth":<String_value>,
    "nnmtotconnrx":<Double_value>,
    "cltothbrx":<Double_value>,
    "clptprx":<Double_value>,
    "nnmtotconntx":<Double_value>,
    "clmasterstate":<String_value>,
    "clptptx":<Double_value>,
    "clptpstate":<String_value>,
    "cltothbtx":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/clusternode/nodeid_value><Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "clusternode": [ {
    "nodeid":<Double_value>,
    "clnodeip":<String_value>,
    "clsyncstate":<String_value>,
    "nnmcurconn":<Double_value>,
    "nnmerrmsend":<Double_value>,
    "clnodeeffectivehealth":<String_value>,
    "nnmtotconnrx":<Double_value>,
    "cltothbrx":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "clptprx":<Double_value>,
    "nnmtotconntx":<Double_value>,
    "clmasterstate":<String_value>,
    "clptptx":<Double_value>,
    "clptpstate":<String_value>,
    "cltothbtx":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Compression

Jul 07, 2017

Compression statistics. The system's feature for compressing HTTP responses to compression-aware browsers.

Name	Description
cmp	Statistics for cmp.
cmppolicy	Statistics for compression policy resource.
cmppolicylabel	Statistics for compression policy label resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmp

May 20, 2016
Statistics for cmp.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
delbwsaving	<Double>	Read-only	Bandwidth saving from delta compression expressed as percentage.
delcmpratio	<Double>	Read-only	Ratio of compressible data received to compressed data transmitted.If this ratio is one (uncmp:1.0) that means compression is disabled or we are not able to compress even a single compressible packet.
decomptcpratio	<Double>	Read-only	Ratio of decompressed data transmitted to compressed data received.
decomptcpbandwidthsaving	<Double>	Read-only	Bandwidth saving from TCP compression expressed as percentage.
comptcpratio	<Double>	Read-only	Ratio of compressible data received to compressed data transmitted.If this ratio is one (uncmp:1.0) that means compression is disabled or we are not able to compress even a single compressible packet.
comptcpbandwidthsaving	<Double>	Read-only	Bandwidth saving from TCP compression expressed as percentage.
comptotaldatacompressionratio	<Double>	Read-only	Ratio of total HTTP data received to total HTTP data transmitted.
comphttpbandwidthsaving	<Double>	Read-only	Bandwidth saving from TCP compression expressed as percentage.
compratio	<Double>	Read-only	Ratio of the compressible data received from the server to the compressed data sent to the client.
comptotalrequests	<Double>	Read-only	Number of HTTP compression requests the NetScaler receives for which the response is successfully compressed. For example, after you enable compression and configure services, if you send requests to the NetScaler with the following header information: ?Accept-Encoding: gzip, deflate?, and NetScaler compresses the corresponding response, this counter is incremented.
comprequestsrate	<Double>	Read-only	Rate (/s) counter for comptotalrequests
comptotalrxbytes	<Double>	Read-only	Number of bytes that can be compressed, which the NetScaler receives from the server. This gives the content length of the response that the NetScaler receives from server.
comprxbytesrate	<Double>	Read-only	Rate (/s) counter for comptotalrxbytes
comptotaltxbytes	<Double>	Read-only	Number of bytes the NetScaler sends to the client after compressing the response from the server.
comptxbytesrate	<Double>	Read-only	Rate (/s) counter for comptotaltxbytes
comptotalrxpackets	<Double>	Read-only	Number of HTTP packets that can be compressed, which the NetScaler receives from the server.
comprxpacketsrate	<Double>	Read-only	Rate (/s) counter for comptotalrxpackets
comptotaltxpackets	<Double>	Read-only	Number of HTTP packets that the NetScaler sends to the client after compressing the response from the server.
comptxpacketsrate	<Double>	Read-only	Rate (/s) counter for comptotaltxpackets
			Number of bytes that can be compressed, which the NetScaler receives from the server. This gives the content length of the

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
comptcprxbytesrate	<Double>	Read-only	Rate (/s) counter for comptcptotalrxbytes
comptcptotalrxpackets	<Double>	Read-only	Total number of compressible packets received by NetScaler.
comptcprxpacketsrate	<Double>	Read-only	Rate (/s) counter for comptcptotalrxpackets
comptcptotaltxbytes	<Double>	Read-only	Number of bytes that the NetScaler sends to the client after compressing the response from the server.
comptcptxbytesrate	<Double>	Read-only	Rate (/s) counter for comptcptotaltxbytes
comptcptotaltxpackets	<Double>	Read-only	Number of TCP packets that the NetScaler sends to the client after compressing the response from the server.
comptcptxpacketsrate	<Double>	Read-only	Rate (/s) counter for comptcptotaltxpackets
comptcptotalquantum	<Double>	Read-only	Number of times the NetScaler compresses a quantum of data. NetScaler buffers the data received from the server till it reaches the quantum size and then compresses the buffered data and transmits to the client.
comptcpquantumrate	<Double>	Read-only	Rate (/s) counter for comptcptotalquantum
comptcptotalpush	<Double>	Read-only	Number of times the NetScaler compresses data on receiving a TCP PUSH flag from the server. The PUSH flag ensures that data is compressed immediately without waiting for the buffered data size to reach the quantum size.
comptcppushrate	<Double>	Read-only	Rate (/s) counter for comptcptotalpush
comptcptotaleoi	<Double>	Read-only	Number of times the NetScaler compresses data on receiving End Of Input (FIN packet). When the NetScaler receives End Of Input (FIN packet), it compresses the buffered data immediately without waiting for the buffered data size to reach the quantum size.
comptcpeoirate	<Double>	Read-only	Rate (/s) counter for comptcptotaleoi
comptcptotaltimer	<Double>	Read-only	Number of times the NetScaler compresses data on expiration of data accumulation timer. The timer expires if the server response is very slow and consequently, the NetScaler does not receive response for a certain amount of time. Under such a condition, the NetScaler compresses the buffered data immediately without waiting for the buffered data size to reach the quantum size.
comptcptimerrate	<Double>	Read-only	Rate (/s) counter for comptcptotaltimer
decomptcprxbytes	<Double>	Read-only	Total number of compressed bytes received by NetScaler.
decomptcprxbytesrate	<Double>	Read-only	Rate (/s) counter for decomptcprxbytes
decomptcprxpackets	<Double>	Read-only	Total number of compressed packets received by NetScaler.
decomptcprxpacketsrate	<Double>	Read-only	Rate (/s) counter for decomptcprxpackets
decomptcptxbytes	<Double>	Read-only	Total number of decompressed bytes transmitted by NetScaler.
decomptcptxbytesrate	<Double>	Read-only	Rate (/s) counter for decomptcptxbytes
decomptcptxpackets	<Double>	Read-only	Total number of decompressed packets transmitted by NetScaler.
decomptcptxpacketsrate	<Double>	Read-only	Rate (/s) counter for decomptcptxpackets
decomptcperrdata	<Double>	Read-only	Number of data errors encountered while decompressing.
decomptcperrdatarate	<Double>	Read-only	Rate (/s) counter for decomptcperrdata

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
decomptcperrlessdata	<Double>	Read-only	Number of times NetScaler received less data than declared by protocol.
decomptcperrlessdatarate	<Double>	Read-only	Rate (/s) counter for decomptcperrlessdata
decomptcperrmoredata	<Double>	Read-only	Number of times NetScaler received more data than declared by protocol.
decomptcperrmoredatarate	<Double>	Read-only	Rate (/s) counter for decomptcperrmoredata
decomptcperrmemory	<Double>	Read-only	Number of times memory failures occurred while decompressing.
decomptcperrmemoryrate	<Double>	Read-only	Rate (/s) counter for decomptcperrmemory
decomptcperrunknown	<Double>	Read-only	Number of times unknown errors occurred while decompressing.
decomptcperrunknownrate	<Double>	Read-only	Rate (/s) counter for decomptcperrunknown
delcomptotalrequests	<Double>	Read-only	Total number of delta compression requests received by NetScaler.
delcomprequestsrate	<Double>	Read-only	Rate (/s) counter for delcomptotalrequests
delcompdone	<Double>	Read-only	Total number of delta compressions done by NetScaler.
delcompdonerate	<Double>	Read-only	Rate (/s) counter for delcompdone
delcomptcprxbytes	<Double>	Read-only	Total number of delta-compressible bytes received by NetScaler.
delcomptcprxbytesrate	<Double>	Read-only	Rate (/s) counter for delcomptcprxbytes
delcomptcptxbytes	<Double>	Read-only	Total number of delta-compressed bytes transmitted by NetScaler.
delcomptcptxbytesrate	<Double>	Read-only	Rate (/s) counter for delcomptcptxbytes
delcompfirstaccess	<Double>	Read-only	Total number of delta compression first accesses.
delcompfirstaccessrate	<Double>	Read-only	Rate (/s) counter for delcompfirstaccess
delcomptcprxpackets	<Double>	Read-only	Number of delta-compressible packets received.
delcomptcprxpacketsrate	<Double>	Read-only	Rate (/s) counter for delcomptcprxpackets
delcomptcptxpackets	<Double>	Read-only	Total number of delta-compressed packets transmitted by NetScaler.
delcomptcptxpacketsrate	<Double>	Read-only	Rate (/s) counter for delcomptcptxpackets
delcompbaseserved	<Double>	Read-only	Total number of basefile requests served by NetScaler.
delcompbaseservedrate	<Double>	Read-only	Rate (/s) counter for delcompbaseserved
delcompbasetcptxbytes	<Double>	Read-only	Number of basefile bytes transmitted by NetScaler.
delcompbasetcptxbytesrate	<Double>	Read-only	Rate (/s) counter for delcompbasetcptxbytes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
delcomperrbypassedrate	<Double>	Read-only	Rate (/s) counter for delcomperrbypassed
delcomperrbfilewhdrfailed	<Double>	Read-only	Number of times basefile could not be updated in NetScaler cache.
delcomperrbfilewhdrfailedrate	<Double>	Read-only	Rate (/s) counter for delcomperrbfilewhdrfailed
delcomperrnostoremiss	<Double>	Read-only	Number of times basefile was not found in NetScaler cache.
delcomperrnostoremissrate	<Double>	Read-only	Rate (/s) counter for delcomperrnostoremiss
delcomperrreqinfotoobig	<Double>	Read-only	Number of times basefile request URL was too large.
delcomperrreqinfotoobigrate	<Double>	Read-only	Rate (/s) counter for delcomperrreqinfotoobig
delcomperrreqinfoallocfail	<Double>	Read-only	Number of times requested basefile could not be allocated.
delcomperrreqinfoallocfailrate	<Double>	Read-only	Rate (/s) counter for delcomperrreqinfoallocfail
delcomperrsessallocfail	<Double>	Read-only	Number of times delta compression session could not be allocated.
delcomperrsessallocfailrate	<Double>	Read-only	Rate (/s) counter for delcomperrsessallocfail

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/cmp

Query-parameters:

args

http://<netscaler-ip-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

address>/nitro/v1/stat/cmp?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get cmp resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmp": [ {
    "comphttpbandwidthsaving":<Double_value>,
    "decomptcptxpacketsrate":<Double_value>,
    "delcompbasetcptxbytesrate":<Double_value>,
    "comptotaltxbytes":<Double_value>,
    "comptcpeoirate":<Double_value>,
    "delcomperrbypassedrate":<Double_value>,
    "delcmpratio":<Double_value>,
    "delcomprequestsrate":<Double_value>,
    "delcomptcprxpacketsrate":<Double_value>,
    "decomptcperrmemory":<Double_value>,
    "comptotaldatacompressionratio":<Double_value>,
    "comprxbytesrate":<Double_value>,
    "delcomperrsessallocfailrate":<Double_value>,
    "delcomptcptxpacketsrate":<Double_value>,
    "decomptcperrmemoryrate":<Double_value>,
    "decomptcperrmoredata":<Double_value>,
    "delcompfirstaccessrate":<Double_value>,
    "comprxpacketsrate":<Double_value>,
    "decomptcprxpacketsrate":<Double_value>,
    "comptcpquantumrate":<Double_value>,
    "comptxbytesrate":<Double_value>,
    "delcompbaseservedrate":<Double_value>,
    "comptcptxbytesrate":<Double_value>,
    "delcomptcprxpackets":<Double_value>,
    "delcomperrbfilewhdrfailed":<Double_value>,
    "decomptcperrmoredatarate":<Double_value>,
    "delcompbaseserved":<Double_value>,
    "compratio":<Double_value>,
    "decomptcperrlessdata":<Double_value>,
    "comptcprxbytesrate":<Double_value>,
    "comptcprxpacketsrate":<Double_value>,
    "delcomptcptxbytesrate":<Double_value>,
    "decomptcperrdatarate":<Double_value>,
    "delcompdonerate":<Double_value>,
    "decomptcpratio":<Double_value>,
    "decomptcperrlessdatarate":<Double_value>,
    "comptcptotaltxbytes":<Double_value>,
    "delcomptotalrequests":<Double_value>,
    "delcomperrreqinfoallocfailrate":<Double_value>,
    "delcomperrnostoremiss":<Double_value>,
    "delcompbasetcptxbytes":<Double_value>,
    "comptcptotalpush":<Double_value>,
    "delcompfirstaccess":<Double_value>,
    "delcompdone":<Double_value>,
    "comptcptotalrxpackets":<Double_value>,
    "delcomperrbypassed":<Double_value>,
    "delbwsaving":<Double_value>,
    "comprequestsrate":<Double_value>,
    "delcomptcptxbytes":<Double_value>,
    "delcomperrreqinfoallocfail":<Double_value>,
    "delcomperrreqinfotoobig":<Double_value>,
    "decomptcprxbytes":<Double_value>,
    "comptcptxpacketsrate":<Double_value>,
    "comptcptotaleoi":<Double_value>,
    "comptcppushrate":<Double_value>,
    "decomptcperrunknownrate":<Double_value>,
    "comptcpbandwidthsaving":<Double_value>,
    "comptotalrxbytes":<Double_value>,
    "decomptcptxbytes":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "decomptcprxpackets":<Double_value>,
    "comptcptotaltimer":<Double_value>,
    "delcomperrnostoremissrate":<Double_value>,
    "decomptcpbandwidthsaving":<Double_value>,
    "delcomperrsessallocfail":<Double_value>,
    "decomptcptxbytesrate":<Double_value>,
    "comptxpacketsrate":<Double_value>,
    "comptotaltxpackets":<Double_value>,
    "delcomperrreqinfotoobigrate":<Double_value>,
    "decomptcprxbytesrate":<Double_value>,
    "comptotalrequests":<Double_value>,
    "decomptcperrunknown":<Double_value>,
    "comptotalrxpackets":<Double_value>,
    "delcomptcprxbytes":<Double_value>,
    "comptcptimerrate":<Double_value>,
    "comptcptotalquantum":<Double_value>,
    "comptcptotaltxpackets":<Double_value>,
    "delcomptcptxpackets":<Double_value>,
    "comptcptotalrxbytes":<Double_value>,
    "delcomptcprxbytesrate":<Double_value>,
    "decomptcptxpackets":<Double_value>,
    "delcomperrbfilewhdrfailedrate":<Double_value>,
    "comptcpratio":<Double_value>,
    "decomptcperrdata":<Double_value>
}]}
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicy

May 20, 2016

Statistics for compression policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the advanced compression policy for which to display statistics. If no name is specified, statistics for all advanced compression polices are shown. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/cmppolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/cmppolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_va**

Use this query-parameter to get cmppolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/cmppolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

cmppolicylabel

May 20, 2016

Statistics for compression policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the compression policy label for which to display statistics. If not specified, statistics are displayed for all compression policy labels. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/cmppolicylabel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/cmppolicylabel?**args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<**

Use this query-parameter to get cmppolicylabel resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicylabel": [ {
      "labelname":<String_value>,
      "pipolicylabelhitsrate":<Double_value>,
      "pipolicylabelhits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/cmppolicylabel/labelname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "cmppolicylabel": [ {
      "labelname":<String_value>,
      "pipolicylabelhitsrate":<Double_value>,
      "pipolicylabelhits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cache Redirection

Jul 07, 2017

Cache redirection statistics. The cache redirection feature can transparently redirect cacheable HTTP requests to a cache and send non-cacheable or dynamic HTTP requests to the origin server. A cache stores or caches frequently requested web content and serves such web content to a client on behalf of the origin servers, alleviating the load on the origin server farm.

Name	Description
crvserver	Statistics for CR virtual server resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

crvserver

May 20, 2016

Statistics for CR virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of a specific cache redirection virtual server. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
tothits	<Double>	Read-only	Total vserver hits
hitsrate	<Double>	Read-only	Rate (/s) counter for tothits
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
totalpktsrecvd	<Double>	Read-only	Total number of packets received by this service or virtual server.
pktsrecvdrate	<Double>	Read-only	Rate (/s) counter for totalpktsrecvd
totalpktssent	<Double>	Read-only	Total number of packets sent.
pktssentrate	<Double>	Read-only	Rate (/s) counter for totalpktssent
invalidrequestresponse	<Double>	Read-only	Number invalid requests/responses on this vserver
invalidrequestresponsedropped	<Double>	Read-only	Number invalid requests/responses dropped on this vserver

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/crvserver

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/crvserver?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_val**

Use this query-parameter to get crvserver resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver": [ {
    "name":<String_value>,
    "totalpktssent":<Double_value>,
    "tothits":<Double_value>,
    "totalrequests":<Double_value>,
    "responsebytesrate":<Double_value>,
    "invalidrequestresponsedropped":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "type":<String_value>,
    "hitsrate":<Double_value>,
    "pktsrecvdrate":<Double_value>,
    "primaryipaddress":<String_value>,
    "responsesrate":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "invalidrequestresponse":<Double_value>,
    "state":<String_value>,
    "totalpktsrecvd":<Double_value>,
    "pktssentrate":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/crvserver/name_value><String>

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "crvserver": [ {
    "name":<String_value>,
    "totalpktssent":<Double_value>,
    "tothits":<Double_value>,
    "totalrequests":<Double_value>,
    "responsebytesrate":<Double_value>,
    "invalidrequestresponsedropped":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "type":<String_value>,
    "hitsrate":<Double_value>,
    "pktsrecvdrate":<Double_value>,
    "primaryipaddress":<String_value>,
    "responsesrate":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "invalidrequestresponse":<Double_value>,
    "state":<String_value>,
    "totalpktsrecvd":<Double_value>,
    "pktssentrate":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Content Switching

May 20, 2016

Content Switching statistics. Content Switching feature that enables you to direct traffic to servers on the basis of content.

Name	Description
csvserver	Statistics for CS virtual server resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

csvserver

May 20, 2016

Statistics for CS virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the content switching virtual server for which to display statistics. To display statistics for all configured Content Switching virtual servers, do not specify a value for this parameter. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
establishedconn	<Double>	Read-only	Number of client connections in ESTABLISHED state.
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
tothits	<Double>	Read-only	Total vserver hits
hitsrate	<Double>	Read-only	Rate (/s) counter for tothits
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
totalpktsrecvd	<Double>	Read-only	Total number of packets received by this service or virtual server.
pktsrecvdrate	<Double>	Read-only	Rate (/s) counter for totalpktsrecvd
totalpktssent	<Double>	Read-only	Total number of packets sent.
pktssentrate	<Double>	Read-only	Rate (/s) counter for totalpktssent
curclntconnections	<Double>	Read-only	Number of current client connections.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
sothreshold	<Double>	Read-only	Spill Over Threshold set on the VServer.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
labeledconn	<Double>	Read-only	Number of labeled connection on this vserver
pushlabel	<Double>	Read-only	Number of labels for this push vserver.
deferredreq	<Double>	Read-only	Number of deferred request on this vserver
deferredreqrate	<Double>	Read-only	Rate (/s) counter for deferredreq
invalidrequestresponse	<Double>	Read-only	Number invalid requests/responses on this vserver
invalidrequestresponsedropped	<Double>	Read-only	Number invalid requests/responses dropped on this vserver
totvserverdownbackuphits	<Double>	Read-only	Number of times traffic was diverted to backup vserver since primary vserver was DOWN.
curmptcpsessions	<Double>	Read-only	Current Multipath TCP sessions
cursubflowconn	<Double>	Read-only	Current Multipath TCP subflows

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/csvserver

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/csvserver?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_val**

Use this query-parameter to get csvserver resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver": [ {
    "name":<String_value>,
    "curclntconnections":<Double_value>,
    "establishedconn":<Double_value>,
    "totalpktssent":<Double_value>,
    "labelledconn":<Double_value>,
    "tothits":<Double_value>,
    "totalrequests":<Double_value>,
    "sothreshold":<Double_value>,
    "cursubflowconn":<Double_value>,
    "responsebytesrate":<Double_value>,
    "invalidrequestresponsedropped":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "type":<String_value>,
    "hitsrate":<Double_value>,
    "cursrvrconnections":<Double_value>,
    "pktsrecvdrate":<Double_value>,
    "primaryipaddress":<String_value>,
    "pushlabel":<Double_value>,
    "responsesrate":<Double_value>,
    "deferredreq":<Double_value>,
    "curmptcpsessions":<Double_value>,
    "totspillovers":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "invalidrequestresponse":<Double_value>,
    "state":<String_value>,
    "deferredreqrate":<Double_value>,
    "totalpktsrecvd":<Double_value>,
    "pktssentrate":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>,
    "totvserverdownbackuphits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/csvserver/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "csvserver": [ {
    "name":<String_value>,
    "curclntconnections":<Double_value>,
    "establishedconn":<Double_value>,
    "totalpktssent":<Double_value>,
    "labelledconn":<Double_value>,
    "tothits":<Double_value>,
    "totalrequests":<Double_value>,
    "sothreshold":<Double_value>,
    "cursubflowconn":<Double_value>,
    "responsebytesrate":<Double_value>,
    "invalidrequestresponsedropped":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "type":<String_value>,
    "hitsrate":<Double_value>,
    "cursrvrconnections":<Double_value>,
    "pktsrecvdrate":<Double_value>,
    "primaryipaddress":<String_value>,
    "pushlabel":<Double_value>,
    "responsesrate":<Double_value>,
    "deferredreq":<Double_value>,
    "curmptcpsessions":<Double_value>,
    "totspillovers":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "invalidrequestresponse":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "state":<String_value>,
    "deferredreqrate":<Double_value>,
    "totalpktsrecvd":<Double_value>,
    "pktssentrate":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>,
    "totvserverdownbackuphits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Domain Name Service

May 20, 2016

Domain Name Service(DNS) statistics.

Name	Description
dns	Statistics for dns.
dnspolicylabel	Statistics for dns policy label resource.
dnsrecords	

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dns

May 20, 2016
Statistics for dns.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
dnstotqueries	<Double>	Read-only	Total number of DNS queries received.
dnsqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotqueries
dnstotmultiquery	<Double>	Read-only	Total number of Multi Query request received.
dnstotanswers	<Double>	Read-only	Total number of DNS responses received.
dnsanswersrate	<Double>	Read-only	Rate (/s) counter for dnstotanswers
dnstotserverresponse	<Double>	Read-only	Total number of Server responses received.
dnsserverresponserate	<Double>	Read-only	Rate (/s) counter for dnstotserverresponse
dnstotrecupdate	<Double>	Read-only	Total number of record updates.
dnstotauthans	<Double>	Read-only	Number of queries which were authoritatively answered.
dnstotserverquery	<Double>	Read-only	Total number of Server queries sent.
dnsserverqueryrate	<Double>	Read-only	Rate (/s) counter for dnstotserverquery
dnstotcacheflush	<Double>	Read-only	Total number of times cache was flushed.
dnstotcacheentriesflush	<Double>	Read-only	Total number of cache entries flushed.
dnscurnoauthentries	<Double>	Read-only	Total number of non-authoritative entries.
dnscurauthentries	<Double>	Read-only	Total number of authoritative entries.
dnstotauthnonames	<Double>	Read-only	Number of queries for which no record was found.
dnstotunsupportedresponseclass	<Double>	Read-only	Total number of responses for which response types were unsupported.
dnstotinvalidqueryformat	<Double>	Read-only	Total number of queries whose format was invalid.
dnstotstrayanswer	<Double>	Read-only	Total number of stray answers.
dnstotresponsebadlen	<Double>	Read-only	Number of DNS responses received with invalid resoure data length.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
dnsfrnullattack	<Double>	Read-only	Total number of queries received where all the counts are 0.
dnstotunsupportedresponsetype	<Double>	Read-only	Total number of responses for which response type requested was unsupported.
dnstotunsupportedqueryclass	<Double>	Read-only	Total number of queries for which query class was unsupported.
dnstotnonauthnodatas	<Double>	Read-only	Total number of responses for which there was a format error.
dnstotnodataresps	<Double>	Read-only	Number of DNS responses received without answer.
dnstotmultiquerydisableerror	<Double>	Read-only	Total number of times a multi query was disabled and received a multi query.
dnstotothererrors	<Double>	Read-only	Total number of other errors.
dns64totqueries	<Double>	Read-only	Total number of DNS64 queries recieved.
dns64queriesrate	<Double>	Read-only	Rate (/s) counter for dns64totqueries
dns64totanswers	<Double>	Read-only	Total number of DNS64 answers served.
dns64answersrate	<Double>	Read-only	Rate (/s) counter for dns64totanswers
dns64totrwanswers	<Double>	Read-only	Total number of DNS64 answers served after rewriting the response.
dns64rwanswersrate	<Double>	Read-only	Rate (/s) counter for dns64totrwanswers
dns64totresponses	<Double>	Read-only	Total number of responses recieved from backend in DNS64 context.
dns64responsesrate	<Double>	Read-only	Rate (/s) counter for dns64totresponses
dns64totgslbqueries	<Double>	Read-only	Total number of DNS64 queries for GSLB domain
dns64gslbqueriesrate	<Double>	Read-only	Rate (/s) counter for dns64totgslbqueries
dns64totgslbanswers	<Double>	Read-only	Total number of DNS64 queries served.
dns64gslbanswersrate	<Double>	Read-only	Rate (/s) counter for dns64totgslbanswers
dns64tottcanswers	<Double>	Read-only	Total number of Answers served with TC bit set in DNS64 context.
dns64tcanswersrate	<Double>	Read-only	Rate (/s) counter for dns64tottcanswers
dns64totsvraqueries	<Double>	Read-only	Total number of Queries sent by DNS64 module to backend.
dns64svraqueriesrate	<Double>	Read-only	Rate (/s) counter for dns64totsvraqueries
dns64totaaaabypass	<Double>	Read-only	Total number of times AAAA query has been bypassed in DNS64 trnsaction.
dns64aaaabypassrate	<Double>	Read-only	Rate (/s) counter for dns64totaaaabypass
dns64tottcpqueries	<Double>	Read-only	Total number of dns64 queries over TCP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
dns64activepolicies	<Double>	Read-only	Total number of active dns64 policies
dns64totnodataresp	<Double>	Read-only	Total number of responses recieved from backend with ancount 0
dns64nodataresprate	<Double>	Read-only	Rate (/s) counter for dns64totnodataresp
dnstotnsrecqueries	<Double>	Read-only	Total number of NS queries received.
dnsnsrecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotnsrecqueries
dnstotsoarecqueries	<Double>	Read-only	Total number of SOA queries received.
dnssoarecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotsoarecqueries
dnstotptrrecqueries	<Double>	Read-only	Total number of PTR queries received.
dnsptrrecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotptrrecqueries
dnstotsrvrecqueries	<Double>	Read-only	Total number of SRV queries received.
dnssrvrecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotsrvrecqueries
dnstotaresponse	<Double>	Read-only	Total number of A responses received.
dnsaresponserate	<Double>	Read-only	Rate (/s) counter for dnstotaresponse
dnstotcnameresponse	<Double>	Read-only	Total number of CNAME responses received.
dnscnameresponserate	<Double>	Read-only	Rate (/s) counter for dnstotcnameresponse
dnstotmxresponse	<Double>	Read-only	Total number of MX responses received.
dnsmxresponserate	<Double>	Read-only	Rate (/s) counter for dnstotmxresponse
dnstotanyresponse	<Double>	Read-only	Total number of ANY responses received.
dnsanyresponserate	<Double>	Read-only	Rate (/s) counter for dnstotanyresponse
dnstotnsrecupdate	<Double>	Read-only	Total number of NS record updates.
dnstotsoarecupdate	<Double>	Read-only	Total number of SOA record updates.
dnstotptrrecupdate	<Double>	Read-only	Total number of PTR record updates.
dnstotsrvrecupdate	<Double>	Read-only	Total number of SRV record updates.
dnstotaaaarecqueries	<Double>	Read-only	Total number of AAAA queries received.
dnsaaaarecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotaaaarecqueries

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
dnsarecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotarecqueries
dnstotcnamerecqueries	<Double>	Read-only	Total number of CNAME queries received.
dnscnamerecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotcnamerecqueries
dnstotmxrecqueries	<Double>	Read-only	Total number of MX queries received.
dnsmxrecqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotmxrecqueries
dnstotanyqueries	<Double>	Read-only	Total number of ANY queries received.
dnsanyqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotanyqueries
dnstotaaaaresponse	<Double>	Read-only	Total number of AAAA responses received.
dnsaaaaresponserate	<Double>	Read-only	Rate (/s) counter for dnstotaaaaresponse
dnstotnsresponse	<Double>	Read-only	Total number of NS responses received.
dnsnsresponserate	<Double>	Read-only	Rate (/s) counter for dnstotnsresponse
dnstotsoaresponse	<Double>	Read-only	Total number of SOA responses received.
dnssoaresponserate	<Double>	Read-only	Rate (/s) counter for dnstotsoaresponse
dnstotptrresponse	<Double>	Read-only	Total number of PTR responses received.
dnsptrresponserate	<Double>	Read-only	Rate (/s) counter for dnstotptrresponse
dnstotsrvresponse	<Double>	Read-only	Total number of SRV responses received.
dnssrvresponserate	<Double>	Read-only	Rate (/s) counter for dnstotsrvresponse
dnstotaaaarecupdate	<Double>	Read-only	Total number of AAAA record updates.
dnstotarecupdate	<Double>	Read-only	Total number of A record updates.
dnstotmxrecupdate	<Double>	Read-only	Total number of MX record updates.
dnstotcnamerecupdate	<Double>	Read-only	Total number of CNAME record updates.
dnscuraaaarecord	<Double>	Read-only	Total number of AAAA records.
dnscurarecord	<Double>	Read-only	Total number of A records.
dnscurmxrecord	<Double>	Read-only	Total number of MX records.
dnscurcnamerecord	<Double>	Read-only	Total number of CNAME records.
dnscurnsrecord	<Double>	Read-only	Total number of NS records.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
dnscursoarecord	<Double>	Read-only	Total number of SOA records.
dnscurptrrecord	<Double>	Read-only	Total number of PTR records.
dnscursrvrecord	<Double>	Read-only	Total number of SRV records.
dnstotaaaarecfailed	<Double>	Read-only	Total number of times AAAA record lookup failed.
dnstotarecfailed	<Double>	Read-only	Total number of times A record lookup failed.
dnstotmxrecfailed	<Double>	Read-only	Total number of times MX record lookup failed.
dnstotptrrecfailed	<Double>	Read-only	Total number of times PTR record lookup failed.
dnstotnsrecfailed	<Double>	Read-only	Total number of times NS record lookup failed.
dnstotcnamerecfailed	<Double>	Read-only	Total number of times CNAME record lookup failed.
dnstotsoarecfailed	<Double>	Read-only	Total number of times SOA record lookup failed.
dnstotsrvrecfailed	<Double>	Read-only	Total number of times SRV record lookup failed.
dnstotanyrecfailed	<Double>	Read-only	Total number of times ANY query lookup failed.
dnstotunsupportedqueries	<Double>	Read-only	Total number of requests for which query type requested was unsupported.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/dns

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/dns?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get dns resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dns": [ {
    "dnstotcnameresponse":<Double_value>,
    "dnsaresponserate":<Double_value>,
    "dns64rwanswersrate":<Double_value>,
    "dnstotaaaarecupdate":<Double_value>,
    "dnstotptrrecupdate":<Double_value>,
    "dns64nodatarespratе":<Double_value>,
    "dnserrnullattack":<Double_value>,
    "dnstotunsupportedqueries":<Double_value>,
    "dnstotauthnonames":<Double_value>,
    "dnstotmxrecupdate":<Double_value>,
    "dnstotanyqueries":<Double_value>,
    "dns64totrwanswers":<Double_value>,
    "dnstotstrayanswer":<Double_value>,
    "dnstotptrrecfailed":<Double_value>,
    "dns64activepolicies":<Double_value>,
    "dnstotcnamerecupdate":<Double_value>,
    "dnscursoarecord":<Double_value>,
    "dnstotcacheentriesflush":<Double_value>,
    "dnsaaaaresponserate":<Double_value>,
    "dnsaaaarecqueriesrate":<Double_value>,
    "dnstotmxrecqueries":<Double_value>,
    "dnstotsoarecqueries":<Double_value>,
    "dnscurarecord":<Double_value>,
    "dnstotinvalidqueryformat":<Double_value>,
    "dnstotsrvrecfailed":<Double_value>,
    "dnsarecqueriesrate":<Double_value>,
    "dnstotsoaresponse":<Double_value>,
    "dnstotserverresponse":<Double_value>,
    "dnsanyqueriesrate":<Double_value>,
    "dnscurauthentries":<Double_value>,
    "dnstotmxresponse":<Double_value>,
    "dnstotptrrecqueries":<Double_value>,
    "dnstotunsupportedqueryclass":<Double_value>,
    "dnsptrresponserate":<Double_value>,
    "dnssoarecqueriesrate":<Double_value>,
    "dns64totgslbanswers":<Double_value>,
    "dns64answersrate":<Double_value>,
    "dns64totqueries":<Double_value>,
    "dnsptrrecqueriesrate":<Double_value>,
    "dnstotsoarecfailed":<Double_value>,
    "dns64tottcpqueries":<Double_value>,
    "dnstotaaaarecqueries":<Double_value>,
    "dns64responsesrate":<Double_value>,
    "dnstotmxrecfailed":<Double_value>,
    "dns64tottcanswers":<Double_value>,
    "dnstotaaaarecfailed":<Double_value>,
    "dnssrvresponserate":<Double_value>,
    "dnsnsrecqueriesrate":<Double_value>,
    "dnstotserverquery":<Double_value>,
    "dnssoaresponserate":<Double_value>,
    "dnstotmultiquery":<Double_value>,
    "dnscuraaaarecord":<Double_value>,
    "dnsqueriesrate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "dns64gslbqueriesrate":<Double_value>,
    "dnsanyresponserate":<Double_value>,
    "dnsanswersrate":<Double_value>,
    "dnstotarecupdate":<Double_value>,
    "dnscnameresponserate":<Double_value>,
    "dnstotothererrors":<Double_value>,
    "dnstotnsrecfailed":<Double_value>,
    "dnscurcnamerecord":<Double_value>,
    "dnscurnoauthentries":<Double_value>,
    "dnstotresponsebadlen":<Double_value>,
    "dns64totaaaabypass":<Double_value>,
    "dns64tcpqueriesrate":<Double_value>,
    "dnstotaaaaresponse":<Double_value>,
    "dns64gslbanswersrate":<Double_value>,
    "dnstotunsupportedresponsetype":<Double_value>,
    "dns64totsvraqueries":<Double_value>,
    "dns64totresponses":<Double_value>,
    "dnstotnsrecqueries":<Double_value>,
    "dns64totanswers":<Double_value>,
    "dnscursrvrecord":<Double_value>,
    "dnscurptrrecord":<Double_value>,
    "dnstotanyresponse":<Double_value>,
    "dnstotanyrecfailed":<Double_value>,
    "dns64aaaabypassrate":<Double_value>,
    "dnstotnsresponse":<Double_value>,
    "dnssrvrecqueriesrate":<Double_value>,
    "dnstotnsrecupdate":<Double_value>,
    "dnstotcnamerecqueries":<Double_value>,
    "dnstotmultiquerydisableerror":<Double_value>,
    "dnstotarecqueries":<Double_value>,
    "dnsserverresponserate":<Double_value>,
    "dnsnsresponserate":<Double_value>,
    "dnstotanswers":<Double_value>,
    "dnsmxrecqueriesrate":<Double_value>,
    "dnstotcnamerecfailed":<Double_value>,
    "dnstotsrvrecqueries":<Double_value>,
    "dnstotaresponse":<Double_value>,
    "dnscnamerecqueriesrate":<Double_value>,
    "dns64totnodataresp":<Double_value>,
    "dnstotqueries":<Double_value>,
    "dnstotsrvresponse":<Double_value>,
    "dnstotunsupportedresponseclass":<Double_value>,
    "dnstotcacheflush":<Double_value>,
    "dnstotarecfailed":<Double_value>,
    "dnstotsrvrecupdate":<Double_value>,
    "dns64svraqueriesrate":<Double_value>,
    "dnsserverqueryrate":<Double_value>,
    "dnscurnsrecord":<Double_value>,
    "dnstotsoarecupdate":<Double_value>,
    "dns64queriesrate":<Double_value>,
    "dnstotnonauthnodatas":<Double_value>,
    "dnstotauthans":<Double_value>,
    "dnstotreqrefusals":<Double_value>,
    "dnscurmxrecord":<Double_value>,
    "dns64tcanswersrate":<Double_value>,
    "dnstotrecupdate":<Double_value>,
    "dns64totgslbqueries":<Double_value>,
    "dnstotnodataresps":<Double_value>,
    "dnstotptrresponse":<Double_value>,
    "dnsmxresponserate":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnspolicylabel

Jul 07, 2017
Statistics for dns policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	The name of the dns policy label for which statistics will be displayed. If not given statistics are shown for all dns policylabels. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/dnspolicylabel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/dnspolicylabel?**args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<S**

Use this query-parameter to get dnspolicylabel resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicylabel": [ {
    "labelname":<String_value>,
    "pipolicylabelhitsrate":<Double_value>,
    "pipolicylabelhits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/dnspolicylabel/labelname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnspolicylabel": [ {
    "labelname":<String_value>,
    "pipolicylabelhitsrate":<Double_value>,
    "pipolicylabelhits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dnsrecords

Jul 07, 2017
Read/write properties

Properties (click to see Operations)

Name	Data Type	Permissions	Description
dnsrecordtype	<String>	Read-write	Display statistics for the specified DNS record or query type or, if a record or query type is not specified, statistics for all record types supported on the NetScaler appliance.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
dnstotentries	<Double>	Read-only	Total number of DNS record entries
dnstotupdates	<Double>	Read-only	Total number of DNS proactive updates
dnstotresponses	<Double>	Read-only	Total number of DNS server responses
dnstotrequests	<Double>	Read-only	Total number of DNS queries recieved
dnscurentries	<Double>	Read-only	Current number of DNS entries
dnstoterrlimits	<Double>	Read-only	Total number of times we have recieved dns record with more entries than we support
dnstoterrrespform	<Double>	Read-only	Total number of times we have recieved malformed responses from the backend
dnstoterraliasex	<Double>	Read-only	Total number of times we have recieved non-cname record for a domain for which an alias exists
dnstoterrnodomains	<Double>	Read-only	Total number of cache misses
dnscurrecords	<Double>	Read-only	Current number of DNS Records

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/dnsrecords

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/dnsrecords?**args=dnsrecordtype:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:**

Use this query-parameter to get dnsrecords resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsrecords": [ {
    "dnsrecordtype":<String_value>,
    "dnscurentries":<Double_value>,
    "dnstotupdates":<Double_value>,
    "dnstotrequests":<Double_value>,
    "dnstoterraliasex":<Double_value>,
    "dnstoterrrespform":<Double_value>,
    "dnstotentries":<Double_value>,
    "dnscurrecords":<Double_value>,
    "dnstoterrnodomains":<Double_value>,
    "dnstoterrlimits":<Double_value>,
    "dnstotresponses":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/dnsrecords/dnsrecordtype_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dnsrecords": [ {
    "dnsrecordtype":<String_value>,
    "dnscurentries":<Double_value>,
    "dnstotupdates":<Double_value>,
    "dnstotrequests":<Double_value>,
    "dnstoterraliasex":<Double_value>,
    "dnstoterrrespform":<Double_value>,
    "dnstotentries":<Double_value>,
    "dnscurrecords":<Double_value>,
    "dnstoterrnodomains":<Double_value>,
    "dnstoterrlimits":<Double_value>,
    "dnstotresponses":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP DoS Protection

May 20, 2016

HTTP DOS protection statistics.

Name	Description
dos	Statistics for dos.
dospolicy	Statistics for DoS policy resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

dos

Jul 07, 2017

Statistics for dos.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
dostotconditiontriggered	<Double>	Read-only	Number of times the NetScaler appliance triggered the DOS JavaScript due to a condition match.
dosconditiontriggeredrate	<Double>	Read-only	Rate (/s) counter for dostotconditiontriggered
dostotvalidcookies	<Double>	Read-only	Number of clients from whom the NetScaler appliance received a valid DOS cookie.
dosvalidcookiesrate	<Double>	Read-only	Rate (/s) counter for dostotvalidcookies
dostotdosprioritycliens	<Double>	Read-only	Number of valid clients that were given DOS priority.
dosdosprioritycliensrate	<Double>	Read-only	Rate (/s) counter for dostotdosprioritycliens

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/dos

Query-parameters:

args

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/stat/dos?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get dos resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dos": [ {
    "dostotvalidcookies":<Double_value>,
    "dosvalidcookiesrate":<Double_value>,
    "dosdospriorityclientsrate":<Double_value>,
    "dosconditiontriggeredrate":<Double_value>,
    "dostotconditiontriggered":<Double_value>,
    "dostotdospriorityclients":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

dospolicy

May 20, 2016

Statistics for DoS policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the DoS protection policy whose statistics must be displayed. If a name is not provided, statistics of all the DoS protection policies are displayed. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
doscurrentcltdetectrate	<Double>	Read-only	Current ratio of JavaScript send rate to the server response rate (Client detect rate)
dosphysicalserviceip	<String>	Read-only	IP address of the service to which this policy is bound.
dosphysicalserviceport	<Integer>	Read-only	Port address of the service to which this policy is bound.
doscurrentqueuesize	<Double>	Read-only	Current queue size of the server to which this policy is bound.
doscurrentqueuesizerate	<Double>	Read-only	Rate (/s) counter for doscurrentqueuesize
dostotjssent	<Double>	Read-only	Total number of DoS JavaScript transactions performed for this policy.
dosjssentrate	<Double>	Read-only	Rate (/s) counter for dostotjssent
dostotjsrefused	<Double>	Read-only	Number of times the DoS JavaScript was not sent because the set JavaScript rate was not met for this policy.
dosjsrefusedrate	<Double>	Read-only	Rate (/s) counter for dostotjsrefused
dostotvalidclients	<Double>	Read-only	Total number of valid DoS cookies received for this policy.
dosvalidclientsrate	<Double>	Read-only	Rate (/s) counter for dostotvalidclients
dostotjsbytessent	<Double>	Read-only	Total number of DoS JavaScript bytes sent for this policy.
dosjsbytessentrate	<Double>	Read-only	Rate (/s) counter for dostotjsbytessent
dostotnongetpostrequests	<Double>	Read-only	Number of non-GET and non-POST requests for which DOS JavaScript was sent.
dosnongetpostrequestsrate	<Double>	Read-only	Rate (/s) counter for dostotnongetpostrequests

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/dospolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/dospolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_va**

Use this query-parameter to get dospolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dospolicy": [ {
    "name":<String_value>,
    "dosjsrefusedrate":<Double_value>,
    "dosphysicalserviceip":<String_value>,
    "dosnongetpostrequestsrate":<Double_value>,
    "dosjssentrate":<Double_value>,
    "dosphysicalserviceport":<Integer_value>,
    "dosjsbytessentrate":<Double_value>,
    "doscurrentcltdetectrate":<Double_value>,
    "dostotnongetpostrequests":<Double_value>,
    "dostotvalidclients":<Double_value>,
    "dosvalidclientsrate":<Double_value>,
    "doscurrentqueuesizerate":<Double_value>,
    "dostotjsbytessent":<Double_value>,
    "dostotjssent":<Double_value>,
    "dostotjsrefused":<Double_value>,
    "doscurrentqueuesize":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/dospolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "dospolicy": [ {
    "name":<String_value>,
    "dosjsrefusedrate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "dosphysicalserviceip":<String_value>,
    "dosnongetpostrequestsrate":<Double_value>,
    "dosjssentrate":<Double_value>,
    "dosphysicalserviceport":<Integer_value>,
    "dosjsbytessentrate":<Double_value>,
    "doscurrentcltdetectrate":<Double_value>,
    "dostotnongetpostrequests":<Double_value>,
    "dostotvalidclients":<Double_value>,
    "dosvalidclientsrate":<Double_value>,
    "doscurrentqueuesizerate":<Double_value>,
    "dostotjsbytessent":<Double_value>,
    "dostotjssent":<Double_value>,
    "dostotjsrefused":<Double_value>,
    "doscurrentqueuesize":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Front-end-optimization

May 20, 2016

Front end optimization statistics. The system's feature to optimize Web content for performance.

Name	Description
feo	Statistics for feo.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

feo

May 20, 2016

Statistics for feo.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
optcacheobjects	<Double>	Read-only	Total number of optimized cache objects ready to be served.
optcacheobjectsrate	<Double>	Read-only	Rate (/s) counter for optcacheobjects
origcacheobjects	<Double>	Read-only	Total number of original cache objects ready to be served.
origcacheobjectsrate	<Double>	Read-only	Rate (/s) counter for origcacheobjects
totalimgsdomainsharded	<Double>	Read-only	Total no of images whose domain has been set from shards.
imgsdomainshardedrate	<Double>	Read-only	Rate (/s) counter for totalimgsdomainsharded
totalimgslazyloaded	<Double>	Read-only	Total number of images modified for lazy loading.
imgslazyloadedrate	<Double>	Read-only	Rate (/s) counter for totalimgslazyloaded
toturireplaced	<Double>	Read-only	Total number of URI replaced.
urireplacedrate	<Double>	Read-only	Rate (/s) counter for toturireplaced
totalimgsinlinedincss	<Double>	Read-only	Total number of images inlined in CSS.
imgsinlinedincssrate	<Double>	Read-only	Rate (/s) counter for totalimgsinlinedincss
totalinlinedjs	<Double>	Read-only	Total number of inlined JS files.
inlinedjsrate	<Double>	Read-only	Rate (/s) counter for totalinlinedjs
totalinlinedcss	<Double>	Read-only	Total number of inlined CSS files.
inlinedcssrate	<Double>	Read-only	Rate (/s) counter for totalinlinedcss
totalinlinedimgs	<Double>	Read-only	Total number of inlined images in HTML.
inlinedimgsrate	<Double>	Read-only	Rate (/s) counter for totalinlinedimgs
totaldatasavings	<Double>	Read-only	Total data savings in bytes.
datasavingsrate	<Double>	Read-only	Rate (/s) counter for totaldatasavings
htmlcommentsremoved	<Double>	Read-only	The total number of HTML comments removed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
htmlcommentsremovedrate	<Double>	Read-only	Rate (/s) counter for htmlcommentsremoved
totalcacheextended	<Double>	Read-only	The total number of objects cache extended.
cacheextendedrate	<Double>	Read-only	Rate (/s) counter for totalcacheextended
totalcsscombined	<Double>	Read-only	The total number of CSS combined.
csscombinedrate	<Double>	Read-only	Rate (/s) counter for totalcsscombined
totalimporttolink	<Double>	Read-only	The total number of CSS imports converted to links
importtolinkrate	<Double>	Read-only	Rate (/s) counter for totalimporttolink
totaljsmoved	<Double>	Read-only	The total number of JS moved to end.
jsmovedrate	<Double>	Read-only	Rate (/s) counter for totaljsmoved
totalcssmoved	<Double>	Read-only	The total number of CSS moved to head.
cssmovedrate	<Double>	Read-only	Rate (/s) counter for totalcssmoved
totaljsmin	<Double>	Read-only	The total number of JS files minified.
jsminrate	<Double>	Read-only	Rate (/s) counter for totaljsmin
totalcssmin	<Double>	Read-only	The total number of CSS files minified.
cssminrate	<Double>	Read-only	Rate (/s) counter for totalcssmin
totalimgstojxr	<Double>	Read-only	The total number of images converted to JPEGXR format.
imgstojxrrate	<Double>	Read-only	Rate (/s) counter for totalimgstojxr
totalimgstowebp	<Double>	Read-only	The total number of images converted to WEBP format.
imgstowebprate	<Double>	Read-only	Rate (/s) counter for totalimgstowebp
totaljpegsoptimized	<Double>	Read-only	The total number of JPEG format images optimized.
jpegsoptimizedrate	<Double>	Read-only	Rate (/s) counter for totaljpegsoptimized
totalgifstopng	<Double>	Read-only	The total number of images converted from GIF to PNG format.
gifstopngrate	<Double>	Read-only	Rate (/s) counter for totalgifstopng
totalimgsresized	<Double>	Read-only	The total number of images resized to dimensions in the <img> tag.
imgsresizedrate	<Double>	Read-only	Rate (/s) counter for totalimgsresized

Operations (click to see Properties)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/feo

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/feo?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get feo resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "feo": [ {
    "totaljsmoved":<Double_value>,
    "totalimgslazyloaded":<Double_value>,
    "htmlcommentsremovedrate":<Double_value>,
    "urireplacedrate":<Double_value>,
    "cssmovedrate":<Double_value>,
    "htmlcommentsremoved":<Double_value>,
    "jsmovedrate":<Double_value>,
    "totalimgsinlinedincss":<Double_value>,
    "totalinlinedcss":<Double_value>,
    "inlinedimgsrate":<Double_value>,
    "toturireplaced":<Double_value>,
    "imgstowebprate":<Double_value>,
    "totalinlinedjs":<Double_value>,
    "totalcssmoved":<Double_value>,
    "csscombinedrate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "imgslazyloadedrate":<Double_value>,
    "totaljsmin":<Double_value>,
    "optcacheobjects":<Double_value>,
    "totalgifstopng":<Double_value>,
    "totalimgsresized":<Double_value>,
    "imgsinlinedincssrate":<Double_value>,
    "inlinedcssrate":<Double_value>,
    "totalimgstowebp":<Double_value>,
    "inlinedjsrate":<Double_value>,
    "origcacheobjectsrate":<Double_value>,
    "cssminrate":<Double_value>,
    "totalimgsdomainsharded":<Double_value>,
    "imgsdomainshardedrate":<Double_value>,
    "gifstopngrate":<Double_value>,
    "importtolinkrate":<Double_value>,
    "totalcsscombined":<Double_value>,
    "totalimgstojxr":<Double_value>,
    "imgsresizedrate":<Double_value>,
    "optcacheobjectsrate":<Double_value>,
    "totaljpegsoptimized":<Double_value>,
    "totalcacheextended":<Double_value>,
    "origcacheobjects":<Double_value>,
    "jsminrate":<Double_value>,
    "jpegsoptimizedrate":<Double_value>,
    "totaldatasavings":<Double_value>,
    "imgstojxrrate":<Double_value>,
    "datasavingsrate":<Double_value>,
    "cacheextendedrate":<Double_value>,
    "totalcssmin":<Double_value>,
    "totalinlinedimgs":<Double_value>,
    "totalimporttolink":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Global Server Load Balancing

May 20, 2016

Global Server Load Balancing (GSLB) statistics. GSLB feature ensures that client requests are directed to a best performing site available in a global enterprise and distributed Internet environment.

Name	Description
gslbdomain	Statistics for GSLB domain resource.
gslbservice	Statistics for GSLB service resource.
gslbsite	Statistics for GSLB site resource.
gslbvserver	Statistics for Global Server Load Balancing Virtual Server resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbdomain

May 20, 2016

Statistics for GSLB domain resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the GSLB domain for which to display statistics. If you do not specify a name, statistics are shown for all configured GSLB domains. Minimum length = 1
dnsrecordtype	<String>	Read-write	. Possible values = A, AAAA, CNAME, NAPTR
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
dnstotalqueries	<Double>	Read-only	Total number of DNS queries received.
dnsqueriesrate	<Double>	Read-only	Rate (/s) counter for dnstotalqueries
gslbdnsrectype	<String>	Read-only	Type of DNS record returned

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/gslbdomain

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/gslbdomain?**args=name:<String_value>,dnsrecordtype:<String_value>,dnsrecordtype:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntim**

Use this query-parameter to get gslbdomain resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbdomain": [ {
    "name":<String_value>,
    "dnsqueriesrate":<Double_value>,
    "dnstotalqueries":<Double_value>,
    "gslbdnsrectype":<String_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/gslbdomain/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbdomain": [ {
    "name":<String_value>,
    "dnsqueriesrate":<Double_value>,
    "dnstotalqueries":<Double_value>,
    "gslbdnsrectype":<String_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbservice

May 20, 2016

Statistics for GSLB service resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
servicename	<String>	Read-write	Name of the GSLB service. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
establishedconn	<Double>	Read-only	Number of client connections in ESTABLISHED state.
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
servicetype	<String>	Read-only	The service type of this service.Possible values are ADNS, DNS, MYSQL, RTSP, SSL_DIAMETER, ADNS_TCP, DNS_TCP, NNTP, SIP_UDP, SSL_TCP, ANY, FTP, RADIUS, SNMP, TCP, DHCPRA, HTTP, RDP, SSL, TFTP, DIAMETER, MSSQL, RPCSVR, SSL_BRIDGE, UDP
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
curload	<Double>	Read-only	Load on the service that is calculated from the bound load based monitor.
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
curclntconnections	<Double>	Read-only	Number of current client connections.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
vsvrservicehits	<Double>	Read-only	Number of times that the service has been provided.
vsvrservicehitsrate	<Double>	Read-only	Rate (/s) counter for vsvrservicehits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/gslbservice

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/gslbservice?**args=servicename:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<St**

Use this query-parameter to get gslbservice resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbservice": [ {
    "servicename":<String_value>,
    "curclntconnections":<Double_value>,
    "servicetype":<String_value>,
    "establishedconn":<Double_value>,
    "totalrequests":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "cursrvrconnections":<Double_value>,
    "primaryipaddress":<String_value>,
    "responsesrate":<Double_value>,
    "curload":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "state":<String_value>,
    "vsvrservicehits":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>,
    "vsvrservicehitsrate":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/gslbservice/servicename_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbservice": [ {
    "servicename":<String_value>,
    "curclntconnections":<Double_value>,
    "servicetype":<String_value>,
    "establishedconn":<Double_value>,
    "totalrequests":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "cursrvrconnections":<Double_value>,
    "primaryipaddress":<String_value>,
    "responsesrate":<Double_value>,
    "curload":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "state":<String_value>,
    "vsvrservicehits":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>,
    "vsvrservicehitsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbsite

May 20, 2016

Statistics for GSLB site resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
sitename	<String>	Read-write	Name of the GSLB site for which to display detailed statistics. If a name is not specified, basic information about all GSLB sites is displayed. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sitepublicip	<String>	Read-only	The public IP address of this GSLB site.
siteip	<String>	Read-only	The private IP address of this GSLB site.
sitemepstatus	<String>	Read-only	Indicates the status of the Metric Exchange Policy at this GSLB site.
persexchange	<String>	Read-only	Indicates whether Persistence entries exchange is enabled or disabled at this GSLB site.
nwmetricexchange	<String>	Read-only	Indicates whether network metric exchange is enabled or disabled at this GSLB site.
sitemetricexchange	<String>	Read-only	Indicates whether metric exchange is enabled or disabled at this GSLB site.
sitetype	<String>	Read-only	Indicates whether this GSLB site is local or remote.
siteipstr	<String>	Read-only	The private IP address of this GSLB site.
sitepublicipstr	<String>	Read-only	The public IP address of this GSLB site.
sitemetricmepstatus	<String>	Read-only	Indicates the status of the site metric Metric Exchange connection at this GSLB site.
nwmetricmepstatus	<String>	Read-only	Indicates the status of the network metric Metric Exchange connection at this GSLB site.
sitetotalrequestbytes	<Double>	Read-only	Total number of request bytes received by the virtual servers represented by all GSLB services associated with this GSLB site.
siterequestbytesrate	<Double>	Read-only	Rate (/s) counter for sitetotalrequestbytes
sitetotalresponsebytes	<Double>	Read-only	Number of response bytes received by the virtual servers represented by all GSLB services associated with this GSLB site.
siteresponsebytesrate	<Double>	Read-only	Rate (/s) counter for sitetotalresponsebytes
sitetotalrequests	<Double>	Read-only	Total number of requests received by the virtual servers represented by all GSLB services associated with this GSLB site.
siterequestsrate	<Double>	Read-only	Rate (/s) counter for sitetotalrequests
sitetotalresponses	<Double>	Read-only	Number of responses received by the virtual servers represented by all GSLB services associated with this GSLB site.
siteresponsesrate	<Double>	Read-only	Rate (/s) counter for sitetotalresponses
sitecurclntconnections	<Double>	Read-only	Number of current client connections to the virtual servers represented by all GSLB services associated with this GSLB site.
sitecursrvrconnections	<Double>	Read-only	Number of current connections to the real servers behind the virtual servers represented by all GSLB services associated with this GSLB site.

Operations (click to see Properties)

GET (ALL) | GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/gslbsite

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/gslbsite?**args=sitename:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_**

Use this query-parameter to get gslbsite resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbsite": [ {
    "sitename":<String_value>,
    "siteresponsesrate":<Double_value>,
    "siteipstr":<String_value>,
    "sitetotalrequests":<Double_value>,
    "siterequestsrate":<Double_value>,
    "nwmetricmepstatus":<String_value>,
    "sitemetricexchange":<String_value>,
    "sitepublicip":<String_value>,
    "sitetotalresponsebytes":<Double_value>,
    "sitecursrvrconnections":<Double_value>,
    "sitepublicipstr":<String_value>,
    "sitetotalrequestbytes":<Double_value>,
    "siteip":<String_value>,
    "sitemepstatus":<String_value>,
    "sitecurclntconnections":<Double_value>,
    "sitetype":<String_value>,
    "nwmetricexchange":<String_value>,
    "persexchange":<String_value>,
    "siterequestbytesrate":<Double_value>,
    "siteresponsebytesrate":<Double_value>,
    "sitetotalresponses":<Double_value>,
    "sitemetricmepstatus":<String_value>
}]}
```

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/stat/gslbsite/sitename_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbsite": [ {
    "sitename":<String_value>,
    "siteresponsesrate":<Double_value>,
    "siteipstr":<String_value>,
    "sitetotalrequests":<Double_value>,
    "siterequestsrate":<Double_value>,
    "nwmetricmepstatus":<String_value>,
    "sitemetricexchange":<String_value>,
    "sitepublicip":<String_value>,
    "sitetotalresponsebytes":<Double_value>,
    "sitecursrvrconnections":<Double_value>,
    "sitepublicipstr":<String_value>,
    "sitetotalrequestbytes":<Double_value>,
    "siteip":<String_value>,
    "sitemepstatus":<String_value>,
    "sitecurclntconnections":<Double_value>,
    "sitetype":<String_value>,
    "nwmetricexchange":<String_value>,
    "persexchange":<String_value>,
    "siterequestbytesrate":<Double_value>,
    "siteresponsebytesrate":<Double_value>,
    "sitetotalresponses":<Double_value>,
    "sitemetricmepstatus":<String_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

gslbvserver

Jul 07, 2017

Statistics for Global Server Load Balancing Virtual Server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the GSLB virtual server for which to display statistics. If you do not specify a name, statistics are displayed for all GSLB virtual servers. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
establishedconn	<Double>	Read-only	Number of client connections in ESTABLISHED state.
inactsvcs	<Double>	Read-only	number of INACTIVE services bound to a vserver
vslbhealth	<Double>	Read-only	Health of the vserver. This gives percentage of UP services bound to this vserver.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
actsvcs	<Double>	Read-only	number of ACTIVE services bound to a vserver
tothits	<Double>	Read-only	Total vserver hits
hitsrate	<Double>	Read-only	Rate (/s) counter for tothits
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
sothreshold	<Double>	Read-only	Spill Over Threshold set on the VServer.
totspillovers	<Double>	Read-only	Number of times vserver experienced spill over.
totvserverdownbackuphits	<Double>	Read-only	Number of times traffic was diverted to backup vserver since primary vserver was DOWN.
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
curclntconnections	<Double>	Read-only	Number of current client connections.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.

Operations (click to see Properties)

GET (ALL) | GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/gslbvserver

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/gslbvserver?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_v**

Use this query-parameter to get gslbvserver resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbvserver": [ {
    "name":<String_value>,
    "curclntconnections":<Double_value>,
    "establishedconn":<Double_value>,
    "tothits":<Double_value>,
    "totalrequests":<Double_value>,
    "sothreshold":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "type":<String_value>,
    "hitsrate":<Double_value>,
    "cursrvrconnections":<Double_value>,
    "responsesrate":<Double_value>,
    "totspillovers":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "state":<String_value>,
    "vslbhealth":<Double_value>,
    "actsvcs":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "requestsrate":<Double_value>,
    "totvserverdownbackuphits":<Double_value>,
    "inactsvcs":<Double_value>
}]}
```

get

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/stat/gslbvserver/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "gslbvserver": [ {
    "name":<String_value>,
    "curclntconnections":<Double_value>,
    "establishedconn":<Double_value>,
    "tothits":<Double_value>,
    "totalrequests":<Double_value>,
    "sothreshold":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "type":<String_value>,
    "hitsrate":<Double_value>,
    "cursrvrconnections":<Double_value>,
    "responsesrate":<Double_value>,
    "totspillovers":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "state":<String_value>,
    "vslbhealth":<Double_value>,
    "actsvcs":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "requestsrate":<Double_value>,
    "totvserverdownbackuphits":<Double_value>,
    "inactsvcs":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

High Availability

Jul 07, 2017

High-availability commands.

Name	Description
hanode	Statistics for node resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

hanode

May 20, 2016

Statistics for node resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
hacurstatus	<String>	Read-only	Whether a NetScaler appliance is configured for high availability. Possible values are YES and NO. If the value is NO, the high availability statistics below are invalid.
hacurstate	<String>	Read-only	State of the HA node, based on its health, in a high availability setup. Possible values are: UP ? Indicates that the node is accessible and can function as either a primary or secondary node. DISABLED ? Indicates that the high availability status of the node has been manually disabled. Synchronization and propagation cannot take place between the peer nodes. INIT ? Indicates that the node is in the process of becoming part of the high availability configuration. PARTIALFAIL ? Indicates that one of the high availability monitored interfaces has failed because of a card or link failure. This state triggers a failover. COMPLETEFAIL ? Indicates that all the interfaces of the node are unusable, because the interfaces on which high availability monitoring is enabled are not connected or are manually disabled. This state triggers a failover. DUMB ? Indicates that the node is in listening mode. It does not participate in high availability transitions or transfer configuration from the peer node. This is a configured value, not a statistic. PARTIALFAILSSL ? Indicates that the SSL card has failed. This state triggers a failover. ROUTEMONITORFAIL ? Indicates that the route monitor has failed. This state triggers a failover.
hacurmasterstate	<String>	Read-only	Indicates the high availability state of the node. Possible values are: STAYSECONDARY ? Indicates that the selected node remains the secondary node in a high availability setup. In this case a forced failover does not change the state but, instead, returns an appropriate error message. This is a configured value and not a statistic. PRIMARY ? Indicates that the selected node is the primary node in a high availability setup. SECONDARY ? Indicates that the selected node is the secondary node in a high availability setup. CLAIMING ? Indicates that the secondary node is in the process of taking over as the primary node. This is the intermediate state in the transition of the secondary node to primary status. FORCE CHANGE - Indicates that the secondary node is forcibly changing its status to primary due to a forced failover issued on the secondary node.
transtime	<String>	Read-only	Time when the last master state transition occurred. You can use this statistic for debugging.
hatotpktrx	<Double>	Read-only	Number of heartbeat packets received from the peer node. Heartbeats are sent at regular intervals (default is 200 milliseconds) to determine the state of the peer node.
hapktrxrate	<Double>	Read-only	Rate (/s) counter for hatotpktrx
hatotpkttx	<Double>	Read-only	Number of heartbeat packets sent to the peer node. Heartbeats are sent at regular intervals (default is 200 milliseconds) to determine the state of the peer node.
hapkttxrate	<Double>	Read-only	Rate (/s) counter for hatotpkttx
haerrproptimeout	<Double>	Read-only	Number of times propagation timed out.
haerrsyncfailure	<Double>	Read-only	Number of times the configuration of the primary and secondary nodes failed to synchronize since that last transition. A synchronization failure results in mismatched configuration. It can be caused by a mismatch in the Remote Procedural Call (RPC) password on the two nodes forming the high availability pair.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/hanode

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/hanode?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get hanode resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "hanode": [ {
    "haerrsyncfailure":<Double_value>,
    "transtime":<String_value>,
    "hacurstatus":<String_value>,
    "hacurmasterstate":<String_value>,
    "hatotpkttx":<Double_value>,
    "hapktrxrate":<Double_value>,
    "haerrproptimeout":<Double_value>,
    "hapkttxrate":<Double_value>,
    "hacurstate":<String_value>,
    "hatotpktrx":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Ica

May 20, 2016

ICA statistics.

Name	Description
icapolicy	Statistics for ICA policy resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

icapolicy

Jul 07, 2017

Statistics for ICA policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the ica policy. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/icapolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/icapolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_val**

Use this query-parameter to get icapolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icapolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/icapolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "icapolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Ipsec

Jul 07, 2017

IPSEC statistics.

Name	Description
ipseccounters	Statistics for counters resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ipseccounters

Jul 07, 2017

Statistics for counters resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
ipsectotrxbytes	<Double>	Read-only	Bytes received during IPsec sessions.
ipsecrxbytesrate	<Double>	Read-only	Rate (/s) counter for ipsectotrxbytes
ipsectottxbytes	<Double>	Read-only	Bytes sent during IPsec sessions.
ipsectxbytesrate	<Double>	Read-only	Rate (/s) counter for ipsectottxbytes
ipsectotrxpkts	<Double>	Read-only	Packets received during IPsec sessions.
ipsecrxpktsrate	<Double>	Read-only	Rate (/s) counter for ipsectotrxpkts
ipsectottxpkts	<Double>	Read-only	Packets sent during IPsec sessions.
ipsectxpktsrate	<Double>	Read-only	Rate (/s) counter for ipsectottxpkts

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> ### Note
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/ipseccounters

Query-parameters:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

args

http://<netscaler-ip-address>/nitro/v1/stat/ipseccounters?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get ipseccounters resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ipseccounters": [ {
      "ipsecrxbytesrate":<Double_value>,
      "ipsectxbytesrate":<Double_value>,
      "ipsectottxbytes":<Double_value>,
      "ipsecrxpktsrate":<Double_value>,
      "ipsectotrxpkts":<Double_value>,
      "ipsectottxpkts":<Double_value>,
      "ipsectotrxbytes":<Double_value>,
      "ipsectxpktsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Load Balancing

May 20, 2016

Load Balancing statistics. The load balancing methods manage the selection of an appropriate physical server in a server farm.

Name	Description
lbvserver	Statistics for Load Balancing Virtual Server resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver

May 20, 2016

Statistics for Load Balancing Virtual Server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the virtual server. If no name is provided, statistical data of all configured virtual servers is displayed. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sortby	<String>	Read-write	use this argument to sort by specific key. Possible values =
sortorder	<String>	Read-write	use this argument to specify sort order. Default value: SORT_DESCENDING Possible values = ascending, descending
vsvrsurgecount	<Double>	Read-only	Number of requests waiting on this vserver.
establishedconn	<Double>	Read-only	Number of client connections in ESTABLISHED state.
inactsvcs	<Double>	Read-only	number of INACTIVE services bound to a vserver
vslbhealth	<Double>	Read-only	Health of the vserver. This gives percentage of UP services bound to this vserver.
primaryipaddress	<String>	Read-only	IP address of the vserver
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
actsvcs	<Double>	Read-only	number of ACTIVE services bound to a vserver
tothits	<Double>	Read-only	Total vserver hits
hitsrate	<Double>	Read-only	Rate (/s) counter for tothits
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes
totalpktsrecvd	<Double>	Read-only	Total number of packets received by this service or virtual server.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
totalpktssent	<Double>	Read-only	Total number of packets sent.
pktssentrate	<Double>	Read-only	Rate (/s) counter for totalpktssent
curclntconnections	<Double>	Read-only	Number of current client connections.
cursrvrconnections	<Double>	Read-only	Number of current connections to the actual servers behind the virtual server.
surgecount	<Double>	Read-only	Number of requests in the surge queue.
svcsurgecount	<Double>	Read-only	Total number of requests in the surge queues of all the services bound to this LB-vserver.
sothreshold	<Double>	Read-only	Spill Over Threshold set on the VServer.
totspillovers	<Double>	Read-only	Number of times vserver experienced spill over.
labelledconn	<Double>	Read-only	Number of Labeled connection on this vserver
pushlabel	<Double>	Read-only	Number of labels for this push vserver.
deferredreq	<Double>	Read-only	Number of deferred request on this vserver
deferredreqrate	<Double>	Read-only	Rate (/s) counter for deferredreq
invalidrequestresponse	<Double>	Read-only	Number invalid requests/responses on this vserver
invalidrequestresponsedropped	<Double>	Read-only	Number invalid requests/responses dropped on this vserver
totvserverdownbackuphits	<Double>	Read-only	Number of times traffic was diverted to backup vserver since primary vserver was DOWN.
curmptcpsessions	<Double>	Read-only	Current Multipath TCP sessions
cursubflowconn	<Double>	Read-only	Current Multipath TCP subflows

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/stat/lbvserver

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/lbvserver?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_val**

Use this query-parameter to get lbvserver resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver": [ {
    "name":<String_value>,
    "curclntconnections":<Double_value>,
    "establishedconn":<Double_value>,
    "totalpktssent":<Double_value>,
    "labelledconn":<Double_value>,
    "tothits":<Double_value>,
    "totalrequests":<Double_value>,
    "sothreshold":<Double_value>,
    "cursubflowconn":<Double_value>,
    "surgecount":<Double_value>,
    "responsebytesrate":<Double_value>,
    "invalidrequestresponsedropped":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "type":<String_value>,
    "hitsrate":<Double_value>,
    "cursrvrconnections":<Double_value>,
    "pktsrecvdrate":<Double_value>,
    "primaryipaddress":<String_value>,
    "vsvrsurgecount":<Double_value>,
    "pushlabel":<Double_value>,
    "responsesrate":<Double_value>,
    "deferredreq":<Double_value>,
    "curmptcpsessions":<Double_value>,
    "totspillovers":<Double_value>,
    "svcsurgecount":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "invalidrequestresponse":<Double_value>,
    "state":<String_value>,
    "vslbhealth":<Double_value>,
    "deferredreqrate":<Double_value>,
    "actsvcs":<Double_value>,
    "totalpktsrecvd":<Double_value>,
    "pktssentrate":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>,
    "totvserverdownbackuphits":<Double_value>,
    "inactsvcs":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/lbvserver/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "lbvserver": [ {
    "name":<String_value>,
    "curclntconnections":<Double_value>,
    "establishedconn":<Double_value>,
    "totalpktssent":<Double_value>,
    "labelledconn":<Double_value>,
    "tothits":<Double_value>,
    "totalrequests":<Double_value>,
    "sothreshold":<Double_value>,
    "cursubflowconn":<Double_value>,
    "surgecount":<Double_value>,
    "responsebytesrate":<Double_value>,
    "invalidrequestresponsedropped":<Double_value>,
    "totalresponses":<Double_value>,
    "requestbytesrate":<Double_value>,
    "type":<String_value>,
    "hitsrate":<Double_value>,
    "cursrvrconnections":<Double_value>,
    "pktsrecvdrate":<Double_value>,
    "primaryipaddress":<String_value>,
    "vsvrsurgecount":<Double_value>,
    "pushlabel":<Double_value>,
    "responsesrate":<Double_value>,
    "deferredreq":<Double_value>,
    "curmptcpsessions":<Double_value>,
    "totspillovers":<Double_value>,
    "svcsurgecount":<Double_value>,
    "totalrequestbytes":<Double_value>,
    "invalidrequestresponse":<Double_value>,
    "state":<String_value>,
    "vslbhealth":<Double_value>,
    "deferredreqrate":<Double_value>,
    "actsvcs":<Double_value>,
    "totalpktsrecvd":<Double_value>,
    "pktssentrate":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>,
    "totvserverdownbackuphits":<Double_value>,
    "inactsvcs":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

LLDP

May 20, 2016

Link Layer Discovery Protocol.

Name	Description
lldp	Statistics for lldp.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lldp

May 20, 2016

Statistics for lldp.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
ifnum	<String>	Read-write	LLDP Statistics per interfaces.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
rxportframestotal	<Double>	Read-only	Total LLDP Packets received.
rxportframesrate	<Double>	Read-only	Rate (/s) counter for rxportframestotal
rxportbytestotal	<Double>	Read-only	Total LLDP bytes received
rxportbytesrate	<Double>	Read-only	Rate (/s) counter for rxportbytestotal
txportframestotal	<Double>	Read-only	Total LLDP Packets transmitted
txportframesrate	<Double>	Read-only	Rate (/s) counter for txportframestotal
txportbytestotal	<Double>	Read-only	Total LLDP bytes transmitted.
txportbytesrate	<Double>	Read-only	Rate (/s) counter for txportbytestotal
rxportframeerrors	<Double>	Read-only	Total errors in LLDP packets.
rxportframeerrorsrate	<Double>	Read-only	Rate (/s) counter for rxportframeerrors
rxporttlvsdiscardedtotal	<Double>	Read-only	Total discarded LLDP packets.
rxporttlvsdiscardedrate	<Double>	Read-only	Rate (/s) counter for rxporttlvsdiscardedtotal
rxporttlvsunrecognizedtotal	<Double>	Read-only	Total TLVs not Recognised.
rxporttlvsunrecognizedrate	<Double>	Read-only	Rate (/s) counter for rxporttlvsunrecognizedtotal

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/lldp

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/lldp?**args=ifnum:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get lldp resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lldp": [ {
    "ifnum":<String_value>,
    "rxporttlvsdiscardedtotal":<Double_value>,
    "rxportframeserrors":<Double_value>,
    "rxportframeserrorsrate":<Double_value>,
    "rxportframestotal":<Double_value>,
    "txportbytestotal":<Double_value>,
    "txportframestotal":<Double_value>,
    "rxporttlvsunrecognizedtotal":<Double_value>,
    "rxportbytesrate":<Double_value>,
    "rxportbytestotal":<Double_value>,
    "rxporttlvsdiscardedrate":<Double_value>,
    "txportbytesrate":<Double_value>,
    "rxporttlvsunrecognizedrate":<Double_value>,
    "rxportframesrate":<Double_value>,
    "txportframesrate":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/lldp/ifnum_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lldp": [ {
    "ifnum":<String_value>,
    "rxporttlvsdiscardedtotal":<Double_value>,
    "rxportframeserrors":<Double_value>,
    "rxportframeserrorsrate":<Double_value>,
    "rxportframestotal":<Double_value>,
    "txportbytestotal":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "txportframestotal":<Double_value>,
    "rxporttlvsunrecognizedtotal":<Double_value>,
    "rxportbytesrate":<Double_value>,
    "rxportbytestotal":<Double_value>,
    "rxporttlvsdiscardedrate":<Double_value>,
    "txportbytesrate":<Double_value>,
    "rxporttlvsunrecognizedrate":<Double_value>,
    "rxportframesrate":<Double_value>,
    "txportframesrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Lsn

May 20, 2016

Large Scale NAT commands

Name	Description
lsn	Statistics for lsn.
lsndslite	
lsngroup	Statistics for LSN group resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsn

May 20, 2016
Statistics for lsn.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
lsntottcprxpkts	<Double>	Read-only	Number of LSN TCP Received packets.
lsntcprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntottcprxpkts
lsntottcprxbytes	<Double>	Read-only	Number of LSN TCP Received bytes.
lsntcprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntottcprxbytes
lsntottcptxpkts	<Double>	Read-only	Number of LSN TCP Transmitted packets.
lsntcptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntottcptxpkts
lsntottcptxbytes	<Double>	Read-only	Number of LSN TCP Transmitted bytes.
lsntcptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntottcptxbytes
lsntottcpdrppkts	<Double>	Read-only	Number of LSN TCP Dropped packets.
lsntcpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntottcpdrppkts
lsncurtcpsessions	<Double>	Read-only	Number of LSN TCP Current Sessions.
lsncurtcpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurtcpsessions
lsntotudprxpkts	<Double>	Read-only	Number of LSN UDP Received packets.
lsnudprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotudprxpkts
lsntotudprxbytes	<Double>	Read-only	Number of LSN UDP Received bytes.
lsnudprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotudprxbytes
lsntotudptxpkts	<Double>	Read-only	Number of LSN UDP Transmitted packets.
lsnudptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotudptxpkts
lsntotudptxbytes	<Double>	Read-only	Number of LSN UDP Transmitted bytes.
lsnudptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotudptxbytes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
lsnudpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotudpdrppkts
lsncurudpsessions	<Double>	Read-only	Number of LSN UDP Current Sessions.
lsncurudpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurudpsessions
lsntoticmprxpkts	<Double>	Read-only	Number of LSN ICMP Received packets.
lsnicmprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntoticmprxpkts
lsntoticmprxbytes	<Double>	Read-only	Number of LSN ICMP Received bytes.
lsnicmprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntoticmprxbytes
lsntoticmptxpkts	<Double>	Read-only	Number of LSN ICMP Transmitted packets.
lsnicmptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntoticmptxpkts
lsntoticmptxbytes	<Double>	Read-only	Number of LSN ICMP Transmitted bytes.
lsnicmptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntoticmptxbytes
lsntoticmpdrppkts	<Double>	Read-only	Number of LSN ICMP Dropped packets.
lsnicmpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntoticmpdrppkts
lsncuricmpsessions	<Double>	Read-only	Number of LSN ICMP Current Sessions.
lsncuricmpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncuricmpsessions
lsncursessions	<Double>	Read-only	Current number of LSN sessions.
lsncursessionsrate	<Double>	Read-only	Rate (/s) counter for lsncursessions
lsncursubscribers	<Double>	Read-only	Current number of LSN subscribers.
lsncursubscribersrate	<Double>	Read-only	Rate (/s) counter for lsncursubscribers

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN= *<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/lsn

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/lsn?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get lsn resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsn": [ {
    "lsncursubscribersrate":<Double_value>,
    "lsntotudptxbytes":<Double_value>,
    "lsncuricmpsessionsrate":<Double_value>,
    "lsncursessions":<Double_value>,
    "lsntottcptxbytes":<Double_value>,
    "lsnudptxbytesrate":<Double_value>,
    "lsntotudpdrppkts":<Double_value>,
    "lsntottcpdrppkts":<Double_value>,
    "lsntoticmptxpkts":<Double_value>,
    "lsntcptxbytesrate":<Double_value>,
    "lsnudprxbytesrate":<Double_value>,
    "lsntcpdrppktsrate":<Double_value>,
    "lsntotudprxbytes":<Double_value>,
    "lsnicmpdrppktsrate":<Double_value>,
    "lsntottcprxbytes":<Double_value>,
    "lsntoticmptxbytes":<Double_value>,
    "lsntoticmprxpkts":<Double_value>,
    "lsnicmptxpktsrate":<Double_value>,
    "lsnicmprxbytesrate":<Double_value>,
    "lsntoticmpdrppkts":<Double_value>,
    "lsntcprxpktsrate":<Double_value>,
    "lsntottcptxpkts":<Double_value>,
    "lsncurtcpsessionsrate":<Double_value>,
    "lsncursubscribers":<Double_value>,
    "lsncurudpsessionsrate":<Double_value>,
    "lsnudptxpktsrate":<Double_value>,
    "lsntottcprxpkts":<Double_value>,
    "lsnicmptxbytesrate":<Double_value>,
    "lsntotudprxpkts":<Double_value>,
    "lsncursessionsrate":<Double_value>,
    "lsntcptxpktsrate":<Double_value>,
    "lsnudprxpktsrate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "lsntotudptxpkts":<Double_value>,
    "lsncurudpsessions":<Double_value>,
    "lsnudpdrppktsrate":<Double_value>,
    "lsntoticmprxbytes":<Double_value>,
    "lsncuricmpsessions":<Double_value>,
    "lsncurtcpsessions":<Double_value>,
    "lsntcprxbytesrate":<Double_value>,
    "lsnicmprxpktsrate":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsndslite

Jul 07, 2017
Read/write properties

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
lsntotdsliterxpkts	<Double>	Read-only	Total number of LSN DS-Lite rx pkts.
lsndsliterxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliterxpkts
lsntotdsliterxbytes	<Double>	Read-only	Total number of LSN DS-Lite rx bytes.
lsndsliterxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliterxbytes
lsntotdslitetxpkts	<Double>	Read-only	Total number of LSN DS-Lite tx pkts.
lsndslitetxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetxpkts
lsntotdslitetxbytes	<Double>	Read-only	Total number of LSN DS-Lite tx bytes.
lsndslitetxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetxbytes
lsntotdslitetcprxpkts	<Double>	Read-only	Number of LSN DS-Lite TCP Received packets.
lsndslitetcprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetcprxpkts
lsntotdslitetcprxbytes	<Double>	Read-only	Number of LSN DS-Lite TCP Received bytes.
lsndslitetcprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetcprxbytes
lsntotdslitetcptxpkts	<Double>	Read-only	Number of LSN DS-Lite TCP Transmitted packets.
lsndslitetcptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetcptxpkts
lsntotdslitetcptxbytes	<Double>	Read-only	Number of LSN DS-Lite TCP Transmitted bytes.
lsndslitetcptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetcptxbytes
lsntotdslitetcpdrppkts	<Double>	Read-only	Number of LSN DS-Lite TCP Dropped packets.
lsndslitetcpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdslitetcpdrppkts
lsncurdslitetcpsessions	<Double>	Read-only	Number of LSN DS-Lite TCP Current Sessions.
lsncurdslitetcpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurdslitetcpsessions
lsntotdsliteudprxpkts	<Double>	Read-only	Number of LSN DS-Lite UDP Received packets.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
lsntotdsliteudprxbytes	<Double>	Read-only	Number of LSN DS-Lite UDP Received bytes.
lsndsliteudprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteudprxbytes
lsntotdsliteudptxpkts	<Double>	Read-only	Number of LSN DS-Lite UDP Transmitted packets.
lsndsliteudptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteudptxpkts
lsntotdsliteudptxbytes	<Double>	Read-only	Number of LSN DS-Lite UDP Transmitted bytes.
lsndsliteudptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteudptxbytes
lsntotdsliteudpdrppkts	<Double>	Read-only	Number of LSN DS-Lite UDP Dropped packets.
lsndsliteudpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteudpdrppkts
lsncurdsliteudpsessions	<Double>	Read-only	Number of LSN DS-Lite UDP Current Sessions.
lsncurdsliteudpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurdsliteudpsessions
lsntotdsliteicmprxpkts	<Double>	Read-only	Number of LSN DS-Lite ICMP Received packets.
lsndsliteicmprxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteicmprxpkts
lsntotdsliteicmprxbytes	<Double>	Read-only	Number of LSN DS-Lite ICMP Received bytes.
lsndsliteicmprxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteicmprxbytes
lsntotdsliteicmptxpkts	<Double>	Read-only	Number of LSN DS-Lite ICMP Transmitted packets.
lsndsliteicmptxpktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteicmptxpkts
lsntotdsliteicmptxbytes	<Double>	Read-only	Number of LSN DS-Lite ICMP Transmitted bytes.
lsndsliteicmptxbytesrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteicmptxbytes
lsntotdsliteicmpdrppkts	<Double>	Read-only	Number of LSN DS-Lite ICMP Dropped packets.
lsndsliteicmpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsntotdsliteicmpdrppkts
lsncurdsliteicmpsessions	<Double>	Read-only	Number of LSN DS-Lite ICMP Current Sessions.
lsncurdsliteicmpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurdsliteicmpsessions
lsncurdslitesessions	<Double>	Read-only	Current number of LSN DS-Lite sessions.
lsncurdslitesessionsrate	<Double>	Read-only	Rate (/s) counter for lsncurdslitesessions
lsndslitecursubscribers	<Double>	Read-only	Current number of LSN DS-Lite subscribers.
lsndslitecursubscribersrate	<Double>	Read-only	Rate (/s) counter for lsndslitecursubscribers

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/lsndslite

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/lsndslite?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get lsndslite resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsndslite": [ {
    "lsntotdsliteudpdrppkts":<Double_value>,
    "lsndsliteudprxpktsrate":<Double_value>,
    "lsntotdsliterxpkts":<Double_value>,
    "lsntotdslitetxbytes":<Double_value>,
    "lsncurdsliteicmpsessionsrate":<Double_value>,
    "lsntotdsliteudptxbytes":<Double_value>,
    "lsntotdsliteicmptxpkts":<Double_value>,
    "lsndslitetxpktsrate":<Double_value>,
    "lsntotdsliteicmptxbytes":<Double_value>,
    "lsntotdsliterxbytes":<Double_value>,
    "lsntotdsliteicmpdrppkts":<Double_value>,
    "lsncurdslitetcpsessionsrate":<Double_value>,
    "lsndsliteudptxbytesrate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "lsntotdsliteudprxbytes":<Double_value>,
    "lsndsliterxbytesrate":<Double_value>,
    "lsncurdsliteicmpsessions":<Double_value>,
    "lsndsliteicmprxbytesrate":<Double_value>,
    "lsndsliteudpdrppktsrate":<Double_value>,
    "lsntotdslitetcpdrppkts":<Double_value>,
    "lsntotdsliteicmprxbytes":<Double_value>,
    "lsncurdslitesessionsrate":<Double_value>,
    "lsntotdsliteicmprxpkts":<Double_value>,
    "lsntotdslitetxpkts":<Double_value>,
    "lsncurdsliteudpsessionsrate":<Double_value>,
    "lsndslitetcptxpktsrate":<Double_value>,
    "lsntotdslitetcptxpkts":<Double_value>,
    "lsndsliteicmptxpktsrate":<Double_value>,
    "lsndsliteicmpdrppktsrate":<Double_value>,
    "lsntotdsliteudprxpkts":<Double_value>,
    "lsncurdslitetcpsessions":<Double_value>,
    "lsndslitetxbytesrate":<Double_value>,
    "lsntotdslitetcprxbytes":<Double_value>,
    "lsndsliteicmptxbytesrate":<Double_value>,
    "lsndsliteicmprxpktsrate":<Double_value>,
    "lsncurdslitesessions":<Double_value>,
    "lsndslitetcprxbytesrate":<Double_value>,
    "lsndslitecursubscribers":<Double_value>,
    "lsntotdslitetcprxpkts":<Double_value>,
    "lsndsliterxpktsrate":<Double_value>,
    "lsncurdsliteudpsessions":<Double_value>,
    "lsndslitetcprxpktsrate":<Double_value>,
    "lsntotdsliteudptxpkts":<Double_value>,
    "lsntotdslitetcptxbytes":<Double_value>,
    "lsndslitetcpdrppktsrate":<Double_value>,
    "lsndslitetcptxbytesrate":<Double_value>,
    "lsndsliteudprxbytesrate":<Double_value>,
    "lsndsliteudptxpktsrate":<Double_value>,
    "lsndslitecursubscribersrate":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngroup

May 20, 2016

Statistics for LSN group resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
groupname	<String>	Read-write	Name of the LSN Group. Minimum length = 1 Maximum length = 127
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
lsngrpcursessions	<Double>	Read-only	Number of Current Sessions for LSN group
lsngrpcursessionsrate	<Double>	Read-only	Rate (/s) counter for lsngrpcursessions
lsngrptottranslbytes	<Double>	Read-only	Number of Translated Bytes for LSN group
lsngrptranslbytesrate	<Double>	Read-only	Rate (/s) counter for lsngrptottranslbytes
lsngrptottranslpkts	<Double>	Read-only	Number of Translated Pkts for LSN group
lsngrptranslpktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptottranslpkts
lsngrptottcptranslpkts	<Double>	Read-only	Number of TCP Translated Pkts for LSN group
lsngrptcptranslpktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptottcptranslpkts
lsngrptottcptranslbytes	<Double>	Read-only	Number of TCP Translated Bytes for LSN group
lsngrptcptranslbytesrate	<Double>	Read-only	Rate (/s) counter for lsngrptottcptranslbytes
lsngrptottcpdrppkts	<Double>	Read-only	Number of TCP Dropped Pkts for LSN group
lsngrptcpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptottcpdrppkts
lsngrpcurtcpsessions	<Double>	Read-only	Number of TCP Current Sessions for LSN group
lsngrpcurtcpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsngrpcurtcpsessions
lsngrptotudptranslpkts	<Double>	Read-only	Number of UDP Translated Pkts for LSN group
lsngrpudptranslpktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptotudptranslpkts
lsngrptotudptranslbytes	<Double>	Read-only	Number of UDP Translated Bytes for LSN group
lsngrpudptranslbytesrate	<Double>	Read-only	Rate (/s) counter for lsngrptotudptranslbytes
lsngrptotudpdrppkts	<Double>	Read-only	Number of UDP Dropped Pkts for LSN group
lsngrpudpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptotudpdrppkts
lsngrpcurudpsessions	<Double>	Read-only	Number of UDP Current Sessions for LSN group
lsngrpcurudpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsngrpcurudpsessions
lsngrptoticmptranslpkts	<Double>	Read-only	Number of ICMP Translated Pkts for LSN group

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lsngrpicmptranslpktsrate **Name**	<Double> **Data Type**	Read-only **Permissions**	Rate (/s) counter for lsngrptoticmptranslpkts **Description**
lsngrptoticmptranslbytes	<Double>	Read-only	Number of ICMP Translated Bytes for LSN group
lsngrpicmptranslbytesrate	<Double>	Read-only	Rate (/s) counter for lsngrptoticmptranslbytes
lsngrptoticmpdrppkts	<Double>	Read-only	Number of ICMP Dropped Pkts for LSN group
lsngrpicmpdrppktsrate	<Double>	Read-only	Rate (/s) counter for lsngrptoticmpdrppkts
lsngrpcuricmpsessions	<Double>	Read-only	Number of ICMP Current Sessions for LSN group
lsngrpcuricmpsessionsrate	<Double>	Read-only	Rate (/s) counter for lsngrpcuricmpsessions
lsngrpcursubscribers	<Double>	Read-only	Number of ICMP Current Sessions for LSN group
lsngrpcursubscribersrate	<Double>	Read-only	Rate (/s) counter for lsngrpcursubscribers

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/lsngroup

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/lsngroup?**args=groupname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<Strin**

Use this query-parameter to get lsngroup resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup": [ {
    "groupname":<String_value>,
    "lsngrptcptranslbytesrate":<Double_value>,
    "lsngrptottranslpkts":<Double_value>,
    "lsngrptoticmptranslpkts":<Double_value>,
    "lsngrpcurtcpsessions":<Double_value>,
    "lsngrpicmptranslbytesrate":<Double_value>,
    "lsngrptotudptranslpkts":<Double_value>,
    "lsngrptottcpdrppkts":<Double_value>,
    "lsngrptranslbytesrate":<Double_value>,
    "lsngrptotudptranslbytes":<Double_value>,
    "lsngrptoticmptranslbytes":<Double_value>,
    "lsngrpcursubscribersrate":<Double_value>,
    "lsngrptottranslbytes":<Double_value>,
    "lsngrptoticmpdrppkts":<Double_value>,
    "lsngrptcptranslpktsrate":<Double_value>,
    "lsngrpudptranslpktsrate":<Double_value>,
    "lsngrpcursubscribers":<Double_value>,
    "lsngrpudptranslbytesrate":<Double_value>,
    "lsngrpcuricmpsessionsrate":<Double_value>,
    "lsngrpcurudpsessions":<Double_value>,
    "lsngrptottcptranslpkts":<Double_value>,
    "lsngrptotudpdrppkts":<Double_value>,
    "lsngrpicmptranslpktsrate":<Double_value>,
    "lsngrpcuricmpsessions":<Double_value>,
    "lsngrpcurtcpsessionsrate":<Double_value>,
    "lsngrpcurudpsessionsrate":<Double_value>,
    "lsngrptranslpktsrate":<Double_value>,
    "lsngrptottcptranslbytes":<Double_value>,
    "lsngrpicmpdrppktsrate":<Double_value>,
    "lsngrpcursessionsrate":<Double_value>,
    "lsngrpudpdrppktsrate":<Double_value>,
    "lsngrptcpdrppktsrate":<Double_value>,
    "lsngrpcursessions":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/lsngroup/groupname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "lsngroup": [ {
    "groupname":<String_value>,
    "lsngrptcptranslbytesrate":<Double_value>,
    "lsngrptottranslpkts":<Double_value>,
    "lsngrptoticmptranslpkts":<Double_value>,
    "lsngrpcurtcpsessions":<Double_value>,
    "lsngrpicmptranslbytesrate":<Double_value>,
    "lsngrptotudptranslpkts":<Double_value>,
    "lsngrptottcpdrppkts":<Double_value>,
    "lsngrptranslbytesrate":<Double_value>,
    "lsngrptotudptranslbytes":<Double_value>,
    "lsngrptoticmptranslbytes":<Double_value>,
    "lsngrpcursubscribersrate":<Double_value>,
    "lsngrptottranslbytes":<Double_value>,
    "lsngrptoticmpdrppkts":<Double_value>,
    "lsngrptcptranslpktsrate":<Double_value>,
    "lsngrpudptranslpktsrate":<Double_value>,
    "lsngrpcursubscribers":<Double_value>,
    "lsngrpudptranslbytesrate":<Double_value>,
    "lsngrpcuricmpsessionsrate":<Double_value>,
    "lsngrpcurudpsessions":<Double_value>,
    "lsngrptottcptranslpkts":<Double_value>,
    "lsngrptotudpdrppkts":<Double_value>,
    "lsngrpicmptranslpktsrate":<Double_value>,
    "lsngrpcuricmpsessions":<Double_value>,
    "lsngrpcurtcpsessionsrate":<Double_value>,
    "lsngrpcurudpsessionsrate":<Double_value>,
    "lsngrptranslpktsrate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "lsngrptottcptranslbytes":<Double_value>,
    "lsngrpicmpdrppktsrate":<Double_value>,
    "lsngrpcursessionsrate":<Double_value>,
    "lsngrpudpdrppktsrate":<Double_value>,
    "lsngrptcpdrppktsrate":<Double_value>,
    "lsngrpcursessions":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mediaclassification

May 20, 2016

Media Classification feature is used to show the stats of different media types that are being served by the netscaler

Name	Description
mediaclassification	Statistics for mediaclassification.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

mediaclassification

May 20, 2016

Statistics for mediaclassification.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
mcencvideo	<Double>	Read-only	Total number of encrypted video streams
mcencvideorate	<Double>	Read-only	Rate (/s) counter for mcencvideo
mctotother	<Double>	Read-only	
mctotvideo	<Double>	Read-only	
mctotaudio	<Double>	Read-only	
mcmssmthstrmvid	<Double>	Read-only	This tells the total number of Microsoft SmoothStreaming videos served by NS
mcmssmthstrmvidrate	<Double>	Read-only	Rate (/s) counter for mcmssmthstrmvid
mcmssmthstrvidpl	<Double>	Read-only	This tells the total number of Microsoft SmoothStreaming playlist served by NS
mcmssmthstrvidplrate	<Double>	Read-only	Rate (/s) counter for mcmssmthstrvidpl
mccaplelivestrmngvid	<Double>	Read-only	This tells the total number of AppleLive videos served by NS
mccaplelivestrmngvidrate	<Double>	Read-only	Rate (/s) counter for mccaplelivestrmngvid
mccapplelivestrmngvidpl	<Double>	Read-only	This tells the total number of AppleLive Playlist served by NS
mccapplelivestrmngvidplrate	<Double>	Read-only	Rate (/s) counter for mccapplelivestrmngvidpl
mcadtsaudio	<Double>	Read-only	This tells the total number of ADTS audios served by NS
mcadtsaudiorate	<Double>	Read-only	Rate (/s) counter for mcadtsaudio
mcaacaudio	<Double>	Read-only	This tells the total number of AAC audios served by NS
mcaacaudiorate	<Double>	Read-only	Rate (/s) counter for mcaacaudio
mcflvvid	<Double>	Read-only	This tells the total number of FLV videos served by NS
mcflvvidrate	<Double>	Read-only	Rate (/s) counter for mcflvvid
mcmp4vid	<Double>	Read-only	This tells the total number of MP4 videos served by NS
mcmp4vidrate	<Double>	Read-only	Rate (/s) counter for mcmp4vid
mc3pvid	<Double>	Read-only	This tells the total number of 3GP videos served by NS
mc3pvidrate	<Double>	Read-only	Rate (/s) counter for mc3pvid

© 1999-2017 Citrix Systems, Inc. All rights reserved.

mcyoutubedash	<Double>	Read-only	This tells the total number of YouTube DASH served by NS
Name	**Data Type**	**Permissions**	**Description**
mcyoutubedashrate	<Double>	Read-only	Rate (/s) counter for mcyoutubedash
mcmssmthstrmvidbytes	<Double>	Read-only	This tells the total number of Microsoft SmoothStreaming video bytes served by NS
mcmssmthstrmvidbytesrate	<Double>	Read-only	Rate (/s) counter for mcmssmthstrmvidbytes
mcmssmthstrmplvidbytespl	<Double>	Read-only	This tells the total number of Microsoft SmoothStreaming playlist bytes served by NS
mcmssmthstrmplvidbytesplrate	<Double>	Read-only	Rate (/s) counter for mcmssmthstrmplvidbytespl
mcapplelivestreamingvidbytes	<Double>	Read-only	This tells the total number of AppleLive video bytes served by NS
mcapplelivestreamingvidbytesrate	<Double>	Read-only	Rate (/s) counter for mcapplelivestreamingvidbytes
mcapplelivestreamingplaylistvidbytespl	<Double>	Read-only	This tells the total number of AppleLive Playlist bytes served by NS
mcapplelivestreamingplaylistvidbytesplrate	<Double>	Read-only	Rate (/s) counter for mcapplelivestreamingplaylistvidbytespl
mcadtsaudiobytes	<Double>	Read-only	This tells the total number of ADTS audio bytes served by NS
mcadtsaudiobytesrate	<Double>	Read-only	Rate (/s) counter for mcadtsaudiobytes
mcaacaudiobytes	<Double>	Read-only	This tells the total number of AAC bytes served by NS
mcaacaudiobytesrate	<Double>	Read-only	Rate (/s) counter for mcaacaudiobytes
mcflvvidbytes	<Double>	Read-only	This tells the total number of FLV bytes served by NS
mcflvvidbytesrate	<Double>	Read-only	Rate (/s) counter for mcflvvidbytes
mcmp4vidbytes	<Double>	Read-only	This tells the total number of MP4 video bytes served by NS
mcmp4vidbytesrate	<Double>	Read-only	Rate (/s) counter for mcmp4vidbytes
mc3gpvidbytes	<Double>	Read-only	This tells the total number of 3GP bytes served by NS
mc3gpvidbytesrate	<Double>	Read-only	Rate (/s) counter for mc3gpvidbytes
mcyoutubedashbytes	<Double>	Read-only	This tells the total number of YouTube DASH bytes served by NS
mcyoutubedashbytesrate	<Double>	Read-only	Rate (/s) counter for mcyoutubedashbytes
mcandroid	<Double>	Read-only	Total number of android requests to netscaler
mcandroidrate	<Double>	Read-only	Rate (/s) counter for mcandroid
mclaptopdesktp	<Double>	Read-only	Total number of laptop/desktop requests to netscaler
mclaptopdesktprate	<Double>	Read-only	Rate (/s) counter for mclaptopdesktp
mcios	<Double>	Read-only	Total number of IOS requests to netscaler
mciosrate	<Double>	Read-only	Rate (/s) counter for mcios
mcother	<Double>	Read-only	Total number of other mobile device requests to netscaler

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
mcotherrate	<Double>	Read-only	Rate (/s) counter for mcother
mcunidentified	<Double>	Read-only	Total number of unidentified requests to netscaler
mcunidentifiedrate	<Double>	Read-only	Rate (/s) counter for mcunidentified
mcenchls	<Double>	Read-only	Total number of encrypted AppleLive video streams
mcenchlsrate	<Double>	Read-only	Rate (/s) counter for mcenchls
mcencdash	<Double>	Read-only	Total number of encrypted Youtube DASH video streams
mcencdashrate	<Double>	Read-only	Rate (/s) counter for mcencdash
mcencandroid	<Double>	Read-only	Total number of encrypted video streams requested by Android devices
mcencandroidrate	<Double>	Read-only	Rate (/s) counter for mcencandroid
mcencios	<Double>	Read-only	Total number of encrypted video streams requested by IOS devices
mcenciosrate	<Double>	Read-only	Rate (/s) counter for mcencios
mcencother	<Double>	Read-only	Total number of encrypted video streams requested by non-Android and non-IOS devices
mcencotherrate	<Double>	Read-only	Rate (/s) counter for mcencother

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/mediaclassification

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/mediaclassification?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get mediaclassification resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "mediaclassification": [ {
    "mcmp4vid":<Double_value>,
    "mcvideorate":<Double_value>,
    "mc3pvid":<Double_value>,
    "mcencandroid":<Double_value>,
    "mcflvvid":<Double_value>,
    "mcunidentifiedrate":<Double_value>,
    "mcencvideo":<Double_value>,
    "mclaptopdesktprate":<Double_value>,
    "mcyoutubedash":<Double_value>,
    "mccaplelivestrmngvidrate":<Double_value>,
    "mcaacaudiobytes":<Double_value>,
    "mcenchls":<Double_value>,
    "mcother":<Double_value>,
    "mcflvvidbytesrate":<Double_value>,
    "mc3gpvidbytes":<Double_value>,
    "mcencvideorate":<Double_value>,
    "mctotaudio":<Double_value>,
    "mcenciosrate":<Double_value>,
    "mcadtsaudiobytes":<Double_value>,
    "mcaacaudiobytesrate":<Double_value>,
    "mcandroid":<Double_value>,
    "mccaplelivestrmngvid":<Double_value>,
    "mciosrate":<Double_value>,
    "mctotvideo":<Double_value>,
    "mcmssmthstrvidplrate":<Double_value>,
    "mcapplelivestreamingvidbytesrate":<Double_value>,
    "mcyoutubedashbytesrate":<Double_value>,
    "mcapplelivestreamingplaylistvidbytespl":<Double_value>,
    "mcmp4vidbytes":<Double_value>,
    "mcmssmthstrmvidbytesrate":<Double_value>,
    "mc3gpvidbytesrate":<Double_value>,
    "mcaudiorate":<Double_value>,
    "mcyoutubedashbytes":<Double_value>,
    "mcmp4vidrate":<Double_value>,
    "mcmp4vidbytesrate":<Double_value>,
    "mcunidentified":<Double_value>,
    "mcios":<Double_value>,
    "mc3pvidrate":<Double_value>,
    "mccapplelivestrmngvidplrate":<Double_value>,
    "mctoherrate":<Double_value>,
    "mcotherrate":<Double_value>,
    "mcenchlsrate":<Double_value>,
    "mcmssmthstrmplvidbytesplrate":<Double_value>,
    "mccapplelivestrmngvidpl":<Double_value>,
    "mcencandroidrate":<Double_value>,
    "mcmssmthstrmvid":<Double_value>,
    "mcencdash":<Double_value>,
    "mcandroidrate":<Double_value>,
    "mcaacaudiorate":<Double_value>,
    "mctotother":<Double_value>,
    "mcencdashrate":<Double_value>,
    "mcmssmthstrmvidbytes":<Double_value>,
    "mcencios":<Double_value>,
    "mcadtsaudiorate":<Double_value>,
    "mcaacaudio":<Double_value>,
    "mcflvvidrate":<Double_value>,
    "mcencotherrate":<Double_value>,
    "mcapplelivestreamingplaylistvidbytesplrate":<Double_value>,
    "mcmssmthstrvidpl":<Double_value>,
    "mcflvvidbytes":<Double_value>,
    "mcmssmthstrmvidrate":<Double_value>,
    "mcyoutubedashrate":<Double_value>,
    "mcapplelivestreamingvidbytes":<Double_value>,
    "mcadtsaudiobytesrate":<Double_value>,
    "mcmssmthstrmplvidbytespl":<Double_value>,
    "mcadtsaudio":<Double_value>,
    "mclaptopdesktp":<Double_value>,
    "mcencother":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Network

May 20, 2016

Networking commands.

Name	Description
Interface	Statistics for interface resource.
bridge	Statistics for bridge resource.
inat	Statistics for inbound nat resource.
inatsession	Statistics for stateful INAT resource.
nat64	Statistics for nat64 config resource.
rnat	Statistics for RNAT configured route resource.
rnatip	Statistics for RNAT ipaddress resource.
tunnelip	Statistics for TUNNEL Remote ipaddress resource.
tunnelip6	Statistics for TUNNEL Remote ip6address resource.
vlan	Statistics for "VLAN" resource.
vpath	Statistics for Vpath resource.
vxlan	Statistics for "VXLAN" resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

bridge

May 20, 2016

Statistics for bridge resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
tcptotbdgmacmoved	<Double>	Read-only	The number of times bridging registered MAC moved
tcpbdgmacmovedrate	<Double>	Read-only	Rate (/s) counter for tcptotbdgmacmoved
tcptotbdgcollisions	<Double>	Read-only	The number of bridging table collisions
tcpbdgcollisionsrate	<Double>	Read-only	Rate (/s) counter for tcptotbdgcollisions
tcperrbdgmuted	<Double>	Read-only	The number of bridging related interface mutes
tcperrbdgmutedrate	<Double>	Read-only	Rate (/s) counter for tcperrbdgmuted
totbdgpkts	<Double>	Read-only	The total number of bridged packets
bdgpktsrate	<Double>	Read-only	Rate (/s) counter for totbdgpkts
totbdgmbits	<Double>	Read-only	The total number of bridged Mbits
bdgmbitsrate	<Double>	Read-only	Rate (/s) counter for totbdgmbits

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/bridge

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/bridge?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get bridge resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "bridge": [ {
    "bdgpktsrate":<Double_value>,
    "totbdgmbits":<Double_value>,
    "bdgmbitsrate":<Double_value>,
    "totbdgpkts":<Double_value>,
    "tcperrbdgmuted":<Double_value>,
    "tcpbdgcollisionsrate":<Double_value>,
    "tcptotbdgcollisions":<Double_value>,
    "tcperrbdgmutedrate":<Double_value>,
    "tcpbdgmacmovedrate":<Double_value>,
    "tcptotbdgmacmoved":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

inat

May 20, 2016

Statistics for inbound nat resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The INAT. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
nat46tottcp46	<Double>	Read-only	Total TCP packets translated (V4->v6).
nat46tcp46rate	<Double>	Read-only	Rate (/s) counter for nat46tottcp46
nat46totudp46	<Double>	Read-only	Total UDP packets translated (V4->v6).
nat46udp46rate	<Double>	Read-only	Rate (/s) counter for nat46totudp46
nat46toticmp46	<Double>	Read-only	Total ICMP packets translated (V4->v6).
nat46icmp46rate	<Double>	Read-only	Rate (/s) counter for nat46toticmp46
nat46totdrop46	<Double>	Read-only	Total IPV4 packets dropped.
nat46drop46rate	<Double>	Read-only	Rate (/s) counter for nat46totdrop46
nat46tottcp64	<Double>	Read-only	Total TCP packets translated (V6->v4).
nat46tcp64rate	<Double>	Read-only	Rate (/s) counter for nat46tottcp64
nat46totudp64	<Double>	Read-only	Total UDP packets translated (V6->v4).
nat46udp64rate	<Double>	Read-only	Rate (/s) counter for nat46totudp64
nat46toticmp64	<Double>	Read-only	Total ICMP packets translated (V6->v4).
nat46icmp64rate	<Double>	Read-only	Rate (/s) counter for nat46toticmp64
nat46totdrop64	<Double>	Read-only	Total IPV6 packets dropped.
nat46drop64rate	<Double>	Read-only	Rate (/s) counter for nat46totdrop64
inatnat46tcp46	<Double>	Read-only	TCP packets translated (V4->v6).
inatnat46tcp46rate	<Double>	Read-only	Rate (/s) counter for inatnat46tcp46
inatnat46udp46	<Double>	Read-only	UDP packets translated (V4->v6).
inatnat46udp46rate	<Double>	Read-only	Rate (/s) counter for inatnat46udp46
inatnat46icmp46	<Double>	Read-only	ICMP packets translated (V4->v6).
inatnat46icmp46rate	<Double>	Read-only	Rate (/s) counter for inatnat46icmp46
inatnat46drop46	<Double>	Read-only	IPV4 packets dropped.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
inatnat46tcp64	<Double>	Read-only	TCP packets translated (V6->v4).
inatnat46tcp64rate	<Double>	Read-only	Rate (/s) counter for inatnat46tcp64
inatnat46udp64	<Double>	Read-only	UDP packets translated (V6->v4).
inatnat46udp64rate	<Double>	Read-only	Rate (/s) counter for inatnat46udp64
inatnat46icmp64	<Double>	Read-only	ICMP packets translated (V6->v4).
inatnat46icmp64rate	<Double>	Read-only	Rate (/s) counter for inatnat46icmp64
inatnat46drop64	<Double>	Read-only	IPV6 packets dropped.
inatnat46drop64rate	<Double>	Read-only	Rate (/s) counter for inatnat46drop64

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/inat

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/inat?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get inat resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "inat": [ {
    "name":<String_value>,
    "nat46udp64rate":<Double_value>,
    "nat46totdrop64":<Double_value>,
    "nat46totudp46":<Double_value>,
    "nat46icmp64rate":<Double_value>,
    "inatnat46tcp46":<Double_value>,
    "nat46totdrop46":<Double_value>,
    "inatnat46tcp64":<Double_value>,
    "inatnat46drop46":<Double_value>,
    "inatnat46drop64":<Double_value>,
    "inatnat46udp46":<Double_value>,
    "nat46tottcp64":<Double_value>,
    "nat46drop46rate":<Double_value>,
    "inatnat46drop46rate":<Double_value>,
    "inatnat46tcp64rate":<Double_value>,
    "inatnat46icmp64":<Double_value>,
    "nat46drop64rate":<Double_value>,
    "inatnat46tcp46rate":<Double_value>,
    "nat46tottcp46":<Double_value>,
    "nat46toticmp46":<Double_value>,
    "inatnat46icmp46rate":<Double_value>,
    "inatnat46udp64":<Double_value>,
    "nat46tcp46rate":<Double_value>,
    "inatnat46drop64rate":<Double_value>,
    "inatnat46udp64rate":<Double_value>,
    "nat46icmp46rate":<Double_value>,
    "inatnat46icmp46":<Double_value>,
    "inatnat46icmp64rate":<Double_value>,
    "nat46tcp64rate":<Double_value>,
    "nat46totudp64":<Double_value>,
    "nat46udp46rate":<Double_value>,
    "nat46toticmp64":<Double_value>,
    "inatnat46udp46rate":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/inat/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "inat": [ {
    "name":<String_value>,
    "nat46udp64rate":<Double_value>,
    "nat46totdrop64":<Double_value>,
    "nat46totudp46":<Double_value>,
    "nat46icmp64rate":<Double_value>,
    "inatnat46tcp46":<Double_value>,
    "nat46totdrop46":<Double_value>,
    "inatnat46tcp64":<Double_value>,
    "inatnat46drop46":<Double_value>,
    "inatnat46drop64":<Double_value>,
    "inatnat46udp46":<Double_value>,
    "nat46tottcp64":<Double_value>,
    "nat46drop46rate":<Double_value>,
    "inatnat46drop46rate":<Double_value>,
    "inatnat46tcp64rate":<Double_value>,
    "inatnat46icmp64":<Double_value>,
    "nat46drop64rate":<Double_value>,
    "inatnat46tcp46rate":<Double_value>,
    "nat46tottcp46":<Double_value>,
    "nat46toticmp46":<Double_value>,
    "inatnat46icmp46rate":<Double_value>,
    "inatnat46udp64":<Double_value>,
    "nat46tcp46rate":<Double_value>,
    "inatnat46drop64rate":<Double_value>,
    "inatnat46udp64rate":<Double_value>,
    "nat46icmp46rate":<Double_value>,
    "inatnat46icmp46":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "inatnat46icmp64rate":<Double_value>,
    "nat46tcp64rate":<Double_value>,
    "nat46totudp64":<Double_value>,
    "nat46udp46rate":<Double_value>,
    "nat46totoicmp64":<Double_value>,
    "inatnat46udp46rate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

inatsession

May 20, 2016

Statistics for stateful INAT resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	INAT name. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
inattothits	<Double>	Read-only	INAT total sessions
inathitsrate	<Double>	Read-only	Rate (/s) counter for inattothits
inatcursessions	<Double>	Read-only	INAT current sessions
inatcursessionsrate	<Double>	Read-only	Rate (/s) counter for inatcursessions
inattotreceivebytes	<Double>	Read-only	INAT total Received Bytes
inatreceivebytesrate	<Double>	Read-only	Rate (/s) counter for inattotreceivebytes
inattotsentbytes	<Double>	Read-only	INAT total Sent Bytes
inatsentbytesrate	<Double>	Read-only	Rate (/s) counter for inattotsentbytes
inattotpktreceived	<Double>	Read-only	INAT total Packets Received
inatpktreceivedrate	<Double>	Read-only	Rate (/s) counter for inattotpktreceived
inattotpktsent	<Double>	Read-only	INAT total Packets Sent
inatpktsentrate	<Double>	Read-only	Rate (/s) counter for inattotpktsent

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/inatsession

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/inatsession?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get inatsession resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "inatsession": [ {
      "name":<String_value>,
      "inattotsentbytes":<Double_value>,
      "inatreceivebytesrate":<Double_value>,
      "inathitsrate":<Double_value>,
      "inattotreceivebytes":<Double_value>,
      "inatcursessionsrate":<Double_value>,
      "inatcursessions":<Double_value>,
      "inatsentbytesrate":<Double_value>,
      "inatpktsentrate":<Double_value>,
      "inattotpktreceived":<Double_value>,
      "inattothits":<Double_value>,
      "inattotpktsent":<Double_value>,
      "inatpktreceivedrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/inatsession/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "inatsession": [ {
    "name":<String_value>,
    "inattotsentbytes":<Double_value>,
    "inatreceivebytesrate":<Double_value>,
    "inathitsrate":<Double_value>,
    "inattotreceivebytes":<Double_value>,
    "inatcursessionsrate":<Double_value>,
    "inatcursessions":<Double_value>,
    "inatsentbytesrate":<Double_value>,
    "inatpktsentrate":<Double_value>,
    "inattotpktreceived":<Double_value>,
    "inattothits":<Double_value>,
    "inattotpktsent":<Double_value>,
    "inatpktreceivedrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Interface

May 20, 2016

Statistics for interface resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<String>	Read-write	Interface number, in C/U format, where C can take one of the following values: * 0 - Indicates a management interface. * 1 - Indicates a 1 Gbps port. * 10 - Indicates a 10 Gbps port. * LA - Indicates a link aggregation port. * LO - Indicates a loop back port. U is a unique integer for representing an interface in a particular port group.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
curintfstate	<String>	Read-only	Current state of the specified interface. The interface state set to UP only if the link state is UP and administrative state is ENABLED .
curlinkuptime	<String>	Read-only	Duration for which the link is UP. This statistic is reset when the state changes to DOWN.
curlinkdowntime	<String>	Read-only	Duration for which the link is DOWN. This statistic is reset when the state changes to UP.
totrxbytes	<Double>	Read-only	Number of bytes received by an interface since the NetScaler appliance was started or the interface statistics were cleared.
rxbytesrate	<Double>	Read-only	Rate (/s) counter for totrxbytes
tottxbytes	<Double>	Read-only	Number of bytes transmitted by an interface since the NetScaler appliance was started or the interface statistics were cleared.
txbytesrate	<Double>	Read-only	Rate (/s) counter for tottxbytes
totrxpkts	<Double>	Read-only	Number of packets received by an interface since the NetScaler appliance was started or the interface statistics were cleared.
rxpktsrate	<Double>	Read-only	Rate (/s) counter for totrxpkts
tottxpkts	<Double>	Read-only	Number of packets transmitted by an interface since the NetScaler appliance was started or the interface statistics were cleared.
txpktsrate	<Double>	Read-only	Rate (/s) counter for tottxpkts
jumbopktsreceived	<Double>	Read-only	Number of Jumbo Packets received on this interface.
jumbopktsreceivedrate	<Double>	Read-only	Rate (/s) counter for jumbopktsreceived
jumbopktstransmitted	<Double>	Read-only	Number of Jumbo packets transmitted on this interface by upper layer, with TSO enabled actual trasmission size could be non Jumbo.
jumbopktstransmittedrate	<Double>	Read-only	Rate (/s) counter for jumbopktstransmitted
nictotmulticastpkts	<Double>	Read-only	Number of multicast packets received by the specified interface since the NetScaler appliance was started or the interface statistics were cleared.
nicmulticastpktsrate	<Double>	Read-only	Rate (/s) counter for nictotmulticastpkts
totnetscalerpkts	<Double>	Read-only	Number of packets, destined to the NetScaler, received by an interface since the NetScaler appliance was started or the interface statistics were cleared. The packets destined to NetScaler are those that have the same MAC address as that of an interface or a VMAC address owned by the NetScaler.
netscalerpktsrate	<Double>	Read-only	Rate (/s) counter for totnetscalerpkts
rxlacpdu	<Double>	Read-only	Number of Link Aggregation Control Protocol Data Units(LACPDUs) received by the specified interface since the NetScaler appliance was started or the interface statistics were cleared.
rxlacpdurate	<Double>	Read-only	Rate (/s) counter for rxlacpdu

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
txlacpdu	<Double>	Read-only	interface statistics were cleared.
txlacpdurate	<Double>	Read-only	Rate (/s) counter for txlacpdu
errpktrx	<Double>	Read-only	Number of inbound packets dropped by the hardware on a specified interface once the NetScaler appliance starts or the interface statistics are cleared. This happens due to following reasons: 1) The hardware receives packets at a rate higher rate than that at which the software is processing packets. In this case, the hardware FIFO overruns and starts dropping the packets . 2) The specified interface fails to receive inbound packets from the appliance because of insufficient memory. 3) The specified interface receives packets with CRC errors (Alignment or Frame Check Sequence). 4) The specified interface receives overly long packets. 5) The specified interface receives packets with alignment errors. 6) The software does less buffering because it is running out of available memory. When hardware detects that there is no space into which to push newly arrived packets, it starts dropping them. 7) The specified interface receives packets with Frame Check Sequence (FCS) errors. 8) The specified interface receives packets smaller than 64 bytes. 9) The specified interface discards error-free inbound packets because of insufficient resources. For example: NIC buffers. 10) Packets are missed because of collision detection, link lost, physical decoding error, or MAC abort.
errpktrxrate	<Double>	Read-only	Rate (/s) counter for errpktrx
errpkttx	<Double>	Read-only	Number of outbound packets dropped by the hardware on a specified interface since the NetScaler appliance was started or the interface statistics were cleared. This could happen due to length (undersize or oversize) errors and lack of resources. This statistic is available only for: (1) Loop back interface (LO) of all platforms. (2) All data ports on the NetScaler 12000 platform. (3) Management ports on the MPX 15000 and 17000 platforms.
errpkttxrate	<Double>	Read-only	Rate (/s) counter for errpkttx
errifindiscards	<Double>	Read-only	Number of error-free inbound packets discarded by the specified interface due to a lack of resources, for example, insufficient receive buffers.
errifindiscardsrate	<Double>	Read-only	Rate (/s) counter for errifindiscards
nicerrifoutdiscards	<Double>	Read-only	Number of error-free outbound packets discarded by the specified interface due to a lack of resources. This statistic is not available on: (1) 10G ports of NetScaler MPX 12500/12500/15500-10G platforms. (2) 10G data ports on NetScaler MPX 17500/19500/21500 platforms.
nicerrifoutdiscardsrate	<Double>	Read-only	Rate (/s) counter for nicerrifoutdiscards
errdroppedrxpkts	<Double>	Read-only	Number of inbound packets dropped by the specified interface. Commonly dropped packets are multicast frames, spanning tree BPDUs, packets destined to a MAC not owned by the NetScaler when L2 mode is disabled, or packets tagged for a VLAN that is not bound to the interface. This statistic will increment in most healthy networks at a steady rate regardless of traffic load. If a sharp spike in dropped packets occurs, it generally indicates an issue with connected L2 switches, such as a forwarding database overflow resulting in packets being broadcast on all ports.
errdroppedrxpktsrate	<Double>	Read-only	Rate (/s) counter for errdroppedrxpkts
errdroppedtxpkts	<Double>	Read-only	Number of packets dropped in transmission by the specified interface due to one of the following reasons. (1) VLAN mismatch. (2) Oversized packets. (3) Interface congestion. (4) Loopback packets sent on non loop back interface.
errdroppedtxpktsrate	<Double>	Read-only	Rate (/s) counter for errdroppedtxpkts
errlinkhangs	<Double>	Read-only	Number of times the specified interface detected hangs in the transmit and receive paths since the NetScaler appliance was started or the interface statistics were cleared.
nicstsstalls	<Double>	Read-only	Number of times the status updates for a specified interface were stalled since the NetScaler appliance was started or the interface statistics were cleared. A status stall is detected when the status of the interface is not updated by the NIC hardware within 0.8 seconds of the last update.
nictxstalls	<Double>	Read-only	Number of times the interface stalled, when transmitting packets, since the NetScaler appliance was started or the interface statistics were cleared. Transmit (Tx) stalls are detected when a packet posted for transmission is not transmitted in 4 seconds.
nicrxstalls	<Double>	Read-only	Number of times the interface stalled, when receiving packets, since the NetScaler appliance was started or the interface statistics were cleared. Receive (Rx) stalls are detected when the following conditions are met: (1)The link is up for more than 10 minutes. (2)Packets are transmitted, but no packets is received for 16 seconds.
nicerrdisables	<Double>	Read-only	Number of times the specified interface is disabled by the NetScaler, due to continuous Receive (Rx) or Transmit (Tx) stalls, since the NetScaler appliance was started or the interface statistics were cleared. The NetScaler disables an interface when one of the following conditions is met: (1) Three consecutive transmit stalls occurs with at most gap of 10 seconds between any two stalls. (2) Three consecutive receive stalls occurs with at most gap of 120 seconds between any two stalls.
errduplexmismatch	<Double>	Read-only	Number of times duplex mismatches were detected on the specified interface since the NetScaler appliance was started or the interface statistics were cleared. A mismatch will occur if the duplex mode is not identically set on both ends of the link. This statistic is only available on the NetScaler Classic edition.
linkreinits	<Double>	Read-only	Number of times the link has been re-initialized. A re-initialization occurs due to link state change, configuration parameter change, or administrative reset operation.
totmacmoved	<Double>	Read-only	Number of MAC moves between ports. If a high rate of MAC moves is observed, it is likely that there is a bridge loop between two interfaces.
macmovedrate	<Double>	Read-only	Rate (/s) counter for totmacmoved
errnicmuted	<Double>	Read-only	Number of times the specified interface stopped transmitting and receiving packets due to MAC moves between ports.
interfacealias	<String>	Read-only	Alias Name for the Interface

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			link to be UP at least one of the slave interfaces needs to be UP.

Operations (click to see Properties)

Name	Data Type	Permissions	Description

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/Interface

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/Interface?**args=id:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get Interface resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "Interface": [ {
    "id":<String_value>,
    "errpkttx":<Double_value>,
    "nicrxstalls":<Double_value>,
    "nictotmulticastpkts":<Double_value>,
    "jumbopktstransmitted":<Double_value>,
    "txbytesrate":<Double_value>,
    "rxlacpdu":<Double_value>,
    "netscalerpktsrate":<Double_value>,
    "tottxpkts":<Double_value>,
    "macmovedrate":<Double_value>,
    "curintfstate":<String_value>,
    "errpktrx":<Double_value>,
    "curlinkdowntime":<String_value>,
    "nicmulticastpktsrate":<Double_value>,
    "rxbytesrate":<Double_value>,
    "curlinkuptime":<String_value>,
    "tottxbytes":<Double_value>,
    "errdroppedtxpktsrate":<Double_value>,
    "errpkttxrate":<Double_value>,
    "nictxstalls":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "rxlacpdurate":<Double_value>,
    "totnetscalerpkts":<Double_value>,
    "jumbopktsreceivedrate":<Double_value>,
    "totrxbytes":<Double_value>,
    "linkreinits":<Double_value>,
    "nicerrdisables":<Double_value>,
    "txpktsrate":<Double_value>,
    "jumbopktsreceived":<Double_value>,
    "txlacpdu":<Double_value>,
    "errdroppedrxpktsrate":<Double_value>,
    "errnicmuted":<Double_value>,
    "errdroppedtxpkts":<Double_value>,
    "nicerrifoutdiscardsrate":<Double_value>,
    "errifindiscardsrate":<Double_value>,
    "nicerrifoutdiscards":<Double_value>,
    "curlinkstate":<String_value>,
    "errifindiscards":<Double_value>,
    "interfacealias":<String_value>,
    "totrxpkts":<Double_value>,
    "totmacmoved":<Double_value>,
    "rxpktsrate":<Double_value>,
    "errduplexmismatch":<Double_value>,
    "jumbopktstransmittedrate":<Double_value>,
    "errpktrxrate":<Double_value>,
    "errlinkhangs":<Double_value>,
    "errdroppedrxpkts":<Double_value>,
    "txlacpdurate":<Double_value>,
    "nicstsstalls":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/Interface/id_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "Interface": [ {
    "id":<String_value>,
    "errpkttx":<Double_value>,
    "nicrxstalls":<Double_value>,
    "nictotmulticastpkts":<Double_value>,
    "jumbopktstransmitted":<Double_value>,
    "txbytesrate":<Double_value>,
    "rxlacpdu":<Double_value>,
    "netscalerpktsrate":<Double_value>,
    "tottxpkts":<Double_value>,
    "macmovedrate":<Double_value>,
    "curintfstate":<String_value>,
    "errpktrx":<Double_value>,
    "curlinkdowntime":<String_value>,
    "nicmulticastpktsrate":<Double_value>,
    "rxbytesrate":<Double_value>,
    "curlinkuptime":<String_value>,
    "tottxbytes":<Double_value>,
    "errdroppedtxpktsrate":<Double_value>,
    "errpkttxrate":<Double_value>,
    "nictxstalls":<Double_value>,
    "rxlacpdurate":<Double_value>,
    "totnetscalerpkts":<Double_value>,
    "jumbopktsreceivedrate":<Double_value>,
    "totrxbytes":<Double_value>,
    "linkreinits":<Double_value>,
    "nicerrdisables":<Double_value>,
    "txpktsrate":<Double_value>,
    "jumbopktsreceived":<Double_value>,
    "txlacpdu":<Double_value>,
    "errdroppedrxpktsrate":<Double_value>,
    "errnicmuted":<Double_value>,
    "errdroppedtxpkts":<Double_value>,
    "nicerrifoutdiscardsrate":<Double_value>,
    "errifindiscardsrate":<Double_value>,
    "nicerrifoutdiscards":<Double_value>,
    "curlinkstate":<String_value>,
    "errifindiscards":<Double_value>,
    "interfacealias":<String_value>,
    "totrxpkts":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "totmacmoved":<Double_value>,
    "rxpktsrate":<Double_value>,
    "errduplexmismatch":<Double_value>,
    "jumbopktstransmittedrate":<Double_value>,
    "errpktrxrate":<Double_value>,
    "errlinkhangs":<Double_value>,
    "errdroppedrxpkts":<Double_value>,
    "txlacpdurate":<Double_value>,
    "nicstsstalls":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nat64

Jul 07, 2017

Statistics for nat64 config resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
nat64tottcpsessions	<Double>	Read-only	Total number of TCP sessions created by NAT64.
nat64tcpsessionsrate	<Double>	Read-only	Rate (/s) counter for nat64tottcpsessions
nat64totudpsessions	<Double>	Read-only	Total number of UDP sessions created by NAT64.
nat64udpsessionsrate	<Double>	Read-only	Rate (/s) counter for nat64totudpsessions
nat64toticmpsessions	<Double>	Read-only	Total number of ICMP sessions created by NAT64.
nat64icmpsessionsrate	<Double>	Read-only	Rate (/s) counter for nat64toticmpsessions
nat64totsessions	<Double>	Read-only	Total number of sessions created by NAT64.
nat64sessionsrate	<Double>	Read-only	Rate (/s) counter for nat64totsessions

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nat64

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nat64?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get nat64 resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nat64": [ {
    "nat64icmpsessionsrate":<Double_value>,
    "nat64toticmpsessions":<Double_value>,
    "nat64sessionsrate":<Double_value>,
    "nat64totudpsessions":<Double_value>,
    "nat64udpsessionsrate":<Double_value>,
    "nat64tottcpsessions":<Double_value>,
    "nat64totsessions":<Double_value>,
    "nat64tcpsessionsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rnat

Jul 07, 2017

Statistics for RNAT configured route resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
rnattotrxbytes	<Double>	Read-only	Bytes received during RNAT sessions.
rnatrxbytesrate	<Double>	Read-only	Rate (/s) counter for rnattotrxbytes
rnattottxbytes	<Double>	Read-only	Bytes sent during RNAT sessions.
rnattxbytesrate	<Double>	Read-only	Rate (/s) counter for rnattottxbytes
rnattotrxpkts	<Double>	Read-only	Packets received during RNAT sessions.
rnatrxpktsrate	<Double>	Read-only	Rate (/s) counter for rnattotrxpkts
rnattottxpkts	<Double>	Read-only	Packets sent during RNAT sessions.
rnattxpktsrate	<Double>	Read-only	Rate (/s) counter for rnattottxpkts
rnattottxsyn	<Double>	Read-only	Requests for connections sent during RNAT sessions.
rnattxsynrate	<Double>	Read-only	Rate (/s) counter for rnattottxsyn
rnatcursessions	<Double>	Read-only	Currently active RNAT sessions.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/rnat

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/rnat?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get rnat resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rnat": [ {
    "rnattottxpkts":<Double_value>,
    "rnatrxbytesrate":<Double_value>,
    "rnattxsynrate":<Double_value>,
    "rnattxpktsrate":<Double_value>,
    "rnattottxsyn":<Double_value>,
    "rnattxbytesrate":<Double_value>,
    "rnatrxpktsrate":<Double_value>,
    "rnatcursessions":<Double_value>,
    "rnattotrxpkts":<Double_value>,
    "rnattotrxbytes":<Double_value>,
    "rnattottxbytes":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

rnatip

May 20, 2016

Statistics for RNAT ipaddress resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
Rnatip	<String>	Read-write	Specifies the NAT IP address of the configured RNAT entry for which you want to see the statistics. If you do not specify an IP address, this displays the statistics for all the configured RNAT entries. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
iptd	<Double>	Read-only	Traffic domain for ipaddr.
iprnattotrxbytes	<Double>	Read-only	Bytes received on this IP address during RNAT sessions.
iprnatrxbytesrate	<Double>	Read-only	Rate (/s) counter for iprnattotrxbytes
iprnattottxbytes	<Double>	Read-only	Bytes sent from this IP address during RNAT sessions.
iprnattxbytesrate	<Double>	Read-only	Rate (/s) counter for iprnattottxbytes
iprnattotrxpkts	<Double>	Read-only	Packets received on this IP address during RNAT sessions.
iprnatrxpktsrate	<Double>	Read-only	Rate (/s) counter for iprnattotrxpkts
iprnattottxpkts	<Double>	Read-only	Packets sent from this IP address during RNAT sessions.
iprnattxpktsrate	<Double>	Read-only	Rate (/s) counter for iprnattottxpkts
iprnattottxsyn	<Double>	Read-only	Requests for connections sent from this IP address during RNAT sessions.
iprnattxsynrate	<Double>	Read-only	Rate (/s) counter for iprnattottxsyn
iprnatcursessions	<Double>	Read-only	Currently active RNAT sessions started from this IP address.

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/rnatip

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/rnatip?**args=Rnatip:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_valu**

Use this query-parameter to get rnatip resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rnatip": [ {
    "Rnatip":<String_value>,
    "iprnatrxpktsrate":<Double_value>,
    "iprnattxpktsrate":<Double_value>,
    "iprnattottxpkts":<Double_value>,
    "iptd":<Double_value>,
    "iprnattottxbytes":<Double_value>,
    "iprnatcursessions":<Double_value>,
    "iprnatrxbytesrate":<Double_value>,
    "iprnattotrxbytes":<Double_value>,
    "iprnattxsynrate":<Double_value>,
    "iprnattxbytesrate":<Double_value>,
    "iprnattotrxpkts":<Double_value>,
    "iprnattottxsyn":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/rnatip/Rnatip_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rnatip": [ {
    "Rnatip":<String_value>,
    "iprnatrxpktsrate":<Double_value>,
    "iprnattxpktsrate":<Double_value>,
    "iprnattottxpkts":<Double_value>,
    "iptd":<Double_value>,
    "iprnattottxbytes":<Double_value>,
    "iprnatcursessions":<Double_value>,
    "iprnatrxbytesrate":<Double_value>,
    "iprnattotrxbytes":<Double_value>,
    "iprnattxsynrate":<Double_value>,
    "iprnattxbytesrate":<Double_value>,
    "iprnattotrxpkts":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "iprnattottxsyn":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

tunnelip

May 20, 2016

Statistics for TUNNEL Remote ipaddress resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
tunnelip	<String>	Read-write	remote IP address of the configured tunnel. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
tnltotrxpkts	<Double>	Read-only	Total number of packets received on the tunnel.
tnlrxpktsrate	<Double>	Read-only	Rate (/s) counter for tnltotrxpkts
tnltottxpkts	<Double>	Read-only	Total number of packets transmitted on the tunnel.
tnltxpktsrate	<Double>	Read-only	Rate (/s) counter for tnltottxpkts
tnltotrxbytes	<Double>	Read-only	Total number of bytes received on the tunnel.
tnlrxbytesrate	<Double>	Read-only	Rate (/s) counter for tnltotrxbytes
tnltottxbytes	<Double>	Read-only	Total number of bytes transmitted on the tunnel.
tnltxbytesrate	<Double>	Read-only	Rate (/s) counter for tnltottxbytes

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/tunnelip

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/tunnelip?**args=tunnelip:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_v**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get tunnelip resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tunnelip": [ {
    "tunnelip":<String_value>,
    "tnltxpktsrate":<Double_value>,
    "tnltxbytesrate":<Double_value>,
    "tnlrxpktsrate":<Double_value>,
    "tnltotrxpkts":<Double_value>,
    "tnltottxbytes":<Double_value>,
    "tnltottxpkts":<Double_value>,
    "tnlrxbytesrate":<Double_value>,
    "tnltotrxbytes":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/tunnelip/tunnelip_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tunnelip": [ {
    "tunnelip":<String_value>,
    "tnltxpktsrate":<Double_value>,
    "tnltxbytesrate":<Double_value>,
    "tnlrxpktsrate":<Double_value>,
    "tnltotrxpkts":<Double_value>,
    "tnltottxbytes":<Double_value>,
    "tnltottxpkts":<Double_value>,
    "tnlrxbytesrate":<Double_value>,
    "tnltotrxbytes":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tunnelip6

May 20, 2016

Statistics for TUNNEL Remote ip6address resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
tunnelip6	<String>	Read-write	remote IPv6 address of the configured tunnel. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
tnltotrxpkts	<Double>	Read-only	Total number of packets received on the tunnel.
tnlrxpktsrate	<Double>	Read-only	Rate (/s) counter for tnltotrxpkts
tnltottxpkts	<Double>	Read-only	Total number of packets transmitted on the tunnel.
tnltxpktsrate	<Double>	Read-only	Rate (/s) counter for tnltottxpkts
tnltotrxbytes	<Double>	Read-only	Total number of bytes received on the tunnel.
tnlrxbytesrate	<Double>	Read-only	Rate (/s) counter for tnltotrxbytes
tnltottxbytes	<Double>	Read-only	Total number of bytes transmitted on the tunnel.
tnltxbytesrate	<Double>	Read-only	Rate (/s) counter for tnltottxbytes

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/tunnelip6

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/tunnelip6?**args=tunnelip6:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String**

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Use this query-parameter to get tunnelip6 resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tunnelip6": [ {
    "tunnelip6":<String_value>,
    "tnltxbytesrate":<Double_value>,
    "tnltotrxpkts":<Double_value>,
    "tnltottxbytes":<Double_value>,
    "tnlrxbytesrate":<Double_value>,
    "tnltotrxbytes":<Double_value>,
    "tnltxpktsrate":<Double_value>,
    "tnlrxpktsrate":<Double_value>,
    "tnltottxpkts":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/tunnelip6/tunnelip6_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tunnelip6": [ {
    "tunnelip6":<String_value>,
    "tnltxbytesrate":<Double_value>,
    "tnltotrxpkts":<Double_value>,
    "tnltottxbytes":<Double_value>,
    "tnlrxbytesrate":<Double_value>,
    "tnltotrxbytes":<Double_value>,
    "tnltxpktsrate":<Double_value>,
    "tnlrxpktsrate":<Double_value>,
    "tnltottxpkts":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vlan

May 20, 2016
Statistics for "VLAN" resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	An integer specifying the VLAN identification number (VID). Possible values: 1 through 4094. Minimum value = 1 Maximum value = 4094
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
vlantotrxpkts	<Double>	Read-only	Packets received on the VLAN.
vlanrxpktsrate	<Double>	Read-only	Rate (/s) counter for vlantotrxpkts
vlantotrxbytes	<Double>	Read-only	Bytes of data received on the VLAN.
vlanrxbytesrate	<Double>	Read-only	Rate (/s) counter for vlantotrxbytes
vlantottxpkts	<Double>	Read-only	Packets transmitted on the VLAN.
vlantxpktsrate	<Double>	Read-only	Rate (/s) counter for vlantottxpkts
vlantottxbytes	<Double>	Read-only	Bytes of data transmitted on the VLAN.
vlantxbytesrate	<Double>	Read-only	Rate (/s) counter for vlantottxbytes
vlantotdroppedpkts	<Double>	Read-only	Inbound packets dropped by the VLAN upon reception.
vlantotbroadcastpkts	<Double>	Read-only	Broadcast packets sent and received on the VLAN.

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/stat/vlan

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/vlan?**args=id:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get vlan resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vlan": [ {
    "id":<Double_value>,
    "vlanrxbytesrate":<Double_value>,
    "vlantottxpkts":<Double_value>,
    "vlantotdroppedpkts":<Double_value>,
    "vlanrxpktsrate":<Double_value>,
    "vlantotrxpkts":<Double_value>,
    "vlantotbroadcastpkts":<Double_value>,
    "vlantxbytesrate":<Double_value>,
    "vlantotrxbytes":<Double_value>,
    "vlantxpktsrate":<Double_value>,
    "vlantottxbytes":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/vlan/id_value><Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vlan": [ {
    "id":<Double_value>,
    "vlanrxbytesrate":<Double_value>,
    "vlantottxpkts":<Double_value>,
    "vlantotdroppedpkts":<Double_value>,
    "vlanrxpktsrate":<Double_value>,
    "vlantotrxpkts":<Double_value>,
    "vlantotbroadcastpkts":<Double_value>,
    "vlantxbytesrate":<Double_value>,
    "vlantotrxbytes":<Double_value>,
    "vlantxpktsrate":<Double_value>,
    "vlantottxbytes":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpath

May 20, 2016

Statistics for Vpath resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
vpathtotl2datarx	<Double>	Read-only	Total number of non-fragmented vPath data packets decapsulated in L2 adjacency
vpathl2datarxrate	<Double>	Read-only	Rate (/s) counter for vpathtotl2datarx
vpathtotl3datarx	<Double>	Read-only	Total number of non-fragmented vPath data packets decapsulated in L3 adjacency
vpathl3datarxrate	<Double>	Read-only	Rate (/s) counter for vpathtotl3datarx
vpathtotl2cntrlpkts	<Double>	Read-only	Total number of vPath control packets received in L2 adjacency
vpathl2cntrlpktsrate	<Double>	Read-only	Rate (/s) counter for vpathtotl2cntrlpkts
vpathtotl3cntrlpkts	<Double>	Read-only	Total number of vPath control packets received in L3 adjacency
vpathl3cntrlpktsrate	<Double>	Read-only	Rate (/s) counter for vpathtotl3cntrlpkts
vpathtotfragpkts	<Double>	Read-only	Total number of vPath fragments received
vpathfragpktsrate	<Double>	Read-only	Rate (/s) counter for vpathtotfragpkts
vpathtotl2encappkts	<Double>	Read-only	Total number of L2 vPath encapsulated packets injected to VEM
vpathl2encappktsrate	<Double>	Read-only	Rate (/s) counter for vpathtotl2encappkts
vpathtotl3encappkts	<Double>	Read-only	Total number of L3 vPath encapsulated packets injected to VEM
vpathl3encappktsrate	<Double>	Read-only	Rate (/s) counter for vpathtotl3encappkts
vpathtotfragencappkts	<Double>	Read-only	Number of fragmented vPath packets transmitted
vpathfragencappktsrate	<Double>	Read-only	Rate (/s) counter for vpathtotfragencappkts
vpathtotoffload	<Double>	Read-only	Number of offloaded vPath packets transmitted
vpathoffloadrate	<Double>	Read-only	Rate (/s) counter for vpathtotoffload

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/vpath

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/vpath?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get vpath resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpath": [ {
    "vpathfragencappktsrate":<Double_value>,
    "vpathtotl2cntrlpkts":<Double_value>,
    "vpathtotoffload":<Double_value>,
    "vpathtotl3encappkts":<Double_value>,
    "vpathtotl2datarx":<Double_value>,
    "vpathtotl2encappkts":<Double_value>,
    "vpathoffloadrate":<Double_value>,
    "vpathl3cntrlpktsrate":<Double_value>,
    "vpathfragpktsrate":<Double_value>,
    "vpathtotl3datarx":<Double_value>,
    "vpathtotl3cntrlpkts":<Double_value>,
    "vpathtotfragencappkts":<Double_value>,
    "vpathl3encappktsrate":<Double_value>,
    "vpathl2encappktsrate":<Double_value>,
    "vpathl2cntrlpktsrate":<Double_value>,
    "vpathl3datarxrate":<Double_value>,
    "vpathl2datarxrate":<Double_value>,
    "vpathtotfragpkts":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vxlan

May 20, 2016
Statistics for "VXLAN" resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	An integer specifying the VXLAN identification number (VNID). Minimum value = 1 Maximum value = 16777215
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
vxlantotrxpkts	<Double>	Read-only	Packets received on the VXLAN.
vxlanrxpktsrate	<Double>	Read-only	Rate (/s) counter for vxlantotrxpkts
vxlantotrxbytes	<Double>	Read-only	Bytes of data received on the VXLAN.
vxlanrxbytesrate	<Double>	Read-only	Rate (/s) counter for vxlantotrxbytes
vxlantottxpkts	<Double>	Read-only	Packets transmitted on the VXLAN.
vxlantxpktsrate	<Double>	Read-only	Rate (/s) counter for vxlantottxpkts
vxlantottxbytes	<Double>	Read-only	Bytes of data transmitted on the VXLAN.
vxlantxbytesrate	<Double>	Read-only	Rate (/s) counter for vxlantottxbytes

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/vxlan

Query-parameters:

args

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://<netscaler-ip-address>/nitro/v1/stat/vxlan?**args=id:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get vxlan resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vxlan": [ {
    "id":<Double_value>,
    "vxlanrxbytesrate":<Double_value>,
    "vxlantottxpkts":<Double_value>,
    "vxlanrxpktsrate":<Double_value>,
    "vxlantotrxbytes":<Double_value>,
    "vxlantxbytesrate":<Double_value>,
    "vxlantxpktsrate":<Double_value>,
    "vxlantottxbytes":<Double_value>,
    "vxlantotrxpkts":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/vxlan/id_value><Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vxlan": [ {
    "id":<Double_value>,
    "vxlanrxbytesrate":<Double_value>,
    "vxlantottxpkts":<Double_value>,
    "vxlanrxpktsrate":<Double_value>,
    "vxlantotrxbytes":<Double_value>,
    "vxlantxbytesrate":<Double_value>,
    "vxlantxpktsrate":<Double_value>,
    "vxlantottxbytes":<Double_value>,
    "vxlantotrxpkts":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NS

May 20, 2016

System/Global level statistics.

Name	Description
ns	Statistics for ns.
nsacl	Statistics for ACL entry resource.
nsacl6	Statistics for ACL6 entry resource.
nslimitidentifier	Statistics for limit Indetifier resource.
nsmemory	
nspartition	Statistics for admin partition resource.
nspbr	Statistics for Policy Based Routing(PBR) entry resource.
nspbr6	Statistics for PBR6 entry resource.
nssimpleacl	Statistics for simple ACL resource.
nssimpleacl6	Statistics for simple ACL6 resource.
nstrafficdomain	Statistics for Traffic Domain resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ns

Jul 07, 2017

Statistics for ns.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
rescpuusagepcnt	<Double>	Read-only	Average CPU utilization percentage. Not applicable for a single-CPU system.
cpuusagepcnt	<Double>	Read-only	CPU utilization percentage.
cachemaxmemorykb	<Double>	Read-only	Largest amount of memory the NetScaler can dedicate to caching, up to 50% of available memory. A 0 value disables caching, but the caching module continues to run.
delcmpratio	<Double>	Read-only	Ratio of compressible data received to compressed data transmitted.If this ratio is one (uncmp:1.0) that means compression is disabled or we are not able to compress even a single compressible packet.
rescpuusage	<Double>	Read-only	Shows average CPU utilization percentage if more than 1 CPU is present.
cpuusage	<Double>	Read-only	CPU utilization percentage.
resmemusage	<Double>	Read-only	Percentage of memory utilization on NetScaler.
comptotaldatacompressionratio	<Double>	Read-only	Ratio of total HTTP data received to total HTTP data transmitted.
compratio	<Double>	Read-only	Ratio of the compressible data received from the server to the compressed data sent to the client.
cacheutilizedmemorykb	<Double>	Read-only	Amount of memory the integrated cache is currently using.
cachemaxmemoryactivekb	<Double>	Read-only	Currently active value of maximum memory
cache64maxmemorykb	<Double>	Read-only	Largest amount of memory the NetScaler can dedicate to caching, up to 50% of available memory. A 0 value disables caching, but the caching module continues to run.
cachepercentoriginbandwidthsaved	<Double>	Read-only	Percentage of origin bandwidth saved, expressed as number of bytes served from the integrated cache divided by all bytes served. The assumption is that all compression is done in the NetScaler.
cachetotmisses	<Double>	Read-only	Intercepted HTTP requests requiring fetches from origin server.
cachemissesrate	<Double>	Read-only	Rate (/s) counter for cachetotmisses
cachetothits	<Double>	Read-only	Responses served from the integrated cache. These responses match a policy with a CACHE action.
cachehitsrate	<Double>	Read-only	Rate (/s) counter for cachetothits
sslnumcardsup	<Double>	Read-only	Number of SSL cards that are UP. If the number of cards UP is lower than a threshold, a failover is initiated.
memusagepcnt	<Double>	Read-only	Percentage of memory utilization on NetScaler.
memuseinmb	<Double>	Read-only	Main memory currently in use, in megabytes.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
mgmtcpuusagepcnt	<Double>	Read-only	Management CPU utilization percentage.
pktcpuusagepcnt	<Double>	Read-only	Average CPU utilization percentage for all packet engines excluding management PE.
starttime	<String>	Read-only	Time when the NetScaler appliance was last started.
transtime	<String>	Read-only	Time when the last master state transition occurred. You can use this statistic for debugging.
hacurstate	<String>	Read-only	State of the HA node, based on its health, in a high availability setup. Possible values are: UP ? Indicates that the node is accessible and can function as either a primary or secondary node. DISABLED ? Indicates that the high availability status of the node has been manually disabled. Synchronization and propagation cannot take place between the peer nodes. INIT ? Indicates that the node is in the process of becoming part of the high availability configuration. PARTIALFAIL ? Indicates that one of the high availability monitored interfaces has failed because of a card or link failure. This state triggers a failover. COMPLETEFAIL ? Indicates that all the interfaces of the node are unusable, because the interfaces on which high availability monitoring is enabled are not connected or are manually disabled. This state triggers a failover. DUMB ? Indicates that the node is in listening mode. It does not participate in high availability transitions or transfer configuration from the peer node. This is a configured value, not a statistic. PARTIALFAILSSL ? Indicates that the SSL card has failed. This state triggers a failover. ROUTEMONITORFAIL ? Indicates that the route monitor has failed. This state triggers a failover.
hacurmasterstate	<String>	Read-only	Indicates the high availability state of the node. Possible values are: STAYSECONDARY ? Indicates that the selected node remains the secondary node in a high availability setup. In this case a forced failover does not change the state but, instead, returns an appropriate error message. This is a configured value and not a statistic. PRIMARY ? Indicates that the selected node is the primary node in a high availability setup. SECONDARY ? Indicates that the selected node is the secondary node in a high availability setup. CLAIMING ? Indicates that the secondary node is in the process of taking over as the primary node. This is the intermediate state in the transition of the secondary node to primary status. FORCE CHANGE - Indicates that the secondary node is forcibly changing its status to primary due to a forced failover issued on the secondary node.
sslcards	<Double>	Read-only	Number of SSL crypto cards present on the NetScaler appliance.
disk0perusage	<Double>	Read-only	Used space in /flash partition of the disk, as a percentage. This is a critical counter. You can configure /flash Used (%) by using the Set snmp alarm DISK-USAGE-HIGH command.
disk1perusage	<Double>	Read-only	Used space in /var partition of the disk, as a percentage. This is a critical counter. You can configure /var Used (%) by using the Set snmp alarm DISK-USAGE-HIGH command.
disk0avail	<Double>	Read-only	Available space in /flash partition of the hard disk.
disk1avail	<Double>	Read-only	Available space in /var partition of the hard disk.
totrxmbits	<Double>	Read-only	Number of megabytes received by the NetScaler appliance.
rxmbitsrate	<Double>	Read-only	Rate (/s) counter for totrxmbits
tottxmbits	<Double>	Read-only	Number of megabytes transmitted by the NetScaler appliance.
txmbitsrate	<Double>	Read-only	Rate (/s) counter for tottxmbits
tcpcurclientconn	<Double>	Read-only	Client connections, including connections in the Opening, Established, and Closing state.
tcpcurclientconnestablished	<Double>	Read-only	Current client connections in the Established state, which indicates that data transfer can occur between the NetScaler and the client.
tcpcurserverconn	<Double>	Read-only	Server connections, including connections in the Opening, Established, and Closing state.
tcpcurserverconnestablished	<Double>	Read-only	Current server connections in the Established state, which indicates that data transfer can occur between the NetScaler and the server.
httptotrequests	<Double>	Read-only	Total number of HTTP requests received.
httprequestsrate	<Double>	Read-only	Rate (/s) counter for httptotrequests

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
httpresponsesrate	<Double>	Read-only	Rate (/s) counter for httptotresponses
httptotrxrequestbytes	<Double>	Read-only	Total number of bytes of HTTP request data received.
httprxrequestbytesrate	<Double>	Read-only	Rate (/s) counter for httptotrxrequestbytes
httptotrxresponsebytes	<Double>	Read-only	Total number of bytes of HTTP response data received.
httprxresponsebytesrate	<Double>	Read-only	Rate (/s) counter for httptotrxresponsebytes
ssltottransactions	<Double>	Read-only	Number of SSL transactions on the NetScaler appliance.
ssltransactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottransactions
ssltotsessionhits	<Double>	Read-only	Number of SSL session reuse hits on the NetScaler appliance.
sslsessionhitsrate	<Double>	Read-only	Rate (/s) counter for ssltotsessionhits
appfirewallrequests	<Double>	Read-only	HTTP/HTTPS requests sent to your protected web servers via the Application Firewall.
appfirewallrequestsrate	<Double>	Read-only	Rate (/s) counter for appfirewallrequests
appfirewallresponses	<Double>	Read-only	HTTP/HTTPS responses sent by your protected web servers via the Application Firewall.
appfirewallresponsesrate	<Double>	Read-only	Rate (/s) counter for appfirewallresponses
appfirewallaborts	<Double>	Read-only	Incomplete HTTP/HTTPS requests aborted by the client before the Application Firewall could finish processing them.
appfirewallabortsrate	<Double>	Read-only	Rate (/s) counter for appfirewallaborts
appfirewallredirects	<Double>	Read-only	HTTP/HTTPS requests redirected by the Application Firewall to a different Web page or web server. (HTTP 302)
appfirewallredirectsrate	<Double>	Read-only	Rate (/s) counter for appfirewallredirects
misccounter0	<Double>	Read-only	Miscellaneous Counter 0.
misccounter1	<Double>	Read-only	Miscellaneous Counter 1.
numcpus	<Double>	Read-only	The number of CPUs on the NetScaler appliance.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN= *<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/ns

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/ns?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get ns resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ns": [ {
    "appfirewallredirectsrate":<Double_value>,
    "ssltottransactions":<Double_value>,
    "sslnumcardsup":<Double_value>,
    "hacurmasterstate":<String_value>,
    "httprxrequestbytesrate":<Double_value>,
    "httptotrxresponsebytes":<Double_value>,
    "disk1perusage":<Double_value>,
    "rescpuusagepcnt":<Double_value>,
    "appfirewallresponsesrate":<Double_value>,
    "delcmpratio":<Double_value>,
    "appfirewallresponses":<Double_value>,
    "comptotaldatacompressionratio":<Double_value>,
    "httprxresponsebytesrate":<Double_value>,
    "disk0perusage":<Double_value>,
    "appfirewallabortsrate":<Double_value>,
    "appfirewallrequestsrate":<Double_value>,
    "ssltotsessionhits":<Double_value>,
    "appfirewallredirects":<Double_value>,
    "mgmtcpuusagepcnt":<Double_value>,
    "cpuusage":<Double_value>,
    "httptotresponses":<Double_value>,
    "compratio":<Double_value>,
    "memuseinmb":<Double_value>,
    "sslsessionhitsrate":<Double_value>,
    "resmemusage":<Double_value>,
    "cache64maxmemorykb":<Double_value>,
    "starttime":<String_value>,
    "cachepercentoriginbandwidthsaved":<Double_value>,
    "misccounter0":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "memusagepcnt":<Double_value>,
    "cachehitsrate":<Double_value>,
    "transtime":<String_value>,
    "cacheutilizedmemorykb":<Double_value>,
    "appfirewallaborts":<Double_value>,
    "disk1avail":<Double_value>,
    "tcpcurserverconnestablished":<Double_value>,
    "tcpcurclientconn":<Double_value>,
    "tcpcurserverconn":<Double_value>,
    "rescpuusage":<Double_value>,
    "appfirewallrequests":<Double_value>,
    "totrxmbits":<Double_value>,
    "tottxmbits":<Double_value>,
    "cachemissesrate":<Double_value>,
    "cachemaxmemorykb":<Double_value>,
    "httptotrequests":<Double_value>,
    "cachetothits":<Double_value>,
    "misccounter1":<Double_value>,
    "numcpus":<Double_value>,
    "pktcpuusagepcnt":<Double_value>,
    "ssltransactionsrate":<Double_value>,
    "httprequestsrate":<Double_value>,
    "cachemaxmemoryactivekb":<Double_value>,
    "httpresponsesrate":<Double_value>,
    "disk0avail":<Double_value>,
    "txmbitsrate":<Double_value>,
    "sslcards":<Double_value>,
    "hacurstate":<String_value>,
    "rxmbitsrate":<Double_value>,
    "tcpcurclientconnestablished":<Double_value>,
    "cpuusagepcnt":<Double_value>,
    "cachetotmisses":<Double_value>,
    "httptotrxrequestbytes":<Double_value>
}]}
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

nsacl

May 20, 2016

Statistics for ACL entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
aclname	<String>	Read-write	Name of the extended ACL rule whose statistics you want the NetScaler appliance to display. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
acltotpktsbridged	<Double>	Read-only	Packets matching a bridge ACL, which is in transparent mode and bypasses service processing.
aclpktsbridgedrate	<Double>	Read-only	Rate (/s) counter for acltotpktsbridged
acltotpktsdenied	<Double>	Read-only	Packets dropped because they match ACLs with processing mode set to DENY.
aclpktsdeniedrate	<Double>	Read-only	Rate (/s) counter for acltotpktsdenied
acltotpktsallowed	<Double>	Read-only	Packets matching ACLs with processing mode set to ALLOW. NetScaler processes these packets.
aclpktsallowedrate	<Double>	Read-only	Rate (/s) counter for acltotpktsallowed
acltotpktsnat	<Double>	Read-only	Packets matching a NAT ACL, resulting in a NAT session.
aclpktsnatrate	<Double>	Read-only	Rate (/s) counter for acltotpktsnat
acltothits	<Double>	Read-only	Packets matching an ACL.
aclhitsrate	<Double>	Read-only	Rate (/s) counter for acltothits
acltotmisses	<Double>	Read-only	Packets not matching any ACL.
aclmissesrate	<Double>	Read-only	Rate (/s) counter for acltotmisses
acltotcount	<Double>	Read-only	Total number of ACL rules configured.
aclperhits	<Double>	Read-only	Number of times the acl was hit
aclperhitsrate	<Double>	Read-only	Rate (/s) counter for aclperhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nsacl

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nsacl?**args=aclname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_valu**

Use this query-parameter to get nsacl resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsacl": [ {
    "aclname":<String_value>,
    "acltotpktsnat":<Double_value>,
    "aclpktsdeniedrate":<Double_value>,
    "aclhitsrate":<Double_value>,
    "aclpktsallowedrate":<Double_value>,
    "acltotpktsallowed":<Double_value>,
    "aclpktsbridgedrate":<Double_value>,
    "acltotpktsdenied":<Double_value>,
    "acltothits":<Double_value>,
    "aclperhits":<Double_value>,
    "aclperhitsrate":<Double_value>,
    "aclpktsnatrate":<Double_value>,
    "acltotcount":<Double_value>,
    "acltotmisses":<Double_value>,
    "acltotpktsbridged":<Double_value>,
    "aclmissesrate":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/nsacl/aclname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsacl": [ {
    "aclname":<String_value>,
    "acltotpktsnat":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "aclpktsdeniedrate":<Double_value>,
    "aclhitsrate":<Double_value>,
    "aclpktsallowedrate":<Double_value>,
    "acltotpktsallowed":<Double_value>,
    "aclpktsbridgedrate":<Double_value>,
    "acltotpktsdenied":<Double_value>,
    "acltothits":<Double_value>,
    "aclperhits":<Double_value>,
    "aclperhitsrate":<Double_value>,
    "aclpktsnatrate":<Double_value>,
    "acltotcount":<Double_value>,
    "acltotmisses":<Double_value>,
    "acltotpktsbridged":<Double_value>,
    "aclmissesrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsacl6

Jul 07, 2017
Statistics for ACL6 entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
acl6name	<String>	Read-write	Name of the ACL6 rule whose statistics you want the NetScaler appliance to display. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
acl6totpktsbridged	<Double>	Read-only	Packets matching a bridge IPv6 ACL, which is in transparent mode and bypasses service processing.
acl6pktsbridgedrate	<Double>	Read-only	Rate (/s) counter for acl6totpktsbridged
acl6totpktsdenied	<Double>	Read-only	Packets dropped because they match IPv6 ACLs with processing mode set to DENY.
acl6pktsdeniedrate	<Double>	Read-only	Rate (/s) counter for acl6totpktsdenied
acl6totpktsallowed	<Double>	Read-only	Packets matching IPv6 ACLs with processing mode set to ALLOW. NetScaler processes these packets.
acl6pktsallowedrate	<Double>	Read-only	Rate (/s) counter for acl6totpktsallowed
acl6totpktsnat	<Double>	Read-only	Packets matching a NAT ACL6, resulting in a NAT session.
acl6pktsnatrate	<Double>	Read-only	Rate (/s) counter for acl6totpktsnat
acl6tothits	<Double>	Read-only	Packets matching an IPv6 ACL.
acl6hitsrate	<Double>	Read-only	Rate (/s) counter for acl6tothits
acl6totmisses	<Double>	Read-only	Packets not matching any IPv6 ACL.
acl6missesrate	<Double>	Read-only	Rate (/s) counter for acl6totmisses
acl6totpktsnat64	<Double>	Read-only	Packets matching a NAT64 ACL6, resulting in a NAT64 translation.
acl6pktsnat64rate	<Double>	Read-only	Rate (/s) counter for acl6totpktsnat64
acl6totcount	<Double>	Read-only	Total number of ACL6 rules configured.
acl6perhits	<Double>	Read-only	Number of times the acl6 was hit
acl6perhitsrate	<Double>	Read-only	Rate (/s) counter for acl6perhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To do this, you must specify the username and password in the request header of the NITRO request as follows:

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nsacl6

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nsacl6?**args=acl6name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_v**

Use this query-parameter to get nsacl6 resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsacl6": [ {
    "acl6name":<String_value>,
    "acl6pktsnatrate":<Double_value>,
    "acl6missesrate":<Double_value>,
    "acl6totmisses":<Double_value>,
    "acl6totpktsnat64":<Double_value>,
    "acl6pktsnat64rate":<Double_value>,
    "acl6totpktsnat":<Double_value>,
    "acl6perhitsrate":<Double_value>,
    "acl6perhits":<Double_value>,
    "acl6tothits":<Double_value>,
    "acl6pktsbridgedrate":<Double_value>,
    "acl6pktsallowedrate":<Double_value>,
    "acl6totpktsbridged":<Double_value>,
    "acl6totpktsdenied":<Double_value>,
    "acl6totcount":<Double_value>,
    "acl6totpktsallowed":<Double_value>,
    "acl6pktsdeniedrate":<Double_value>,
    "acl6hitsrate":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/nsacl6/acl6name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "nsacl6": [ {
      "acl6name":<String_value>,
      "acl6pktsnatrate":<Double_value>,
      "acl6missesrate":<Double_value>,
      "acl6totmisses":<Double_value>,
      "acl6totpktsnat64":<Double_value>,
      "acl6pktsnat64rate":<Double_value>,
      "acl6totpktsnat":<Double_value>,
      "acl6perhitsrate":<Double_value>,
      "acl6perhits":<Double_value>,
      "acl6tothits":<Double_value>,
      "acl6pktsbridgedrate":<Double_value>,
      "acl6pktsallowedrate":<Double_value>,
      "acl6totpktsbridged":<Double_value>,
      "acl6totpktsdenied":<Double_value>,
      "acl6totcount":<Double_value>,
      "acl6totpktsallowed":<Double_value>,
      "acl6pktsdeniedrate":<Double_value>,
      "acl6hitsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nslimitidentifier

May 20, 2016

Statistics for limit Indetifier resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the identifier. Minimum length = 1
pattern	<String[]>	Read-write	Pattern for the selector field, ? means field is required, * means field value does not matter, anything else is a regular pattern.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sortby	<String>	Read-write	use this argument to sort by specific key. Possible values =
sortorder	<String>	Read-write	use this argument to specify sort order. Default value: SORT_DESCENDING Possible values = ascending, descending
ratelmtobjhits	<Double>	Read-only	Total hits.
ratelmtobjdrops	<Double>	Read-only	Total drops
ratelmtsessionobjhits	<Double>	Read-only	Total hits.

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nslimitidentifier

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nslimitidentifier?**args=name:<String_value>,pattern:<String[]_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<Strin**

Use this query-parameter to get nslimitidentifier resources based on additional properties.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslimitidentifier": [ {
    "name":<String_value>,
    "ratelmtobjhits":<Double_value>,
    "ratelmtsessionobjhits":<Double_value>,
    "ratelmtobjdrops":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/nslimitidentifier/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nslimitidentifier": [ {
    "name":<String_value>,
    "ratelmtobjhits":<Double_value>,
    "ratelmtsessionobjhits":<Double_value>,
    "ratelmtobjdrops":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nsmemory

Jul 07, 2017
Statistics for resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
pool	<String>	Read-write	Feature name for which to display memory statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
allocf	<Double>	Read-only	Memory allocation failure for particular feature.
memcurallocper	<Double>	Read-only	Percentage of NetScaler memory used by the feature.
memcurinkb	<Double>	Read-only	Total current NetScaler memory available for use by the feature, in kilobytes.

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nsmemory

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nsmemory?**args=pool:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_va**

Use this query-parameter to get nsmemory resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsmemory": [ {
    "pool":<String_value>,
    "memcurinkb":<Double_value>,
    "allocf":<Double_value>,
    "memcurallocper":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/nsmemory/pool_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nsmemory": [ {
    "pool":<String_value>,
    "memcurinkb":<Double_value>,
    "allocf":<Double_value>,
    "memcurallocper":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nspartition

May 20, 2016

Statistics for admin partition resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
partitionname	<String>	Read-write	Name of the partition. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
minbandwidth	<Double>	Read-only	Minimum Bandwidth required by the partition.
maxbandwidth	<Double>	Read-only	Maximum Banwidth allowed for the partition.
maxconnection	<Double>	Read-only	Maximum Connection allowed for the partition.
maxmemory	<Double>	Read-only	Maximum memory limit for the partition.
currentbandwidth	<Double>	Read-only	Current Bandwidth usage for the partition.
currentconnections	<Double>	Read-only	Current Connections on this partition.
memoryusagepcnt	<Double>	Read-only	Memory usage(%) on this partition.
totaldrops	<Double>	Read-only	Total packet drops for the partition.
dropsrate	<Double>	Read-only	Rate (/s) counter for totaldrops
totaltokendrops	<Double>	Read-only	Total drops(KB) for the partition.
tokendropsrate	<Double>	Read-only	Rate (/s) counter for totaltokendrops
totalconnectiondrops	<Double>	Read-only	Total connection drops for the partition.
connectiondropsrate	<Double>	Read-only	Rate (/s) counter for totalconnectiondrops

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nspartition

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nspartition?**args=partitionname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:**

Use this query-parameter to get nspartition resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspartition": [ {
    "partitionname":<String_value>,
    "currentconnections":<Double_value>,
    "tokendropsrate":<Double_value>,
    "totaltokendrops":<Double_value>,
    "maxbandwidth":<Double_value>,
    "minbandwidth":<Double_value>,
    "totaldrops":<Double_value>,
    "connectiondropsrate":<Double_value>,
    "totalconnectiondrops":<Double_value>,
    "dropsrate":<Double_value>,
    "currentbandwidth":<Double_value>,
    "maxconnection":<Double_value>,
    "maxmemory":<Double_value>,
    "memoryusagepcnt":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/nspartition/partitionname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspartition": [ {
    "partitionname":<String_value>,
    "currentconnections":<Double_value>,
    "tokendropsrate":<Double_value>,
    "totaltokendrops":<Double_value>,
    "maxbandwidth":<Double_value>,
    "minbandwidth":<Double_value>,
    "totaldrops":<Double_value>,
    "connectiondropsrate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "totalconnectiondrops":<Double_value>,
    "dropsrate":<Double_value>,
    "currentbandwidth":<Double_value>,
    "maxconnection":<Double_value>,
    "maxmemory":<Double_value>,
    "memoryusagepcnt":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nspbr

May 20, 2016

Statistics for Policy Based Routing(PBR) entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the PBR whose statistics you want the NetScaler appliance to display. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pbrtotpktsallowed	<Double>	Read-only	Total packets that matched the PBR (Policy-Based Routes) with action ALLOW
pbrpktsallowedrate	<Double>	Read-only	Rate (/s) counter for pbrtotpktsallowed
pbrtotpktsdenied	<Double>	Read-only	Total packets that matched the PBR with action DENY
pbrpktsdeniedrate	<Double>	Read-only	Rate (/s) counter for pbrtotpktsdenied
pbrtothits	<Double>	Read-only	Total packets that matched one of the configured PBR
pbrhitsrate	<Double>	Read-only	Rate (/s) counter for pbrtothits
pbrtotmisses	<Double>	Read-only	Total packets that did not match any PBR
pbrmissesrate	<Double>	Read-only	Rate (/s) counter for pbrtotmisses
pbrperhits	<Double>	Read-only	Number of times the pbr was hit
pbrperhitsrate	<Double>	Read-only	Rate (/s) counter for pbrperhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nspbr

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nspbr?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get nspbr resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspbr": [ {
    "name":<String_value>,
    "pbrpktsdeniedrate":<Double_value>,
    "pbrmissesrate":<Double_value>,
    "pbrtothits":<Double_value>,
    "pbrperhitsrate":<Double_value>,
    "pbrhitsrate":<Double_value>,
    "pbrperhits":<Double_value>,
    "pbrtotmisses":<Double_value>,
    "pbrtotpktsallowed":<Double_value>,
    "pbrtotpktsdenied":<Double_value>,
    "pbrpktsallowedrate":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/nspbr/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspbr": [ {
    "name":<String_value>,
    "pbrpktsdeniedrate":<Double_value>,
    "pbrmissesrate":<Double_value>,
    "pbrtothits":<Double_value>,
    "pbrperhitsrate":<Double_value>,
    "pbrhitsrate":<Double_value>,
    "pbrperhits":<Double_value>,
    "pbrtotmisses":<Double_value>,
    "pbrtotpktsallowed":<Double_value>,
    "pbrtotpktsdenied":<Double_value>,
    "pbrpktsallowedrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nspbr6

May 20, 2016

Statistics for PBR6 entry resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the PBR6 whose statistics you want the NetScaler appliance to display. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pbr6totpktsallowed	<Double>	Read-only	Total packets that matched the PBR6 with action ALLOW
pbr6pktsallowedrate	<Double>	Read-only	Rate (/s) counter for pbr6totpktsallowed
pbr6totpktsdenied	<Double>	Read-only	Total packets that matched PBR6 with action DENY
pbr6pktsdeniedrate	<Double>	Read-only	Rate (/s) counter for pbr6totpktsdenied
pbr6tothits	<Double>	Read-only	Total packets that matched one of the configured PBR6
pbr6hitsrate	<Double>	Read-only	Rate (/s) counter for pbr6tothits
pbr6totmisses	<Double>	Read-only	Total packets that did not match any PBR6
pbr6missesrate	<Double>	Read-only	Rate (/s) counter for pbr6totmisses
pbr6perhits	<Double>	Read-only	Number of times the pbr6 was hit
pbr6perhitsrate	<Double>	Read-only	Rate (/s) counter for pbr6perhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nspbr6

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nspbr6?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value**

Use this query-parameter to get nspbr6 resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspbr6": [ {
    "name":<String_value>,
    "pbr6pktsallowedrate":<Double_value>,
    "pbr6pktsdeniedrate":<Double_value>,
    "pbr6perhits":<Double_value>,
    "pbr6totpktsallowed":<Double_value>,
    "pbr6missesrate":<Double_value>,
    "pbr6tothits":<Double_value>,
    "pbr6perhitsrate":<Double_value>,
    "pbr6totpktsdenied":<Double_value>,
    "pbr6hitsrate":<Double_value>,
    "pbr6totmisses":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/nspbr6/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nspbr6": [ {
    "name":<String_value>,
    "pbr6pktsallowedrate":<Double_value>,
    "pbr6pktsdeniedrate":<Double_value>,
    "pbr6perhits":<Double_value>,
    "pbr6totpktsallowed":<Double_value>,
    "pbr6missesrate":<Double_value>,
    "pbr6tothits":<Double_value>,
    "pbr6perhitsrate":<Double_value>,
    "pbr6totpktsdenied":<Double_value>,
    "pbr6hitsrate":<Double_value>,
    "pbr6totmisses":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

nssimpleacl

May 20, 2016

Statistics for simple ACL resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sacltothits	<Double>	Read-only	Packets matching a SimpleACL.
saclhitsrate	<Double>	Read-only	Rate (/s) counter for sacltothits
sacltotmisses	<Double>	Read-only	Packets not matching any SimpleACL.
saclmissesrate	<Double>	Read-only	Rate (/s) counter for sacltotmisses
saclscount	<Double>	Read-only	Number of SimpleACLs configured.
sacltotpktsallowed	<Double>	Read-only	Total packets that matched a SimpleACL with action ALLOW and got consumed by NetScaler.
saclpktsallowedrate	<Double>	Read-only	Rate (/s) counter for sacltotpktsallowed
sacltotpktsbridged	<Double>	Read-only	Total packets that matched a SimpleACL with action BRIDGE and got bridged by NetScaler.
saclpktsbridgedrate	<Double>	Read-only	Rate (/s) counter for sacltotpktsbridged
sacltotpktsdenied	<Double>	Read-only	Packets dropped because they match SimpleACL (Access Control List) with processing mode set to DENY.
saclpktsdeniedrate	<Double>	Read-only	Rate (/s) counter for sacltotpktsdenied

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nssimpleacl

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nssimpleacl?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get nssimpleacl resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nssimpleacl": [ {
    "sacltothits":<Double_value>,
    "saclscount":<Double_value>,
    "saclhitsrate":<Double_value>,
    "saclpktsbridgedrate":<Double_value>,
    "sacltotpktsdenied":<Double_value>,
    "sacltotmisses":<Double_value>,
    "saclmissesrate":<Double_value>,
    "saclpktsallowedrate":<Double_value>,
    "sacltotpktsbridged":<Double_value>,
    "saclpktsdeniedrate":<Double_value>,
    "sacltotpktsallowed":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nssimpleacl6

May 20, 2016

Statistics for simple ACL6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sacl6tothits	<Double>	Read-only	Packets matching a SimpleACL6.
sacl6hitsrate	<Double>	Read-only	Rate (/s) counter for sacl6tothits
sacl6totmisses	<Double>	Read-only	Packets not matching any SimpleACL6.
sacl6missesrate	<Double>	Read-only	Rate (/s) counter for sacl6totmisses
sacl6scount	<Double>	Read-only	Number of SimpleACL6s configured.
sacl6totpktsallowed	<Double>	Read-only	Total packets that matched a SimpleACL6 with action ALLOW and got consumed by NetScaler.
sacl6pktsallowedrate	<Double>	Read-only	Rate (/s) counter for sacl6totpktsallowed
sacl6totpktsbridged	<Double>	Read-only	Total packets that matched a SimpleACL6 with action BRIDGE and got bridged by NetScaler.
sacl6pktsbridgedrate	<Double>	Read-only	Rate (/s) counter for sacl6totpktsbridged
sacl6totpktsdenied	<Double>	Read-only	Packets dropped because they match SimpleACL6 with processing mode set to DENY.
sacl6pktsdeniedrate	<Double>	Read-only	Rate (/s) counter for sacl6totpktsdenied

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nssimpleacl6

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nssimpleacl6?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get nssimpleacl6 resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nssimpleacl6": [ {
    "sacl6scount":<Double_value>,
    "sacl6pktsbridgedrate":<Double_value>,
    "sacl6totpktsallowed":<Double_value>,
    "sacl6totmisses":<Double_value>,
    "sacl6missesrate":<Double_value>,
    "sacl6hitsrate":<Double_value>,
    "sacl6tothits":<Double_value>,
    "sacl6totpktsdenied":<Double_value>,
    "sacl6pktsallowedrate":<Double_value>,
    "sacl6totpktsbridged":<Double_value>,
    "sacl6pktsdeniedrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

nstrafficdomain

Jul 07, 2017
Statistics for Traffic Domain resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
td	<Double>	Read-write	An integer specifying the Traffic Domain ID. Possible values: 1 through 4094. Minimum value = 1 Maximum value = 4094
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
nstdtotrxpkts	<Double>	Read-only	Packets received on this TD.
nstdrxpktsrate	<Double>	Read-only	Rate (/s) counter for nstdtotrxpkts
nstdtottxpkts	<Double>	Read-only	Packets transmitted from this TD.
nstdtxpktsrate	<Double>	Read-only	Rate (/s) counter for nstdtottxpkts
nstdtotdroppedpkts	<Double>	Read-only	Inbound packets dropped on this TD by reception.
nstddroppedpktsrate	<Double>	Read-only	Rate (/s) counter for nstdtotdroppedpkts

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/nstrafficdomain

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/nstrafficdomain?**args=td:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String**

Use this query-parameter to get nstrafficdomain resources based on additional properties.

HTTP Method: GET

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstrafficdomain": [ {
    "td":<Double_value>,
    "nstdtotrxpkts":<Double_value>,
    "nstddroppedpktsrate":<Double_value>,
    "nstdtotdroppedpkts":<Double_value>,
    "nstdrxpktsrate":<Double_value>,
    "nstdtottxpkts":<Double_value>,
    "nstdtxpktsrate":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/nstrafficdomain/td_value><Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "nstrafficdomain": [ {
    "td":<Double_value>,
    "nstdtotrxpkts":<Double_value>,
    "nstddroppedpktsrate":<Double_value>,
    "nstdtotdroppedpkts":<Double_value>,
    "nstdrxpktsrate":<Double_value>,
    "nstdtottxpkts":<Double_value>,
    "nstdtxpktsrate":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Priority Queuing

May 20, 2016

Priority Queuing statistics. Priority Queuing ensures that key customer web sites get the timely responses they need. System provides priority-based queuing, which provides different levels of service to customers.

Name	Description
pq	Statistics for pq.
pqpolicy	Statistics for PQ policy resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

pq

Jul 07, 2017

Statistics for pq.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pqtotalpolicymatches	<Double>	Read-only	Number of times the Netscaler appliance matched an incoming request using any priority queuing policy.
pqpolicymatchesrate	<Double>	Read-only	Rate (/s) counter for pqtotalpolicymatches
pqtotalthresholdfailed	<Double>	Read-only	Number of times the Netscaler appliance failed to match an incoming request to any of priority queing policy.
pqthresholdfailedrate	<Double>	Read-only	Rate (/s) counter for pqtotalthresholdfailed
pqpriority1requests	<Double>	Read-only	Number of priority 1 requests that the Netscaler appliance received.
pqpriority1requestsrate	<Double>	Read-only	Rate (/s) counter for pqpriority1requests
pqpriority2requests	<Double>	Read-only	Number of priority 2 requests that the Netscaler appliance received.
pqpriority2requestsrate	<Double>	Read-only	Rate (/s) counter for pqpriority2requests
pqpriority3requests	<Double>	Read-only	Number of priority 3 requests that the Netscaler appliance received.
pqpriority3requestsrate	<Double>	Read-only	Rate (/s) counter for pqpriority3requests

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> Note

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/pq

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/pq?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get pq resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "pq": [ {
    "pqpriority2requestsrate":<Double_value>,
    "pqpolicymatchesrate":<Double_value>,
    "pqpriority1requestsrate":<Double_value>,
    "pqthresholdfailedrate":<Double_value>,
    "pqtotalpolicymatches":<Double_value>,
    "pqpriority1requests":<Double_value>,
    "pqpriority3requestsrate":<Double_value>,
    "pqpriority3requests":<Double_value>,
    "pqpriority2requests":<Double_value>,
    "pqtotalthresholdfailed":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

pqpolicy

May 20, 2016

Statistics for PQ policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
policyname	<String>	Read-write	Name of the priority queuing policy whose statistics must be displayed. If a name is not provided, statistics of all priority queuing policies are shown. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pqtotavgqueuewaittime	<Double>	Read-only	Average wait time for clients for this priority queuing policy.
pqavgqueuewaittimerate	<Double>	Read-only	Rate (/s) counter for pqtotavgqueuewaittime
pqavgclienttransactiontimems	<Double>	Read-only	Average time taken by a priority queuing client to complete its transaction for this priority queuing policy.
pqavgclienttransactiontimemsrate	<Double>	Read-only	Rate (/s) counter for pqavgclienttransactiontimems
pqvserverip	<String>	Read-only	IP address of the virtual server to which this priority queuing policy is bound.
pqvserverport	<Integer>	Read-only	Port number of the virtual server to which this priority queuing policy is bound.
pqqdepth	<Double>	Read-only	Number of clients waiting currently for this priority queuing policy.
pqqdepthrate	<Double>	Read-only	Rate (/s) counter for pqqdepth
pqcurrentclientconnections	<Double>	Read-only	Current number of server connections established for serving clients for this priority queuing policy.
pqcurrentclientconnectionsrate	<Double>	Read-only	Rate (/s) counter for pqcurrentclientconnections
pqtotclientconnections	<Double>	Read-only	Total number of server connections established for serving clients for this priority queuing policy.
pqclientconnectionsrate	<Double>	Read-only	Rate (/s) counter for pqtotclientconnections
pqdropped	<Double>	Read-only	Total number of dropped transactions for this priority queuing policy.
pqdroppedrate	<Double>	Read-only	Rate (/s) counter for pqdropped
totclienttransactions	<Double>	Read-only	Total number of client transactions for this priority queuing policy.
clienttransactionsrate	<Double>	Read-only	Rate (/s) counter for totclienttransactions
pqtotqueuedepth	<Double>	Read-only	Total number of waiting clients for this priority queuing policy.
pqqueuedepthrate	<Double>	Read-only	Rate (/s) counter for pqtotqueuedepth

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/pqpolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/pqpolicy?**args=policyname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<Strin**

Use this query-parameter to get pqpolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "pqpolicy": [ {
    "policyname":<String_value>,
    "pqvserverip":<String_value>,
    "pqqueuedepthrate":<Double_value>,
    "pqdroppedrate":<Double_value>,
    "pqtotqueuedepth":<Double_value>,
    "pqvserverport":<Integer_value>,
    "totclienttransactions":<Double_value>,
    "clienttransactionsrate":<Double_value>,
    "pqcurrentclientconnectionsrate":<Double_value>,
    "pqqdepthrate":<Double_value>,
    "pqqdepth":<Double_value>,
    "pqclientconnectionsrate":<Double_value>,
    "pqavgclienttransactiontimems":<Double_value>,
    "pqavgclienttransactiontimemsrate":<Double_value>,
    "pqtotavgqueuewaittime":<Double_value>,
    "pqdropped":<Double_value>,
    "pqtotclientconnections":<Double_value>,
    "pqavgqueuewaittimerate":<Double_value>,
    "pqcurrentclientconnections":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/pqpolicy/policyname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "pqpolicy": [ {
    "policyname":<String_value>,
    "pqvserverip":<String_value>,
    "pqqueuedepthrate":<Double_value>,
    "pqdroppedrate":<Double_value>,
    "pqtotqueuedepth":<Double_value>,
    "pqvserverport":<Integer_value>,
    "totclienttransactions":<Double_value>,
    "clienttransactionsrate":<Double_value>,
    "pqcurrentclientconnectionsrate":<Double_value>,
    "pqqdepthrate":<Double_value>,
    "pqqdepth":<Double_value>,
    "pqclientconnectionsrate":<Double_value>,
    "pqavgclienttransactiontimems":<Double_value>,
    "pqavgclienttransactiontimemsrate":<Double_value>,
    "pqtotavgqueuewaittime":<Double_value>,
    "pqdropped":<Double_value>,
    "pqtotclientconnections":<Double_value>,
    "pqavgqueuewaittimerate":<Double_value>,
    "pqcurrentclientconnections":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Protocol

Jul 07, 2017

Protocol statistics.

Name	Description
protocolhttp	Statistics for HTTP resource.
protocolicmp	Statistics for icmp resource.
protocolicmpv6	Statistics for icmpv6 resource.
protocolip	Statistics for protocolip resource.
protocolipv6	Statistics for ip v6 resource.
protocoltcp	Statistics for tcp protocol resource.
protocoludp	Statistics for UDP Protocol resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

protocolhttp

May 20, 2016

Statistics for HTTP resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
spdytotstreams	<Double>	Read-only	Total number of requests received over SPDYv2 and SPDYv3
spdystreamsrate	<Double>	Read-only	Rate (/s) counter for spdytotstreams
httptotrequests	<Double>	Read-only	Total number of HTTP requests received.
httprequestsrate	<Double>	Read-only	Rate (/s) counter for httptotrequests
httptotresponses	<Double>	Read-only	Total number of HTTP responses sent.
httpresponsesrate	<Double>	Read-only	Rate (/s) counter for httptotresponses
httptotrxrequestbytes	<Double>	Read-only	Total number of bytes of HTTP request data received.
httprxrequestbytesrate	<Double>	Read-only	Rate (/s) counter for httptotrxrequestbytes
httptotrxresponsebytes	<Double>	Read-only	Total number of bytes of HTTP response data received.
httprxresponsebytesrate	<Double>	Read-only	Rate (/s) counter for httptotrxresponsebytes
httptotgets	<Double>	Read-only	Total number of HTTP requests received with the GET method.
httpgetsrate	<Double>	Read-only	Rate (/s) counter for httptotgets
httptotposts	<Double>	Read-only	Total number of HTTP requests received with the POST method.
httppostsrate	<Double>	Read-only	Rate (/s) counter for httptotposts
httptotothers	<Double>	Read-only	Total number of HTTP requests received with methods other than GET and POST. Some of the other well-defined HTTP methods are HEAD, PUT, DELETE, OPTIONS, and TRACE. User-defined methods are also allowed.
httptohersrate	<Double>	Read-only	Rate (/s) counter for httptotothers
httptot10requests	<Double>	Read-only	Total number of HTTP/1.0 requests received.
http10requestsrate	<Double>	Read-only	Rate (/s) counter for httptot10requests
httptot11requests	<Double>	Read-only	Total number of HTTP/1.1 requests received.
http11requestsrate	<Double>	Read-only	Rate (/s) counter for httptot11requests
httptotclenrequests	<Double>	Read-only	Total number of HTTP requests in which the Content-length field of the HTTP header has been set. Content-length specifies the length of the content, in bytes, in the associated HTTP body.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
httptotchunkedrequests	<Double>	Read-only	Total number of HTTP requests in which the Transfer-Encoding field of the HTTP header has been set to chunked.
httpchunkedrequestsrate	<Double>	Read-only	Rate (/s) counter for httptotchunkedrequests
httptottxrequestbytes	<Double>	Read-only	Total number of bytes of HTTP request data transmitted.
httptxrequestbytesrate	<Double>	Read-only	Rate (/s) counter for httptottxrequestbytes
httptot10responses	<Double>	Read-only	Total number of HTTP/1.0 responses sent.
http10responsesrate	<Double>	Read-only	Rate (/s) counter for httptot10responses
httptot11responses	<Double>	Read-only	Total number of HTTP/1.1 responses sent.
http11responsesrate	<Double>	Read-only	Rate (/s) counter for httptot11responses
httptotclenresponses	<Double>	Read-only	Total number of HTTP responses sent in which the Content-length field of the HTTP header has been set. Content-length specifies the length of the content, in bytes, in the associated HTTP body.
httpclenresponsesrate	<Double>	Read-only	Rate (/s) counter for httptotclenresponses
httptotchunkedresponses	<Double>	Read-only	Total number of HTTP responses sent in which the Transfer-Encoding field of the HTTP header has been set to chunked. This setting is used when the server wants to start sending the response before knowing its total length. The server breaks the response into chunks and sends them in sequence, inserting the length of each chunk before the actual data. The message ends with a chunk of size zero.
httpchunkedresponsesrate	<Double>	Read-only	Rate (/s) counter for httptotchunkedresponses
httperrnoreusemultipart	<Double>	Read-only	Total number of HTTP multi-part responses sent. In multi-part responses, one or more entities are encapsulated within the body of a single message.
httperrnoreusemultipartrate	<Double>	Read-only	Rate (/s) counter for httperrnoreusemultipart
httptotnoclenchunkresponses	<Double>	Read-only	Total number of FIN-terminated responses sent. In FIN-terminated responses, the server finishes sending the data and closes the connection.
httpnoclenchunkresponsesrate	<Double>	Read-only	Rate (/s) counter for httptotnoclenchunkresponses
httptottxresponsebytes	<Double>	Read-only	Total number of bytes of HTTP response data transmitted.
httptxresponsebytesrate	<Double>	Read-only	Rate (/s) counter for httptottxresponsebytes
httperrincompleteheaders	<Double>	Read-only	Total number of HTTP requests and responses received in which the HTTP header spans more than one packet.
httperrincompleterequests	<Double>	Read-only	Total number of HTTP requests received in which the header spans more than one packet.
httperrincompleterequestsrate	<Double>	Read-only	Rate (/s) counter for httperrincompleterequests
httperrincompleteresponses	<Double>	Read-only	Total number of HTTP responses received in which the header spans more than one packet.
httperrincompleteresponsesrate	<Double>	Read-only	Rate (/s) counter for httperrincompleteresponses
httperrserverbusy	<Double>	Read-only	Total number of HTTP error responses received. Some of the error responses are: 500 Internal Server Error 501 Not Implemented 502 Bad Gateway 503 Service Unavailable 504 Gateway Timeout 505 HTTP Version Not Supported
httperrserverbusyrate	<Double>	Read-only	Rate (/s) counter for httperrserverbusy
httperrlargecontent	<Double>	Read-only	Total number of requests and responses received with large body.
httperrlargechunk	<Double>	Read-only	Total number of requests received with large chunk size, in which the Transfer-Encoding field of the HTTP header has been set to chunked.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
httpcontentlength	<Double>	Read-only	specifies the length of the content, in bytes, in the associated HTTP body.
spdyv2totstreams	<Double>	Read-only	Total number of requests received over SPDYv2
spdyv2streamsrate	<Double>	Read-only	Rate (/s) counter for spdyv2totstreams
spdyv3totstreams	<Double>	Read-only	Total number of requests received over SPDYv3
spdyv3streamsrate	<Double>	Read-only	Rate (/s) counter for spdyv3totstreams

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/protocolhttp

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/protocolhttp?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get protocolhttp resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "protocolhttp": [ {
    "httptotclenresponses":<Double_value>,
    "httperrnoreusemultipart":<Double_value>,
    "spdyv3streamsrate":<Double_value>,
    "httprxrequestbytesrate":<Double_value>,
    "httptotrxresponsebytes":<Double_value>,
    "httptotchunkedresponses":<Double_value>,
    "httperrincompleterequests":<Double_value>,
    "httptot10requests":<Double_value>,
    "httptotgets":<Double_value>,
    "httperrlargecontent":<Double_value>,
    "http10responsesrate":<Double_value>,
    "httperrserverbusy":<Double_value>,
    "httptxrequestbytesrate":<Double_value>,
    "httptotposts":<Double_value>,
    "httptotclenrequests":<Double_value>,
    "httprxresponsebytesrate":<Double_value>,
    "spdyv2streamsrate":<Double_value>,
    "httperrincompleteresponsesrate":<Double_value>,
    "httpnoclenchunkresponsesrate":<Double_value>,
    "httpchunkedrequestsrate":<Double_value>,
    "http11responsesrate":<Double_value>,
    "httptotrequests":<Double_value>,
    "httptotresponses":<Double_value>,
    "httperrincompleteresponses":<Double_value>,
    "spdytotstreams":<Double_value>,
    "httptottxrequestbytes":<Double_value>,
    "httperrserverbusyrate":<Double_value>,
    "http11requestsrate":<Double_value>,
    "spdystreamsrate":<Double_value>,
    "httptxresponsebytesrate":<Double_value>,
    "httptohersrate":<Double_value>,
    "httpchunkedresponsesrate":<Double_value>,
    "httptot11responses":<Double_value>,
    "http10requestsrate":<Double_value>,
    "httprequestsrate":<Double_value>,
    "httpresponsesrate":<Double_value>,
    "httptot10responses":<Double_value>,
    "httptotnoclenchunkresponses":<Double_value>,
    "httperrlargechunk":<Double_value>,
    "httperrnoreusemultipartrate":<Double_value>,
    "spdyv3totstreams":<Double_value>,
    "httperrincompleteheaders":<Double_value>,
    "httptot11requests":<Double_value>,
    "httppostsrate":<Double_value>,
    "httpgetsrate":<Double_value>,
    "httpclenresponsesrate":<Double_value>,
    "httperrlargectlen":<Double_value>,
    "httptottxresponsebytes":<Double_value>,
    "spdyv2totstreams":<Double_value>,
    "httptotchunkedrequests":<Double_value>,
    "httperrincompleterequestsrate":<Double_value>,
    "httpclenrequestsrate":<Double_value>,
    "httptotrxrequestbytes":<Double_value>,
    "httptotothers":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

protocolicmp

May 20, 2016

Statistics for icmp resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
icmptotrxpkts	<Double>	Read-only	ICMP packets received.
icmprxpktsrate	<Double>	Read-only	Rate (/s) counter for icmptotrxpkts
icmptotrxbytes	<Double>	Read-only	Bytes of ICMP data received.
icmprxbytesrate	<Double>	Read-only	Rate (/s) counter for icmptotrxbytes
icmptottxpkts	<Double>	Read-only	ICMP packets transmitted.
icmptxpktsrate	<Double>	Read-only	Rate (/s) counter for icmptottxpkts
icmptottxbytes	<Double>	Read-only	Bytes of ICMP data transmitted.
icmptxbytesrate	<Double>	Read-only	Rate (/s) counter for icmptottxbytes
icmptotrxechoreply	<Double>	Read-only	ICMP Ping echo replies received.
icmprxechoreplyrate	<Double>	Read-only	Rate (/s) counter for icmptotrxechoreply
icmptottxechoreply	<Double>	Read-only	ICMP Ping echo replies transmitted.
icmptxechoreplyrate	<Double>	Read-only	Rate (/s) counter for icmptottxechoreply
icmptotrxecho	<Double>	Read-only	ICMP Ping Echo Request and Echo Reply packets received.
icmprxechorate	<Double>	Read-only	Rate (/s) counter for icmptotrxecho
icmptotdstiplookup	<Double>	Read-only	Total number of MTU lookup on destination IP info received on a need fragmentation ICMP error message failed.
icmpcurratethreshold	<Double>	Read-only	Limit for ICMP packets handled every 10 milliseconds. Default value, 0, applies no limit. This is a configurable value using the set rateControl command.
icmptotportunreachablerx	<Double>	Read-only	ICMP Port Unreachable error messages received. This error is generated when there is no service is running on the port.
icmptotportunreachabletx	<Double>	Read-only	ICMP Port Unreachable error messages generated. This error is generated when there is no service is running on the port.
icmptotneedfragrx	<Double>	Read-only	ICMP Fragmentation Needed error messages received for packets that need to be fragmented but for which Don't Fragment is specified the header.
icmptotthresholdexceeds	<Double>	Read-only	Times the ICMP rate threshold is exceeded. If this counter continuously increases, first make sure the ICMP packets received are genuine. If they are, increase the current rate threshold.
icmptotpktsdropped	<Double>	Read-only	ICMP packets dropped because the rate threshold has been exceeded.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
icmptotnonfirstipfrag	<Double>	Read-only	ICMP Fragmentation Needed error messages received that were generated by an IP fragment other than the first one.
icmptotinvalidbodylen	<Double>	Read-only	ICMP Fragmentation Needed error messages received that specified an invalid body length.
icmptotnotcpconn	<Double>	Read-only	ICMP Need Fragmentation error messages received for TCP packets. The state of the connection for these packets is not maintained on the NetScaler.
icmptotnoudpconn	<Double>	Read-only	ICMP Need Fragmentation error messages received for UDP packets. The state of the connection for these packets is not maintained on the NetScaler.
icmptotinvalidtcpseqno	<Double>	Read-only	ICMP Fragmentation Needed error messages received for packets that contain an invalid TCP address.
icmptotinvalidnextmtuval	<Double>	Read-only	ICMP Fragmentation Needed error messages received in which the Maximum Transmission Unit (MTU) for the next hop is out of range. The range for the MTU is 576-1500.
icmptotbignextmtu	<Double>	Read-only	ICMP Fragmentation Needed error messages received in which the value for the next MTU is higher than that of the current MTU.
icmptotinvalidprotocol	<Double>	Read-only	ICMP Fragmentation Needed error messages received that contain a protocol other than TCP and UDP.
icmptotbadpmtuipchecksum	<Double>	Read-only	ICMP Fragmentation Needed error messages received with an IP checksum error.
icmptotpmtunolink	<Double>	Read-only	ICMP Fragmentation Needed error messages received on a Protocol Control Block (PCB) with no link. The PCB maintains the state of the connection.
icmptotpmtudiscoverydisabled	<Double>	Read-only	ICMP Need Fragmentation error messages received when the PMTU Discovery mode is not enabled.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/protocolicmp

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/protocolicmp?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get protocolicmp resources based on additional properties.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "protocolicmp": [ {
    "icmprxbytesrate":<Double_value>,
    "icmptotpktsdropped":<Double_value>,
    "icmptotrxechoreply":<Double_value>,
    "icmptotthresholdexceeds":<Double_value>,
    "icmptotbadpmtuipchecksum":<Double_value>,
    "icmprxechoreplyrate":<Double_value>,
    "icmptxpktsrate":<Double_value>,
    "icmptotbadchecksum":<Double_value>,
    "icmptotnonfirstipfrag":<Double_value>,
    "icmptotinvalidbodylen":<Double_value>,
    "icmptottxechoreply":<Double_value>,
    "icmptotpmtunolink":<Double_value>,
    "icmptotinvalidprotocol":<Double_value>,
    "icmptottxpkts":<Double_value>,
    "icmptotinvalidnextmtuval":<Double_value>,
    "icmptotportunreachablerx":<Double_value>,
    "icmptotneedfragrx":<Double_value>,
    "icmptotrxecho":<Double_value>,
    "icmptotdstiplookup":<Double_value>,
    "icmprxechorate":<Double_value>,
    "icmptotbignextmtu":<Double_value>,
    "icmptxechoreplyrate":<Double_value>,
    "icmptotpmtudiscoverydisabled":<Double_value>,
    "icmprxpktsrate":<Double_value>,
    "icmptotnoudpconn":<Double_value>,
    "icmptottxbytes":<Double_value>,
    "icmptotrxbytes":<Double_value>,
    "icmptotrxpkts":<Double_value>,
    "icmptotportunreachabletx":<Double_value>,
    "icmptotnotcpconn":<Double_value>,
    "icmptotinvalidtcpseqno":<Double_value>,
    "icmpcurratethreshold":<Double_value>,
    "icmptxbytesrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

protocolicmpv6

May 20, 2016

Statistics for icmpv6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
icmpv6totrxpkts	<Double>	Read-only	ICMPv6 packets received.
icmpv6rxpktsrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxpkts
icmpv6totrxbytes	<Double>	Read-only	Bytes of ICMPv6 data received.
icmpv6rxbytesrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxbytes
icmpv6tottxpkts	<Double>	Read-only	ICMPv6 packets transmitted.
icmpv6txpktsrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxpkts
icmpv6tottxbytes	<Double>	Read-only	Bytes of ICMPv6 data transmitted.
icmpv6txbytesrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxbytes
icmpv6totrxna	<Double>	Read-only	ICMPv6 neighbor advertisement packets received. These packets are received in response to a neighbor solicitation message sent out by this node, or if the link layer address of a neighbor has changed.
icmpv6rxnarate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxna
icmpv6totrxns	<Double>	Read-only	ICMPv6 neighbor solicitation packets received. These packets are received if the link layer address of a neighbor has changed, or in response to a neighbor solicitation message sent out by this node.
icmpv6rxnsrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxns
icmpv6totrxra	<Double>	Read-only	ICMPv6 router advertisement packets received. These are received at defined intervals or in response to a router solicitation message.
icmpv6rxrarate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxra
icmpv6totrxrs	<Double>	Read-only	ICMPv6 router solicitation packets received. These could be sent by a neighboring router to initiate address resolution.
icmpv6rxrsrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxrs
icmpv6totrxechoreq	<Double>	Read-only	ICMPv6 Ping Echo Request packets received.
icmpv6rxechoreqrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxechoreq
icmpv6totrxechoreply	<Double>	Read-only	ICMPv6 Ping Echo Reply packets received.
icmpv6rxechoreplyrate	<Double>	Read-only	Rate (/s) counter for icmpv6totrxechoreply
icmpv6tottxna	<Double>	Read-only	ICMPv6 neighbor advertisement packets transmitted. These packets are sent in response to a neighbor solicitation packet, or if the link layer address of this node has changed.
icmpv6txnarate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxna

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
icmpv6tottxns	<Double>	Read-only	are reachable.
icmpv6txnsrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxns
icmpv6tottxra	<Double>	Read-only	ICMPv6 router advertisement packets transmitted. These packets are sent at regular intervals or in response to a router solicitation packet from a neighbor.
icmpv6txrarate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxra
icmpv6tottxrs	<Double>	Read-only	ICMPv6 router solicitation packets transmitted. These packets are sent to request neighboring routers to generate router advertisements immediately rather than wait for the next defined time.
icmpv6txrsrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxrs
icmpv6tottxechoreq	<Double>	Read-only	ICMPv6 Ping Echo Request packets transmitted.
icmpv6txechoreqrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxechoreq
icmpv6tottxechoreply	<Double>	Read-only	ICMP Ping Echo Reply packets transmitted.
icmpv6txechoreplyrate	<Double>	Read-only	Rate (/s) counter for icmpv6tottxechoreply
icmpv6errra	<Double>	Read-only	ICMPv6 router advertisement error packets received that contain an error in the header, such as an incorrect source IP address, destination IP address, or packet length.
icmpv6errna	<Double>	Read-only	ICMPv6 neighbor advertisement error packets received that contain an error in the header, such as an incorrect source IP address, destination IP address, or packet length.
icmpv6errns	<Double>	Read-only	ICMPv6 neighbor solicitation error packets received that contain an error in the header, such as an incorrect source IP address, destination IP address, or packet length.
icmpv6badchecksums	<Double>	Read-only	Packets received with an ICMPv6 checksum error.
icmpv6unspt	<Double>	Read-only	ICMPv6 packets received that are not supported by the NetScaler.
icmpv6rtthsld	<Double>	Read-only	Packets dropped because the default threshold of 100 requests per 10 milliseconds has been exceeded. This is a configurable value using the set rateControl command.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/protocolicmpv6

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/protocolicmpv6?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get protocolicmpv6 resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "protocolicmpv6": [ {
    "icmpv6tottxns":<Double_value>,
    "icmpv6txpktsrate":<Double_value>,
    "icmpv6unspt":<Double_value>,
    "icmpv6totrxechoreply":<Double_value>,
    "icmpv6tottxra":<Double_value>,
    "icmpv6tottxechoreply":<Double_value>,
    "icmpv6badchecksums":<Double_value>,
    "icmpv6txnarate":<Double_value>,
    "icmpv6txrsrate":<Double_value>,
    "icmpv6tottxna":<Double_value>,
    "icmpv6totrxra":<Double_value>,
    "icmpv6tottxrs":<Double_value>,
    "icmpv6rxpktsrate":<Double_value>,
    "icmpv6totrxpkts":<Double_value>,
    "icmpv6totrxns":<Double_value>,
    "icmpv6rxechoreplyrate":<Double_value>,
    "icmpv6txechoreqrate":<Double_value>,
    "icmpv6rxrsrate":<Double_value>,
    "icmpv6errna":<Double_value>,
    "icmpv6rxnsrate":<Double_value>,
    "icmpv6rxbytesrate":<Double_value>,
    "icmpv6rtthsld":<Double_value>,
    "icmpv6totrxrs":<Double_value>,
    "icmpv6totrxna":<Double_value>,
    "icmpv6txechoreplyrate":<Double_value>,
    "icmpv6totrxbytes":<Double_value>,
    "icmpv6txrarate":<Double_value>,
    "icmpv6errns":<Double_value>,
    "icmpv6totrxechoreq":<Double_value>,
    "icmpv6tottxechoreq":<Double_value>,
    "icmpv6rxrarate":<Double_value>,
    "icmpv6rxechoreqrate":<Double_value>,
    "icmpv6tottxpkts":<Double_value>,
    "icmpv6tottxbytes":<Double_value>,
    "icmpv6rxnarate":<Double_value>,
    "icmpv6txnsrate":<Double_value>,
    "icmpv6errra":<Double_value>,
    "icmpv6txbytesrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

protocolip

May 20, 2016

Statistics for protocolip resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
iptotrxpkts	<Double>	Read-only	IP packets received.
iprxpktsrate	<Double>	Read-only	Rate (/s) counter for iptotrxpkts
iptotrxbytes	<Double>	Read-only	Bytes of IP data received.
iprxbytesrate	<Double>	Read-only	Rate (/s) counter for iptotrxbytes
iptottxpkts	<Double>	Read-only	IP packets transmitted.
iptxpktsrate	<Double>	Read-only	Rate (/s) counter for iptottxpkts
iptottxbytes	<Double>	Read-only	Bytes of IP data transmitted.
iptxbytesrate	<Double>	Read-only	Rate (/s) counter for iptottxbytes
iptotrxmbits	<Double>	Read-only	Megabits of IP data received.
iprxmbitsrate	<Double>	Read-only	Rate (/s) counter for iptotrxmbits
iptottxmbits	<Double>	Read-only	Megabits of IP data transmitted.
iptxmbitsrate	<Double>	Read-only	Rate (/s) counter for iptottxmbits
iptotroutedpkts	<Double>	Read-only	Total routed packets.
iproutedpktsrate	<Double>	Read-only	Rate (/s) counter for iptotroutedpkts
iptotroutedmbits	<Double>	Read-only	Total routed Mbits
iproutedmbitsrate	<Double>	Read-only	Rate (/s) counter for iptotroutedmbits
iptotfragments	<Double>	Read-only	IP fragments received.
iptotsuccreassembly	<Double>	Read-only	Fragmented IP packets successfully reassembled on the NetScaler.
iptotreassemblyattempt	<Double>	Read-only	IP packets that the NetScaler attempts to reassemble. If one of the fragments is missing, the whole packet is dropped.
iptotaddrlookup	<Double>	Read-only	IP address lookups performed by the NetScaler. When a packet is received on a non-established session, the NetScaler checks if the destination IP address is one of the NetScaler owned IP addresses.
iptotaddrlookupfail	<Double>	Read-only	IP address lookups performed by the NetScaler that have failed because the destination IP address of the packet does not match any of the NetScaler owned IP addresses.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
iptottcpfragmentsfwd	<Double>	Read-only	TCP fragments forwarded to the client or the server.
iptotfragpktsgen	<Double>	Read-only	Fragmented packets created by the NetScaler.
iptotbadchecksums	<Double>	Read-only	Packets received with an IP checksum error.
iptotunsuccreassembly	<Double>	Read-only	Packets received that could not be reassembled. This can occur when there is a checksum failure, an identification field mismatch, or when one of the fragments is missing.
iptottoobig	<Double>	Read-only	Packets received for which the reassembled data exceeds the Ethernet packet data length of 1500 bytes.
iptotzerofragmentlen	<Double>	Read-only	Packets received with a fragment length of 0 bytes.
iptotdupfragments	<Double>	Read-only	Duplicate IP fragments received. This can occur when the acknowledgement was not received within the expected time.
iptotoutoforderfrag	<Double>	Read-only	Fragments received that are out of order.
iptotunknowndstrcvd	<Double>	Read-only	Packets received in which the destination IP address was not reachable or not owned by the NetScaler.
iptotbadtransport	<Double>	Read-only	Packets received in which the protocol specified in the IP header is unknown to the NetScaler.
iptotvipdown	<Double>	Read-only	Packets received for which the VIP is down. This can occur when all the services bound to the VIP are down or the VIP is manually disabled.
iptotfixheaderfail	<Double>	Read-only	Packets received that contain an error in one or more components of the IP header.
iptotttlexpired	<Double>	Read-only	Packets for which the time-to-live (TTL) expired during transit. These packets are dropped.
iptotmaxclients	<Double>	Read-only	Attempts to open a new connection to a service for which the maximum limit has been exceeded. Default value, 0, applies no limit.
iptotunknownsvcs	<Double>	Read-only	Packets received on a port or service that is not configured.
iptotlandattacks	<Double>	Read-only	Land-attack packets received. The source and destination addresses are the same.
iptotinvalidheadersz	<Double>	Read-only	Packets received in which an invalid data length is specified, or the value in the length field and the actual data length do not match. The range for the Ethernet packet data length is 0-1500 bytes.
iptotinvalidpacketsize	<Double>	Read-only	Total number of packets received by NetScaler with invalid IP packet size.
iptottruncatedpackets	<Double>	Read-only	Truncated IP packets received. An overflow in the routers along the path can truncate IP packets.
noniptottruncatedpackets	<Double>	Read-only	Truncated non-IP packets received.
iptotzeronexthop	<Double>	Read-only	Packets received that contain a 0 value in the next hop field. These packets are dropped.
iptotbadlens	<Double>	Read-only	Packets received with a length greater than the normal maximum transmission unit of 1514 bytes.
iptotbadmacaddrs	<Double>	Read-only	IP packets transmitted with a bad MAC address.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/protocolip

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/protocolip?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get protocolip resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "protocolip": [ {
    "iproutedpktsrate":<Double_value>,
    "iptotoutoforderfrag":<Double_value>,
    "noniptottruncatedpackets":<Double_value>,
    "iptotmaxclients":<Double_value>,
    "iptotzerofragmentlen":<Double_value>,
    "iptxbytesrate":<Double_value>,
    "iptotvipdown":<Double_value>,
    "iptotroutedpkts":<Double_value>,
    "iprxmbitsrate":<Double_value>,
    "iprxpktsrate":<Double_value>,
    "iptxmbitsrate":<Double_value>,
    "iptotroutedmbits":<Double_value>,
    "iptotinvalidheadersz":<Double_value>,
    "iptotaddrlookupfail":<Double_value>,
    "iptotbadtransport":<Double_value>,
    "iptotfragpktsgen":<Double_value>,
    "iptotsuccreassembly":<Double_value>,
    "iptotrxmbits":<Double_value>,
    "iptottxbytes":<Double_value>,
    "iptotbadmacaddrs":<Double_value>,
    "iptottcpfragmentsfwd":<Double_value>,
    "iptotrxpkts":<Double_value>,
    "iptotrxbytes":<Double_value>,
    "iptotlandattacks":<Double_value>,
    "iptotunknowndstrcvd":<Double_value>,
    "iptottruncatedpackets":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "iptotttlexpired":<Double_value>,
    "iprxbytesrate":<Double_value>,
    "iptottxmbits":<Double_value>,
    "iptotbadlens":<Double_value>,
    "iptotunknownsvcs":<Double_value>,
    "iptotdupfragments":<Double_value>,
    "iptottoobig":<Double_value>,
    "iptotzeronexthop":<Double_value>,
    "iptotaddrlookup":<Double_value>,
    "iptotfragments":<Double_value>,
    "iptotinvalidpacketsize":<Double_value>,
    "iptotunsuccreassembly":<Double_value>,
    "iptotreassemblyattempt":<Double_value>,
    "iptottxpkts":<Double_value>,
    "iptxpktsrate":<Double_value>,
    "iptotbadchecksums":<Double_value>,
    "iptotfixheaderfail":<Double_value>,
    "iproutedmbitsrate":<Double_value>,
    "iptotudpfragmentsfwd":<Double_value>
}}}
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

protocolipv6

May 20, 2016

Statistics for ip v6 resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
ipv6totrxpkts	<Double>	Read-only	IPv6 packets received.
ipv6rxpktsrate	<Double>	Read-only	Rate (/s) counter for ipv6totrxpkts
ipv6totrxbytes	<Double>	Read-only	Bytes of IPv6 data received.
ipv6rxbytesrate	<Double>	Read-only	Rate (/s) counter for ipv6totrxbytes
ipv6tottxpkts	<Double>	Read-only	IPv6 packets transmitted
ipv6txpktsrate	<Double>	Read-only	Rate (/s) counter for ipv6tottxpkts
ipv6tottxbytes	<Double>	Read-only	Bytes of IPv6 data transmitted.
ipv6txbytesrate	<Double>	Read-only	Rate (/s) counter for ipv6tottxbytes
ipv6totroutepkts	<Double>	Read-only	IPv6 packets routed.
ipv6routepktsrate	<Double>	Read-only	Rate (/s) counter for ipv6totroutepkts
ipv6totroutembits	<Double>	Read-only	IPv6 total Mbits.
ipv6routembitsrate	<Double>	Read-only	Rate (/s) counter for ipv6totroutembits
ipv6fragtotrxpkts	<Double>	Read-only	IPv6 fragments received.
ipv6fragrxpktsrate	<Double>	Read-only	Rate (/s) counter for ipv6fragtotrxpkts
ipv6fragtottcpreassembled	<Double>	Read-only	TCP fragments processed after reassembly.
ipv6fragtcpreassembledrate	<Double>	Read-only	Rate (/s) counter for ipv6fragtottcpreassembled
ipv6fragtotudpreassembled	<Double>	Read-only	UDP fragments processed after reassembly.
ipv6fragudpreassembledrate	<Double>	Read-only	Rate (/s) counter for ipv6fragtotudpreassembled
ipv6fragtotpktsprocessnoreass	<Double>	Read-only	IPv6 fragments processed without reassembly.
ipv6fragpktsprocessnoreassrate	<Double>	Read-only	Rate (/s) counter for ipv6fragtotpktsprocessnoreass
ipv6fragtotpktsforward	<Double>	Read-only	IPv6 fragments forwarded to the client or server without reassembly.
ipv6fragpktsforwardrate	<Double>	Read-only	Rate (/s) counter for ipv6fragtotpktsforward

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ipv6errhdr	<Double>	Read-only	Packets received that contain an error in one or more components of the IPv6 header.
ipv6nextheadererr	<Double>	Read-only	Packets received that contain an unsupported next header. The supported next headers are TCP, ICMP, UDP, OSPF, and FRAGMENT.
ipv6landattack	<Double>	Read-only	Land-attack packets received. The source and destination addresses are the same. If not dropped, these packets can lock up the appliance.
ipv6fragpkttoobig	<Double>	Read-only	Packets received for which the reassembled data exceeds the Ethernet packet data length of 1500 bytes.
ipv6fragzerolenpkt	<Double>	Read-only	Packets received with a fragment length of 0 bytes.
ipv6toticmpnarx	<Double>	Read-only	Number of ICMPv6 NA packets received by NetScaler (OBSOLETE).
ipv6icmpnarxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmpnarx
ipv6toticmpnsrx	<Double>	Read-only	Number of ICMPv6 NS packets received by NetScaler (OBSOLETE).
ipv6icmpnsrxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmpnsrx
ipv6toticmpnatx	<Double>	Read-only	Number of ICMPv6 NA packets transmitted by NetScaler (OBSOLETE).
ipv6icmpnatxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmpnatx
ipv6toticmpnstx	<Double>	Read-only	Number of ICMPv6 NS packets transmitted by NetScaler (OBSOLETE).
ipv6icmpnstxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmpnstx
ipv6toticmprarx	<Double>	Read-only	Number of ICMPv6 RA packets received by NetScaler (OBSOLETE).
ipv6icmprarxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmprarx
ipv6toticmprstx	<Double>	Read-only	Number of ICMPv6 RS packets transmitted by NetScaler (OBSOLETE).
ipv6icmprstxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmprstx
ipv6toticmprx	<Double>	Read-only	Number of ICMPv6 packets received by NetScaler (OBSOLETE).
ipv6icmprxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmprx
ipv6toticmptx	<Double>	Read-only	Number of ICMPv6 packets transmitted by NetScaler (OBSOLETE).
ipv6icmptxrate	<Double>	Read-only	Rate (/s) counter for ipv6toticmptx
ipv6errrxhdr	<Double>	Read-only	Number of erroneous header packets received (OBSOLETE).
ipv6errrxpkt	<Double>	Read-only	Number of erroneous packets received (OBSOLETE).
ipv6badcksum	<Double>	Read-only	Number of bad checksum packets received (OBSOLETE).
icmpv6error	<Double>	Read-only	Number of erroneous ICMPv6 packets received (OBSOLETE).
icmpv6unspt	<Double>	Read-only	Number of ICMPv6 unsupported packets received (OBSOLETE).
icmpv6rtthsld	<Double>	Read-only	Number of ICMPv6 packets dropped for rate threshold exceeded (OBSOLETE).

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/protocolipv6

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/protocolipv6?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get protocolipv6 resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "protocolipv6": [ {
    "ipv6fragpktsforwardrate":<Double_value>,
    "ipv6icmprstxrate":<Double_value>,
    "ipv6fragrxpktsrate":<Double_value>,
    "ipv6totroutepkts":<Double_value>,
    "ipv6fragpkttoobig":<Double_value>,
    "ipv6toticmpnarx":<Double_value>,
    "ipv6fragtotrxpkts":<Double_value>,
    "ipv6icmpnatxrate":<Double_value>,
    "ipv6toticmpnsrx":<Double_value>,
    "ipv6routembitsrate":<Double_value>,
    "ipv6txpktsrate":<Double_value>,
    "icmpv6rtthsld":<Double_value>,
    "icmpv6unspt":<Double_value>,
    "ipv6badcksum":<Double_value>,
    "ipv6errrxhdr":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ipv6totrxbytes":<Double_value>,
    "ipv6rxpktsrate":<Double_value>,
    "ipv6icmpnstxrate":<Double_value>,
    "ipv6icmprarxrate":<Double_value>,
    "ipv6errhdr":<Double_value>,
    "ipv6totrxpkts":<Double_value>,
    "ipv6toticmprx":<Double_value>,
    "ipv6fragtotpktsprocessnoreass":<Double_value>,
    "ipv6rxbytesrate":<Double_value>,
    "ipv6icmpnsrxrate":<Double_value>,
    "ipv6toticmpnatx":<Double_value>,
    "ipv6routepktsrate":<Double_value>,
    "ipv6icmptxrate":<Double_value>,
    "ipv6toticmprstx":<Double_value>,
    "ipv6totroutembits":<Double_value>,
    "ipv6icmprxrate":<Double_value>,
    "ipv6fragpktsprocessnoreassrate":<Double_value>,
    "ipv6fragtcpreassembledrate":<Double_value>,
    "ipv6tottxpkts":<Double_value>,
    "ipv6icmpnarxrate":<Double_value>,
    "ipv6fragtotudpreassembled":<Double_value>,
    "ipv6fragtottcpreassembled":<Double_value>,
    "ipv6errrxpkt":<Double_value>,
    "ipv6nextheadererr":<Double_value>,
    "ipv6toticmptx":<Double_value>,
    "ipv6toticmpnstx":<Double_value>,
    "ipv6fragudpreassembledrate":<Double_value>,
    "ipv6txbytesrate":<Double_value>,
    "icmpv6error":<Double_value>,
    "ipv6toticmprarx":<Double_value>,
    "ipv6tottxbytes":<Double_value>,
    "ipv6landattack":<Double_value>,
    "ipv6fragtotpktsforward":<Double_value>,
    "ipv6fragzerolenpkt":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

protocoltcp

May 20, 2016

Statistics for tcp protocol resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
tcpactiveserverconn	<Double>	Read-only	Connections to a server currently responding to requests.
tcpcurserverconnopening	<Double>	Read-only	Server connections in the Opening state, which indicates that the handshakes are not yet complete.
tcpcurclientconnopening	<Double>	Read-only	Client connections in the Opening state, which indicates that the handshakes are not yet complete.
tcpcurclientconnestablished	<Double>	Read-only	Current client connections in the Established state, which indicates that data transfer can occur between the NetScaler and the client.
tcpcurserverconnestablished	<Double>	Read-only	Current server connections in the Established state, which indicates that data transfer can occur between the NetScaler and the server.
tcptotrxpkts	<Double>	Read-only	TCP packets received.
tcprxpktsrate	<Double>	Read-only	Rate (/s) counter for tcptotrxpkts
tcptotrxbytes	<Double>	Read-only	Bytes of TCP data received.
tcprxbytesrate	<Double>	Read-only	Rate (/s) counter for tcptotrxbytes
tcptottxpkts	<Double>	Read-only	TCP packets transmitted.
tcptxpktsrate	<Double>	Read-only	Rate (/s) counter for tcptottxpkts
tcptottxbytes	<Double>	Read-only	Bytes of TCP data transmitted.
tcptxbytesrate	<Double>	Read-only	Rate (/s) counter for tcptottxbytes
tcpcurclientconn	<Double>	Read-only	Client connections, including connections in the Opening, Established, and Closing state.
tcpcurclientconnclosing	<Double>	Read-only	Client connections in the Closing state, which indicates that the connection termination process has initiated but is not complete.
tcptotclientconnopened	<Double>	Read-only	Client connections opened by the NetScaler since startup (after three-way handshake). This counter is reset when the NetScaler is restarted.
tcpclientconnopenedrate	<Double>	Read-only	Rate (/s) counter for tcptotclientconnopened
tcpcurserverconn	<Double>	Read-only	Server connections, including connections in the Opening, Established, and Closing state.
tcpcurserverconnclosing	<Double>	Read-only	Server connections in the Closing state, which indicates that the connection termination process has initiated but is not complete.
tcptotserverconnopened	<Double>	Read-only	Server connections initiated by the NetScaler since startup. This counter is reset when the NetScaler is restarted.
tcpserverconnopenedrate	<Double>	Read-only	Rate (/s) counter for tcptotserverconnopened

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
			connections have been reached, the NetScaler queues these requests.
tcpspareconn	<Double>	Read-only	Spare connections available. To save time and resources in establishing another connection for a new client, the connection on the server is not closed after completing the request from the first client and is available for serving future requests.
tcptotzombiecltconnflushed	<Double>	Read-only	Client connections that are flushed because the client has been idle for some time.
tcpzombiecltconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiecltconnflushed
tcptotzombiehalfopencltconnflushed	<Double>	Read-only	Half-opened client connections that are flushed because the three-way handshakes are not complete.
tcpzombiehalfopencltconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiehalfopencltconnflushed
tcptotzombieactivehalfclosecltconnflushed	<Double>	Read-only	Active half-closed client connections that are flushed because the client has closed the connection and there has been no activity on the connection.
tcpzombieactivehalfclosecltconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombieactivehalfclosecltconnflushed
tcptotzombiepassivehalfclosecltconnflushed	<Double>	Read-only	Passive half-closed client connections that are flushed because the NetScaler has closed the connection and there has been no activity on the connection.
tcpzombiepassivehalfclosecltconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiepassivehalfclosecltconnflushed
tcptotzombiesvrconnflushed	<Double>	Read-only	Server connections that are flushed because there have been no client requests in the queue for some time.
tcpzombiesvrconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiesvrconnflushed
tcptotzombiehalfopensvrconnflushed	<Double>	Read-only	Half-opened server connections that are flushed because the three-way handshakes are not complete.
tcpzombiehalfopensvrconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiehalfopensvrconnflushed
tcptotzombieactivehalfclosesvrconnflushed	<Double>	Read-only	Active half-closed server connections that are flushed because the server has closed the connection and there has been no activity on the connection.
tcpzombieactivehalfclosesvrconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombieactivehalfclosesvrconnflushed
tcptotzombiepassivehalfclosesrvconnflushed	<Double>	Read-only	Passive half-closed server connections that are flushed because the NetScaler has closed the connection and there has been no activity on the connection.
tcpzombiepassivehalfclosesrvconnflushedrate	<Double>	Read-only	Rate (/s) counter for tcptotzombiepassivehalfclosesrvconnflushed
pcbtotzombiecall	<Double>	Read-only	Times the Zombie cleanup function is called. Every time a connection is flushed, it is marked for cleanup. The Zombie cleanup function clears all these connections at predefined intervals.
pcbzombiecallrate	<Double>	Read-only	Rate (/s) counter for pcbtotzombiecall
tcptotsyn	<Double>	Read-only	SYN packets received
tcpsynrate	<Double>	Read-only	Rate (/s) counter for tcptotsyn
tcptotsynprobe	<Double>	Read-only	Probes from the NetScaler to a server. The NetScaler sends a SYN packet to the server to check its availability and expects a SYN_ACK packet from the server before a specified response timeout.
tcpsynproberate	<Double>	Read-only	Rate (/s) counter for tcptotsynprobe
tcptotsvrfin	<Double>	Read-only	FIN packets received from the server.
tcpsvrfinrate	<Double>	Read-only	Rate (/s) counter for tcptotsvrfin
tcptotcltfin	<Double>	Read-only	FIN packets received from the clients.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
tcpwaittosyn	<Double>	Read-only	SYN packets (packets used to initiate a TCP connection) received on connections that are in the TIME_WAIT state. Packets cannot be transferred on a connection in this state.
tcpwaittosynrate	<Double>	Read-only	Rate (/s) counter for tcpwaittosyn
tcpwaittodata	<Double>	Read-only	Bytes of data received on connections that are in the TIME_WAIT state. Data cannot be transferred on a connection that is in this state.
tcpwaittodatarate	<Double>	Read-only	Rate (/s) counter for tcpwaittodata
tcptotsynheld	<Double>	Read-only	SYN packets held on the NetScaler that are waiting for a server connection.
tcpsynheldrate	<Double>	Read-only	Rate (/s) counter for tcptotsynheld
tcptotsynflush	<Double>	Read-only	SYN packets flushed on the NetScaler because of no response from the server for three or more seconds.
tcpsynflushrate	<Double>	Read-only	Rate (/s) counter for tcptotsynflush
tcptotfinwaitclosed	<Double>	Read-only	Connections closed on the NetScaler because the number of connections in the TIME_WAIT state has exceeded the default value of 7000.
tcpfinwaitclosedrate	<Double>	Read-only	Rate (/s) counter for tcptotfinwaitclosed
tcperrbadchecksum	<Double>	Read-only	Packets received with a TCP checksum error.
tcperrbadchecksumrate	<Double>	Read-only	Rate (/s) counter for tcperrbadchecksum
tcperrdataafterfin	<Double>	Read-only	Packets received following a connection termination request. This error is usually caused by a reordering of packets during transmission.
tcperrdataafterfinrate	<Double>	Read-only	Rate (/s) counter for tcperrdataafterfin
tcperrsyninsynrcvd	<Double>	Read-only	SYN packets received on a connection that is in the SYN_RCVD state. A connection goes into the SYN_RCVD state after receiving a SYN packet.
tcperrsyninsynrcvdrate	<Double>	Read-only	Rate (/s) counter for tcperrsyninsynrcvd
tcperrsyninest	<Double>	Read-only	SYN packets received on a connection that is in the ESTABLISHED state. A SYN packet is not expected on an ESTABLISHED connection.
tcperrsyninestrate	<Double>	Read-only	Rate (/s) counter for tcperrsyninest
tcperrsynsentbadack	<Double>	Read-only	Incorrect ACK packets received on a connection that is in the SYN_SENT state. An incorrect ACK packet is the third packet in the three-way handshake that has an incorrect sequence number.
tcperrsynsentbadackrate	<Double>	Read-only	Rate (/s) counter for tcperrsynsentbadack
tcperrrst	<Double>	Read-only	Reset packets received from a client or a server.
tcperrrstrate	<Double>	Read-only	Rate (/s) counter for tcperrrst
tcperrrstnonest	<Double>	Read-only	Reset packets received on a connection that is not in the ESTABLISHED state.
tcperrrstnonestrate	<Double>	Read-only	Rate (/s) counter for tcperrrstnonest
tcperrrstoutofwindow	<Double>	Read-only	Reset packets received on a connection that is out of the current TCP window.
tcperrrstoutofwindowrate	<Double>	Read-only	Rate (/s) counter for tcperrrstoutofwindow
tcperrrstintimewait	<Double>	Read-only	Reset packets received on a connection that is in the TIME_WAIT state. Packets cannot be transferred on a connection in the TIME_WAIT state.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
tcperrrstintimewaitrate	<Double>	Read-only	Rate (/s) counter for tcperrrstintimewait
tcperrsvroutoforder	<Double>	Read-only	Out of order TCP packets received from a server.
tcperrsvroutoforderrate	<Double>	Read-only	Rate (/s) counter for tcperrsvroutoforder
tcperrcltoutoforder	<Double>	Read-only	Out of order TCP packets received from a client.
tcperrcltoutoforderrate	<Double>	Read-only	Rate (/s) counter for tcperrcltoutoforder
tcperrclthole	<Double>	Read-only	TCP holes created on a client connection. When out of order packets are received from a client, a hole is created on the NetScaler for each group of missing packets.
tcperrcltholerate	<Double>	Read-only	Rate (/s) counter for tcperrclthole
tcperrsvrhole	<Double>	Read-only	TCP holes created on a server connection. When out of order packets are received from a server, a hole is created on the NetScaler for each group of missing packets.
tcperrsvrholerate	<Double>	Read-only	Rate (/s) counter for tcperrsvrhole
tcperrcookiepktseqreject	<Double>	Read-only	SYN cookie packets rejected because they contain an incorrect sequence number.
tcperrcookiepktseqrejectrate	<Double>	Read-only	Rate (/s) counter for tcperrcookiepktseqreject
tcperrcookiepktsigreject	<Double>	Read-only	SYN cookie packets rejected because they contain an incorrect signature.
tcperrcookiepktsigrejectrate	<Double>	Read-only	Rate (/s) counter for tcperrcookiepktsigreject
tcperrcookiepktseqdrop	<Double>	Read-only	SYN cookie packets dropped because the sequence number specified in the packets is outside the current window.
tcperrcookiepktseqdroprate	<Double>	Read-only	Rate (/s) counter for tcperrcookiepktseqdrop
tcperrcookiepktmssreject	<Double>	Read-only	SYN cookie packets rejected because the maximum segment size (MSS) specified in the packets is incorrect.
tcperrcookiepktmssrejectrate	<Double>	Read-only	Rate (/s) counter for tcperrcookiepktmssreject
tcperranyportfail	<Double>	Read-only	Port allocations that have failed on a mapped IP address because the maximum limit of 65536 has been exceeded.
tcperranyportfailrate	<Double>	Read-only	Rate (/s) counter for tcperranyportfail
tcperripportfail	<Double>	Read-only	Port allocations that have failed on a subnet IP address or vserver IP address because the maximum limit of 65536 has been exceeded.
tcperripportfailrate	<Double>	Read-only	Rate (/s) counter for tcperripportfail
tcperrstraypkt	<Double>	Read-only	Number of stray or misrouted packets.
tcperrstraypktrate	<Double>	Read-only	Rate (/s) counter for tcperrstraypkt
tcperrsentrst	<Double>	Read-only	Reset packets sent to a client or a server.
tcperrsentrstrate	<Double>	Read-only	Rate (/s) counter for tcperrsentrst
tcperrbadstateconn	<Double>	Read-only	Connections that are not in a valid TCP state.
tcperrbadstateconnrate	<Double>	Read-only	Rate (/s) counter for tcperrbadstateconn
			Reset packets dropped because the default threshold of 100 resets per 10 milliseconds has been exceeded. This is a

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
tcperrstthresholdrate	<Double>	Read-only	Rate (/s) counter for tcperrrstthreshold
tcperroutofwindowpkts	<Double>	Read-only	Packets received that are out of the current advertised window.
tcperroutofwindowpktsrate	<Double>	Read-only	Rate (/s) counter for tcperroutofwindowpkts
tcperrsyndroppedcongestion	<Double>	Read-only	SYN packets dropped because of network congestion.
tcperrsyndroppedcongestionrate	<Double>	Read-only	Rate (/s) counter for tcperrsyndroppedcongestion
tcperrcltretrasmit	<Double>	Read-only	Packets retransmitted by a client. This usually occurs because the acknowledgement from the NetScaler has not reached the client.
tcperrcltretrasmitrate	<Double>	Read-only	Rate (/s) counter for tcperrcltretrasmit
tcperrfullretrasmit	<Double>	Read-only	Full packets retransmitted by the client or the server.
tcperrfullretrasmitrate	<Double>	Read-only	Rate (/s) counter for tcperrfullretrasmit
tcperrsynretry	<Double>	Read-only	SYN packets resent to a server.
tcperrsynretryrate	<Double>	Read-only	Rate (/s) counter for tcperrsynretry
tcperrsyngiveup	<Double>	Read-only	Attempts to establish a connection on the NetScaler that timed out.
tcperrsyngiveuprate	<Double>	Read-only	Rate (/s) counter for tcperrsyngiveup
tcperrretransmit	<Double>	Read-only	TCP packets retransmitted. The NetScaler attempts to retransmit the packet up to seven times, after which it resets the other half of the TCP connection.
tcperrretransmitrate	<Double>	Read-only	Rate (/s) counter for tcperrretransmit
tcperrfirstretransmissions	<Double>	Read-only	Packets retransmitted once by the NetScaler.
tcperrfirstretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrfirstretransmissions
tcperrthirdretransmissions	<Double>	Read-only	Packets retransmitted three times by the NetScaler.
tcperrthirdretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrthirdretransmissions
tcperrfifthretransmissions	<Double>	Read-only	Packets retransmitted five times by the NetScaler.
tcperrfifthretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrfifthretransmissions
tcperrseventhretransmissions	<Double>	Read-only	Packets retransmitted seven times by the NetScaler. If this fails, the NetScaler terminates the connection.
tcperrseventhretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrseventhretransmissions
tcperrfastretransmissions	<Double>	Read-only	TCP packets on which the NetScaler performs a fast retransmission in response to three duplicate acknowledgements or a partial acknowledgement. The NetScaler assumes that the packet is lost and retransmits the packet before its time-out.
tcperrfastretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrfastretransmissions
tcperrsvrretrasmit	<Double>	Read-only	Packets retransmitted by a server. This usually occurs because the acknowledgement from the NetScaler has not reached the server.
tcperrsvrretrasmitrate	<Double>	Read-only	Rate (/s) counter for tcperrsvrretrasmit

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			window advertised by the NetScaler is not big enough to hold the full packet.
Name	**Data Type**	**Permissions**	**Description**
tcperrpartialretrasmitrate	<Double>	Read-only	Rate (/s) counter for tcperrpartialretrasmit
tcperrfinretry	<Double>	Read-only	FIN packets resent to a server or a client.
tcperrfinretryrate	<Double>	Read-only	Rate (/s) counter for tcperrfinretry
tcperrfingiveup	<Double>	Read-only	Connections that were timed out by the NetScaler because of not receiving the ACK packet after retransmitting the FIN packet four times.
tcperrfingiveuprate	<Double>	Read-only	Rate (/s) counter for tcperrfingiveup
tcperrsecondretransmissions	<Double>	Read-only	Packets retransmitted twice by the NetScaler.
tcperrsecondretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrsecondretransmissions
tcperrforthretransmissions	<Double>	Read-only	Packets retransmitted four times by the NetScaler.
tcperrforthretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrforthretransmissions
tcperrsixthretransmissions	<Double>	Read-only	Packets retransmitted six times by the NetScaler.
tcperrsixthretransmissionsrate	<Double>	Read-only	Rate (/s) counter for tcperrsixthretransmissions
tcperrretransmitgiveup	<Double>	Read-only	Number of times NetScaler terminates a connection after retransmitting the packet seven times on that connection.Retrasnmission happens when recieving end doesn't acknowledges the packet.
tcperrretransmitgiveuprate	<Double>	Read-only	Rate (/s) counter for tcperrretransmitgiveup
tcperrcipalloc	<Double>	Read-only	Number of times TCP level client header insertion failure
tcperrcipallocrate	<Double>	Read-only	Rate (/s) counter for tcperrcipalloc

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/protocoltcp

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/protocoltcp?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get protocoltcp resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "protocoltcp": [ {
    "tcperrfingiveuprate":<Double_value>,
    "tcperrretransmit":<Double_value>,
    "tcperrcookiepktmssreject":<Double_value>,
    "tcptotzombiehalfopencltconnflushed":<Double_value>,
    "tcperrsentrstrate":<Double_value>,
    "tcperrcltoutoforder":<Double_value>,
    "tcpsurgequeuelen":<Double_value>,
    "tcperrcookiepktseqrejectrate":<Double_value>,
    "tcperrrstoutofwindowrate":<Double_value>,
    "tcptotrxpkts":<Double_value>,
    "tcpcurclientconnopening":<Double_value>,
    "tcperrcookiepktseqreject":<Double_value>,
    "tcperrforthretransmissionsrate":<Double_value>,
    "tcperrrst":<Double_value>,
    "tcpsvrfinrate":<Double_value>,
    "tcptxbytesrate":<Double_value>,
    "tcptotsyn":<Double_value>,
    "tcpserverconnopenedrate":<Double_value>,
    "tcperrsyninsynrcvdrate":<Double_value>,
    "tcperrpartialretrasmitrate":<Double_value>,
    "tcpzombieactivehalfclosecltconnflushedrate":<Double_value>,
    "tcperrfinretry":<Double_value>,
    "tcperrsvroutoforderrate":<Double_value>,
    "pcbzombiecallrate":<Double_value>,
    "tcperrfirstretransmissionsrate":<Double_value>,
    "tcperrstraypkt":<Double_value>,
    "tcperrsyninsynrcvd":<Double_value>,
    "tcprxbytesrate":<Double_value>,
    "tcpcurserverconnestablished":<Double_value>,
    "tcperroutofwindowpktsrate":<Double_value>,
    "tcperranyportfailrate":<Double_value>,
    "tcpsynproberate":<Double_value>,
    "tcpzombiesvrconnflushedrate":<Double_value>,
    "tcpcurclientconn":<Double_value>,
    "tcperrsyninestrate":<Double_value>,
    "tcpcurserverconn":<Double_value>,
    "tcperripportfailrate":<Double_value>,
    "tcperrcookiepktsigrejectrate":<Double_value>,
    "tcperrforthretransmissions":<Double_value>,
    "tcpzombiehalfopensvrconnflushedrate":<Double_value>,
    "tcperrsixthretransmissions":<Double_value>,
    "tcpfinwaitclosedrate":<Double_value>,
    "tcperrbadchecksum":<Double_value>,
    "tcpsynflushrate":<Double_value>,
    "tcperrcipallocrate":<Double_value>,
    "tcperrrstthreshold":<Double_value>,
    "tcptotzombieactivehalfclosecltconnflushed":<Double_value>,
    "tcperrsynsentbadackrate":<Double_value>,
    "tcpwaittodatarate":<Double_value>,
    "tcperrseventhretransmissionsrate":<Double_value>,
    "tcperrretransmitrate":<Double_value>,
    "tcperrfullretrasmitrate":<Double_value>,
    "tcptotsynflush":<Double_value>,
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

"tcperrsixthretransmissionsrate":<Double_value>,
"tcperrfastretransmissions":<Double_value>,
"tcperrsecondretransmissionsrate":<Double_value>,
"tcpzombiepassivehalfclosecltconnflushedrate":<Double_value>,
"tcpcltfinrate":<Double_value>,
"tcpsynheldrate":<Double_value>,
"tcperrrstnonestrate":<Double_value>,
"tcperrcltretrasmit":<Double_value>,
"tcptotsynprobe":<Double_value>,
"tcptotfinwaitclosed":<Double_value>,
"tcpzombiepassivehalfclosesrvconnflushedrate":<Double_value>,
"tcperrsynsentbadack":<Double_value>,
"tcptotsynheld":<Double_value>,
"tcperrrstintimewaitrate":<Double_value>,
"tcperrrstintimewait":<Double_value>,
"tcperrthirdretransmissions":<Double_value>,
"tcptotrxbytes":<Double_value>,
"tcperrsvrretrasmit":<Double_value>,
"tcperrfastretransmissionsrate":<Double_value>,
"tcptotzombiepassivehalfclosecltconnflushed":<Double_value>,
"tcperrcltholerate":<Double_value>,
"tcperrsvroutoforder":<Double_value>,
"tcprxpktsrate":<Double_value>,
"tcptotclientconnopened":<Double_value>,
"tcperrstraypktrate":<Double_value>,
"tcperrfinretryrate":<Double_value>,
"tcptotserverconnopened":<Double_value>,
"tcperrretransmitgiveup":<Double_value>,
"tcperrthirdretransmissionsrate":<Double_value>,
"tcpwaittosyn":<Double_value>,
"tcperrsyndroppedcongestion":<Double_value>,
"tcperrfingiveup":<Double_value>,
"tcptotsvrfin":<Double_value>,
"tcperrcltretrasmitrate":<Double_value>,
"tcperrcookiepktmssrejectrate":<Double_value>,
"tcperrsvrretrasmitrate":<Double_value>,
"tcpcurserverconnclosing":<Double_value>,
"tcperrrstrate":<Double_value>,
"tcperrfifthretransmissionsrate":<Double_value>,
"tcperrdataafterfin":<Double_value>,
"tcperrsvrhole":<Double_value>,
"tcperrsyngiveup":<Double_value>,
"tcperrfirstretransmissions":<Double_value>,
"tcptotzombiesvrconnflushed":<Double_value>,
"tcperrsynretryrate":<Double_value>,
"tcperrcookiepktsigreject":<Double_value>,
"tcperroutofwindowpkts":<Double_value>,
"tcperrcookiepktseqdrop":<Double_value>,
"tcperrbadstateconnrate":<Double_value>,
"tcperrsyndroppedcongestionrate":<Double_value>,
"tcperrsvrholerate":<Double_value>,
"tcperrdataafterfinrate":<Double_value>,
"tcperrcltoutoforderrate":<Double_value>,
"tcptotzombiepassivehalfclosesrvconnflushed":<Double_value>,
"tcperrsyninest":<Double_value>,
"tcperrsecondretransmissions":<Double_value>,
"tcperrbadstateconn":<Double_value>,
"tcpspareconn":<Double_value>,
"tcperrretransmitgiveuprate":<Double_value>,
"tcptottxpkts":<Double_value>,
"tcptotcltfin":<Double_value>,
"tcpsynrate":<Double_value>,
"tcpclientconnopenedrate":<Double_value>,
"tcperranyportfail":<Double_value>,
"tcpwaittosynrate":<Double_value>,
"tcperrrstnonest":<Double_value>,
"tcperripportfail":<Double_value>,
"tcperrpartialretrasmit":<Double_value>,
"tcpzombieactivehalfclosesvrconnflushedrate":<Double_value>,
"tcperrcltthole":<Double_value>,
"tcpcurclientconnestablished":<Double_value>,
"tcpwaittodata":<Double_value>,
"tcperrsentrst":<Double_value>,
"tcperrsyngiveuprate":<Double_value>,
"tcperrbadchecksumrate":<Double_value>,
"tcperrcookiepktseqdroprate":<Double_value>,
"tcperrseventhretransmissions":<Double_value>,
"tcperrrstoutofwindow":<Double_value>,
"tcpcurserverconnopening":<Double_value>,
"tcptotzombieactivehalfclosesvrconnflushed":<Double_value>,
"tcperrfullretrasmit":<Double_value>,
"tcperrcipalloc":<Double_value>,
"tcptotzombieclteonnflushed":<Double_value>,
"tcptotzombiehalfopensvrconnflushed":<Double_value>,
"pcbtotzombiecall":<Double_value>,
"tcpactiveserverconn":<Double_value>,
"tcperrsynretry":<Double_value>,
"tcptottxbytes":<Double_value>,
"tcpcurclientconnclosing":<Double_value>,
"tcperrrstthresholdrate":<Double_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "tcperrfifthretransmissions":<Double_value>,
    "tcpzombiehalfopencltconnflushedrate":<Double_value>,
    "tcpzombiecltconnflushedrate":<Double_value>,
    "tcptxpktsrate":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

protocoludp

May 20, 2016

Statistics for UDP Protocol resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
udptotrxpkts	<Double>	Read-only	Total number of UDP packets received.
udprxpktsrate	<Double>	Read-only	Rate (/s) counter for udptotrxpkts
udptotrxbytes	<Double>	Read-only	Total number of UDP data received in bytes.
udprxbytesrate	<Double>	Read-only	Rate (/s) counter for udptotrxbytes
udptottxpkts	<Double>	Read-only	Total number of UDP packets transmitted.
udptxpktsrate	<Double>	Read-only	Rate (/s) counter for udptottxpkts
udptottxbytes	<Double>	Read-only	Total number of UDP data transmitted in bytes.
udptxbytesrate	<Double>	Read-only	Rate (/s) counter for udptottxbytes
udpcurratethreshold	<Double>	Read-only	Limit for UDP packets handled every 10 milliseconds. Default value, 0, applies no limit. This is a configurable value using the set rateControl command.
udptotunknownsvcpkts	<Double>	Read-only	Stray UDP packets dropped due to no configured listening service.
udpbadchecksum	<Double>	Read-only	Packets received with a UDP checksum error.
udpcurratethresholdexceeds	<Double>	Read-only	Number of times the UDP rate threshold is exceeded. If this counter continuously increases, first make sure the UDP packets received are genuine. If they are, increase the current rate threshold. This is a configurable value using the set rateControl command.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

© 1999-2017 Citrix Systems, Inc. All rights reserved.

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/protocoludp

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/protocoludp?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get protocoludp resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "protocoludp": [ {
    "udptxpktsrate":<Double_value>,
    "udpcurratethreshold":<Double_value>,
    "udptotrxpkts":<Double_value>,
    "udptottxpkts":<Double_value>,
    "udptotrxbytes":<Double_value>,
    "udptxbytesrate":<Double_value>,
    "udprxpktsrate":<Double_value>,
    "udpbadchecksum":<Double_value>,
    "udptottxbytes":<Double_value>,
    "udptotunknownsvcpkts":<Double_value>,
    "udprxbytesrate":<Double_value>,
    "udpcurratethresholdexceeds":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Qos

May 20, 2016

QoS statistics.

Name	Description
qos	Statistics for qos.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

qos

Jul 07, 2017

Statistics for qos.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
ipcmessagessent	<Double>	Read-only	IPC messages sent from qos system.
ipcmessagessentrate	<Double>	Read-only	Rate (/s) counter for ipcmessagessent
ipcmessagesfailed	<Double>	Read-only	IPC messages failed to send from qos system.
ipcmessagesfailedrate	<Double>	Read-only	Rate (/s) counter for ipcmessagesfailed
ipcmessagesreceived	<Double>	Read-only	IPC messages received by qos.
ipcmessagesreceivedrate	<Double>	Read-only	Rate (/s) counter for ipcmessagesreceived
ipcpe2qossent	<Double>	Read-only	IPC messages sent to qos system.
ipcpe2qossentrate	<Double>	Read-only	Rate (/s) counter for ipcpe2qossent
ipcpe2qosfailed	<Double>	Read-only	IPC messages failed to send to qos system.
ipcpe2qosfailedrate	<Double>	Read-only	Rate (/s) counter for ipcpe2qosfailed
ipcpe2qosreceived	<Double>	Read-only	IPC messages received from qos system.
ipcpe2qosreceivedrate	<Double>	Read-only	Rate (/s) counter for ipcpe2qosreceived
qosbytesdropped	<Double>	Read-only	Bytes QoS marked for drop
qosbytesdroppedrate	<Double>	Read-only	Rate (/s) counter for qosbytesdropped
qosbytessentnotclassified	<Double>	Read-only	Bytes scheduled by QoS that were not classified.
qosbytessentnotclassifiedrate	<Double>	Read-only	Rate (/s) counter for qosbytessentnotclassified
qosbytesdroppednoconnection	<Double>	Read-only	Bytes dropped by QoS when no connection was found.
qosbytesdroppednoconnectionrate	<Double>	Read-only	Rate (/s) counter for qosbytesdroppednoconnection
qosinputpackets	<Double>	Read-only	Packets sent to QoS for scheduling.
qosinputpacketsrate	<Double>	Read-only	Rate (/s) counter for qosinputpackets
qosoutputpackets	<Double>	Read-only	Packets from QoS to be sent

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
qosoutputpacketsrate	<Double>	Read-only	Rate (/s) counter for qosoutputpackets
qosdroppackets	<Double>	Read-only	Packets Dropped by QoS.
qosdroppacketsrate	<Double>	Read-only	Rate (/s) counter for qosdroppackets
qosrewritemacs	<Double>	Read-only	Number of packets with inband classification in source MAC.
qosrewritemacsrate	<Double>	Read-only	Rate (/s) counter for qosrewritemacs
qospacketsunclassified	<Double>	Read-only	Number of packets without classification.
qospacketsunclassifiedrate	<Double>	Read-only	Rate (/s) counter for qospacketsunclassified
qospacketsclassified	<Double>	Read-only	Number of packets with classification.
qospacketsclassifiedrate	<Double>	Read-only	Rate (/s) counter for qospacketsclassified
qoslearnedmac	<Double>	Read-only	QoS learned true MAC
qoslearnedmacrate	<Double>	Read-only	Rate (/s) counter for qoslearnedmac
qosinputbytes	<Double>	Read-only	Bytes sent to QoS for scheduling
qosinputbytesrate	<Double>	Read-only	Rate (/s) counter for qosinputbytes
qosoutputbytes	<Double>	Read-only	Bytes received from QoS to be sent
qosoutputbytesrate	<Double>	Read-only	Rate (/s) counter for qosoutputbytes
qosfreeheldlist	<Double>	Read-only	No. more packets QoS can hold onto.
qoslink00sent	<Double>	Read-only	QoS bytes sent on Link 00
qoslink00sentrate	<Double>	Read-only	Rate (/s) counter for qoslink00sent
qoslink00drop	<Double>	Read-only	QoS bytes dropped on Link 00
qoslink00droprate	<Double>	Read-only	Rate (/s) counter for qoslink00drop
qoslink01sent	<Double>	Read-only	QoS bytes sent on Link 01
qoslink01sentrate	<Double>	Read-only	Rate (/s) counter for qoslink01sent
qoslink01drop	<Double>	Read-only	QoS bytes dropped on Link 01
qoslink01droprate	<Double>	Read-only	Rate (/s) counter for qoslink01drop
qoslink02sent	<Double>	Read-only	QoS bytes sent on Link 02
qoslink02sentrate	<Double>	Read-only	Rate (/s) counter for qoslink02sent

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
qoslink02droprate	<Double>	Read-only	Rate (/s) counter for qoslink02drop
qoslink03sent	<Double>	Read-only	QoS bytes sent on Link 03
qoslink03sentrate	<Double>	Read-only	Rate (/s) counter for qoslink03sent
qoslink03drop	<Double>	Read-only	QoS bytes dropped on Link 03
qoslink03droprate	<Double>	Read-only	Rate (/s) counter for qoslink03drop
qoslink04sent	<Double>	Read-only	QoS bytes sent on Link 04
qoslink04sentrate	<Double>	Read-only	Rate (/s) counter for qoslink04sent
qoslink04drop	<Double>	Read-only	QoS bytes dropped on Link 04
qoslink04droprate	<Double>	Read-only	Rate (/s) counter for qoslink04drop
qoslink05sent	<Double>	Read-only	QoS bytes sent on Link 05
qoslink05sentrate	<Double>	Read-only	Rate (/s) counter for qoslink05sent
qoslink05drop	<Double>	Read-only	QoS bytes dropped on Link 05
qoslink05droprate	<Double>	Read-only	Rate (/s) counter for qoslink05drop
qoslink06sent	<Double>	Read-only	QoS bytes sent on Link 06
qoslink06sentrate	<Double>	Read-only	Rate (/s) counter for qoslink06sent
qoslink06drop	<Double>	Read-only	QoS bytes dropped on Link 06
qoslink06droprate	<Double>	Read-only	Rate (/s) counter for qoslink06drop
qoslink07sent	<Double>	Read-only	QoS bytes sent on Link 07
qoslink07sentrate	<Double>	Read-only	Rate (/s) counter for qoslink07sent
qoslink07drop	<Double>	Read-only	QoS bytes dropped on Link 07
qoslink07droprate	<Double>	Read-only	Rate (/s) counter for qoslink07drop
qoslink08sent	<Double>	Read-only	QoS bytes sent on Link 08
qoslink08sentrate	<Double>	Read-only	Rate (/s) counter for qoslink08sent
qoslink08drop	<Double>	Read-only	QoS bytes dropped on Link 08
qoslink08droprate	<Double>	Read-only	Rate (/s) counter for qoslink08drop
qoslink09sent	<Double>	Read-only	QoS bytes sent on Link 09

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
qoslink09sentrate	<Double>	Read-only	Rate (/s) counter for qoslink09sent
qoslink09drop	<Double>	Read-only	QoS bytes dropped on Link 09
qoslink09droprate	<Double>	Read-only	Rate (/s) counter for qoslink09drop
qoslink10sent	<Double>	Read-only	QoS bytes sent on Link 10
qoslink10sentrate	<Double>	Read-only	Rate (/s) counter for qoslink10sent
qoslink10drop	<Double>	Read-only	QoS bytes dropped on Link 10
qoslink10droprate	<Double>	Read-only	Rate (/s) counter for qoslink10drop

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/qos

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/qos?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get qos resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:


```
{ "qos": [ {
    "qoslink06droprate":<Double_value>,
    "ipcpe2qosfailed":<Double_value>,
    "qoslink02sent":<Double_value>,
    "qoslink10sent":<Double_value>,
    "ipcmessagessentrate":<Double_value>,
    "qoslink02sentrate":<Double_value>,
    "qosrewritemacsrate":<Double_value>,
    "qoslink08sentrate":<Double_value>,
    "qoslink04drop":<Double_value>,
    "qoslink01droprate":<Double_value>,
    "qosoutputbytesrate":<Double_value>,
    "qoslink08drop":<Double_value>,
    "qoslink05sent":<Double_value>,
    "qoslink10droprate":<Double_value>,
    "qoslink00drop":<Double_value>,
    "ipcmessagesfailedrate":<Double_value>,
    "qoslink05droprate":<Double_value>,
    "qoslink08droprate":<Double_value>,
    "qosbytessentnotclassifiedrate":<Double_value>,
    "qoslink03droprate":<Double_value>,
    "qoslink09sent":<Double_value>,
    "qospacketsunclassifiedrate":<Double_value>,
    "qosfreeheldlist":<Double_value>,
    "qospacketsclassified":<Double_value>,
    "qosbytesdroppednoconnectionrate":<Double_value>,
    "qoslink01sentrate":<Double_value>,
    "qoslink09sentrate":<Double_value>,
    "qoslink03drop":<Double_value>,
    "qoslink03sentrate":<Double_value>,
    "qoslink02drop":<Double_value>,
    "qosinputbytes":<Double_value>,
    "qosinputpacketsrate":<Double_value>,
    "qosoutputbytes":<Double_value>,
    "ipcmessagesfailed":<Double_value>,
    "qoslink07sentrate":<Double_value>,
    "qosinputbytesrate":<Double_value>,
    "qoslink04droprate":<Double_value>,
    "ipcpe2qosfailedrate":<Double_value>,
    "qosoutputpackets":<Double_value>,
    "qosbytesdroppedrate":<Double_value>,
    "qoslink06sent":<Double_value>,
    "qoslink04sent":<Double_value>,
    "qoslink00sent":<Double_value>,
    "qoslink07sent":<Double_value>,
    "qoslink09droprate":<Double_value>,
    "qoslink10drop":<Double_value>,
    "qospacketsunclassified":<Double_value>,
    "qosoutputpacketsrate":<Double_value>,
    "qoslink03sent":<Double_value>,
    "qoslink05sentrate":<Double_value>,
    "qoslink04sentrate":<Double_value>,
    "qosbytesdropped":<Double_value>,
    "qosdroppackets":<Double_value>,
    "qoslink06drop":<Double_value>,
    "qosinputpackets":<Double_value>,
    "qoslink07droprate":<Double_value>,
    "ipcmessagesreceived":<Double_value>,
    "qoslink09drop":<Double_value>,
    "qoslink02droprate":<Double_value>,
    "qoslink07drop":<Double_value>,
    "qoslink05drop":<Double_value>,
    "ipcpe2qosreceivedrate":<Double_value>,
    "qoslink00sentrate":<Double_value>,
    "ipcpe2qossent":<Double_value>,
    "qoslink01sent":<Double_value>,
    "qoslearnedmacrate":<Double_value>,
    "qoslink08sent":<Double_value>,
    "qoslink06sentrate":<Double_value>,
    "qoslink01drop":<Double_value>,
    "ipcmessagesreceivedrate":<Double_value>,
    "ipcpe2qossentrate":<Double_value>,
```


 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "qosbytessentnotclassified":<Double_value>,
    "qoslearnedmac":<Double_value>,
    "ipcmessagessent":<Double_value>,
    "qoslink00droprate":<Double_value>,
    "qoslink10sentrate":<Double_value>,
    "ipcpe2qosreceived":<Double_value>,
    "qosdroppacketsrate":<Double_value>,
    "qosbytesdroppednoconnection":<Double_value>,
    "qospacketsclassifiedrate":<Double_value>,
    "qosrewritemacs":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Rewrite

May 20, 2016

Rewrite statistics.

Name	Description
rewritepolicy	Statistics for rewrite policy resource.
rewritepolicylabel	Statistics for rewrite policy label resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicy

May 20, 2016

Statistics for rewrite policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the rewrite policy. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/rewritepolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/rewritepolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String**

Use this query-parameter to get rewritepolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicy": [ {
    "name":<String_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>,
    "pipolicyundefhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/rewritepolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicy": [ {
    "name":<String_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>,
    "pipolicyundefhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

rewritepolicylabel

Jul 07, 2017
Statistics for rewrite policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the rewrite policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/rewritepolicylabel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/rewritepolicylabel?**args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstat**

Use this query-parameter to get rewritepolicylabel resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicylabel": [ {
      "labelname":<String_value>,
      "pipolicylabelhitsrate":<Double_value>,
      "pipolicylabelhits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/rewritepolicylabel/labelname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "rewritepolicylabel": [ {
      "labelname":<String_value>,
      "pipolicylabelhitsrate":<Double_value>,
      "pipolicylabelhits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Responder

Jul 07, 2017

Responder statistics.

Name	Description
responderpolicy	Statistics for responder policy resource.
responderpolicylabel	Statistics for responder policy label resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicy

May 20, 2016

Statistics for responder policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the responder policy for which to show detailed statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/responderpolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/responderpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<Stri**

Use this query-parameter to get responderpolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/responderpolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

responderpolicylabel

May 20, 2016

Statistics for responder policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	Name of the responder policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/responderpolicylabel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/responderpolicylabel?**args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearst**

Use this query-parameter to get responderpolicylabel resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicylabel": [ {
    "labelname":<String_value>,
    "pipolicylabelhitsrate":<Double_value>,
    "pipolicylabelhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/responderpolicylabel/labelname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "responderpolicylabel": [ {
    "labelname":<String_value>,
    "pipolicylabelhitsrate":<Double_value>,
    "pipolicylabelhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Sure Connect

Jul 07, 2017

SureConnect statistics. SureConnect allows the system to deliver alternate content to clients when the system is under heavy load for the original URL. The original content will be served to the client, automatically, after an estimated delay period.

Name	Description
sc	Statistics for sc.
scpolicy	Statistics for SureConnect policy resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

sc

May 20, 2016
Statistics for sc.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sctotcondtriggered	<Double>	Read-only	Number of times that SureConnect conditions were triggered.
sccondtriggeredrate	<Double>	Read-only	Rate (/s) counter for sctotcondtriggered
scpolicyurlhits	<Double>	Read-only	Total number of incoming requests that matched configured sureconnect policies.
scpolicyurlhitsrate	<Double>	Read-only	Rate (/s) counter for scpolicyurlhits
scpopups	<Double>	Read-only	Total number of in-memory java script served which throws the pop-up window.
scpopupsrate	<Double>	Read-only	Rate (/s) counter for scpopups
sctotreissuedrequests	<Double>	Read-only	Total number of reissued SureConnect requests.
screissuedrequestsrate	<Double>	Read-only	Rate (/s) counter for sctotreissuedrequests
scsessionreqs	<Double>	Read-only	Total number of requests that were handled in a single SureConnect session.
scsessionreqsrate	<Double>	Read-only	Rate (/s) counter for scsessionreqs
scaltconturls	<Double>	Read-only	Total number of alternate content served which throws the pop-up window.
scaltconturlsrate	<Double>	Read-only	Rate (/s) counter for scaltconturls
scpostreqs	<Double>	Read-only	Total number of HTTP POST requests that triggered SureConnect feature.
scpostreqsrate	<Double>	Read-only	Rate (/s) counter for scpostreqs
scresetstats	<Double>	Read-only	Toal number of times that SureConnect statistics were reset.
scresetstatsrate	<Double>	Read-only	Rate (/s) counter for scresetstats
scunsupbrow	<Double>	Read-only	Total number of requests that came from all unsupported browsers.
scunsupbrowrate	<Double>	Read-only	Rate (/s) counter for scunsupbrow
scfaultycookies	<Double>	Read-only	Total number of corrupted SureConnect cookies.
scfaultycookiesrate	<Double>	Read-only	Rate (/s) counter for scfaultycookies

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/sc

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/sc?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get sc resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sc": [ {
    "scpopups":<Double_value>,
    "scfaultycookiesrate":<Double_value>,
    "screissuedrequestsrate":<Double_value>,
    "scpostreqs":<Double_value>,
    "scaltconturls":<Double_value>,
    "scresetstatsrate":<Double_value>,
    "scpopupsrate":<Double_value>,
    "sctotreissuedrequests":<Double_value>,
    "scunsupbrow":<Double_value>,
    "scpolicyurlhitsrate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "sccondtriggeredrate":<Double_value>,
    "scpolicyurlhits":<Double_value>,
    "scsessionreqsrate":<Double_value>,
    "scresetstats":<Double_value>,
    "sctotcondtriggered":<Double_value>,
    "scfaultycookies":<Double_value>,
    "scsessionreqs":<Double_value>,
    "scaltconturlsrate":<Double_value>,
    "scunsupbrowrate":<Double_value>,
    "scpostreqsrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

scpolicy

May 20, 2016

Statistics for SureConnect policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy about which to display statistics. To display statistics about all SureConnect policies, do not set this parameter. Minimum length = 1 Maximum length = 31
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
avgservertransactiontime	<Double>	Read-only	Average server transaction time in seconds for this SureConnect Policy.
avgservertransactiontimerate	<Double>	Read-only	Rate (/s) counter for avgservertransactiontime
scaverageclientttlb	<Double>	Read-only	Average value of the client Time-To-Last-Byte in seconds for this SureConnect policy.
scaverageclientttlbrate	<Double>	Read-only	Rate (/s) counter for scaverageclientttlb
scphysicalserviceip	<String>	Read-only	IP address of the service in dotted notation for which these statistics are maintained.
scphysicalserviceport	<Integer>	Read-only	Port of the service for which these statistics are maintained.
sccurrentclientconnections	<Double>	Read-only	Number of clients currently allowed a server connection by this SureConnect policy.
sccurrentclientconnectionsrate	<Double>	Read-only	Rate (/s) counter for sccurrentclientconnections
sccurrentwaitingclients	<Double>	Read-only	Current number of SureConnect priority clients that are waiting for a server connection.
sccurrentwaitingclientsrate	<Double>	Read-only	Rate (/s) counter for sccurrentwaitingclients
totopenconn	<Double>	Read-only	Current number of open connections to the servers matching this policy.
openconnrate	<Double>	Read-only	Rate (/s) counter for totopenconn
sccurrentwaitingtime	<Double>	Read-only	Value of the currently estimated waiting time in seconds for the configured URL.
sccurrentwaitingtimerate	<Double>	Read-only	Rate (/s) counter for sccurrentwaitingtime
sctotalclientconnections	<Double>	Read-only	Total number of clients that were allowed a server connection by this SureConnect policy.
scclientconnectionsrate	<Double>	Read-only	Rate (/s) counter for sctotalclientconnections
sctotalserverconnections	<Double>	Read-only	Total number of server connections that were established through this SureConnect policy.
scserverconnectionsrate	<Double>	Read-only	Rate (/s) counter for sctotalserverconnections
totclienttransaction	<Double>	Read-only	Total number of client transactions processed by this SureConnect policy.
clienttransactionrate	<Double>	Read-only	Rate (/s) counter for totclienttransaction
sctotalservertransactions	<Double>	Read-only	Number of 200 OK responses received from the web server by this SureConnect policy.
scservertransactionsrate	<Double>	Read-only	Rate (/s) counter for sctotalservertransactions
sctotalrequestsreceived	<Double>	Read-only	Total number of requests received by this SureConnect policy.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

screquestsreceivedrate **Name**	<Double> **Data Type**	Read-only **Permissions**	Rate (/s) counter for sctotalrequestsreceived **Description**
sctotalrequestbytes	<Double>	Read-only	Total number of request bytes received by this SureConnect policy.
screquestbytesrate	<Double>	Read-only	Rate (/s) counter for sctotalrequestbytes
sctotalresponsesreceived	<Double>	Read-only	Total number of server responses received by this SureConnect policy.
screspons esreceivedrate	<Double>	Read-only	Rate (/s) counter for sctotalresponsesreceived
sctotalresponsebytes	<Double>	Read-only	Total number of response bytes received by this SureConnect policy.
screspons ebytesrate	<Double>	Read-only	Rate (/s) counter for sctotalresponsebytes

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/scpolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/scpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_valu**

Use this query-parameter to get scpolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
{ "scpolicy": [ {
    "name":<String_value>,
    "sctotalresponsesreceived":<Double_value>,
    "sccurrentwaitingtime":<Double_value>,
    "clienttransactionrate":<Double_value>,
    "sccurrentwaitingtimerate":<Double_value>,
    "sccurrentwaitingclients":<Double_value>,
    "sctotalrequestsreceived":<Double_value>,
    "screquestsreceivedrate":<Double_value>,
    "scresponsebytesrate":<Double_value>,
    "screquestbytesrate":<Double_value>,
    "sctotalresponsebytes":<Double_value>,
    "scserverconnectionsrate":<Double_value>,
    "sctotalservertransactions":<Double_value>,
    "sccurrentwaitingclientsrate":<Double_value>,
    "scaverageclientttlb":<Double_value>,
    "avgservertransactiontime":<Double_value>,
    "scphysicalserviceip":<String_value>,
    "sccurrentclientconnectionsrate":<Double_value>,
    "sccurrentclientconnections":<Double_value>,
    "openconnrate":<Double_value>,
    "scclientconnectionsrate":<Double_value>,
    "scaverageclientttlbrate":<Double_value>,
    "totclienttransaction":<Double_value>,
    "scservertransactionsrate":<Double_value>,
    "sctotalclientconnections":<Double_value>,
    "scphysicalserviceport":<Integer_value>,
    "totopenconn":<Double_value>,
    "avgservertransactiontimerate":<Double_value>,
    "scresponsesreceivedrate":<Double_value>,
    "sctotalrequestbytes":<Double_value>,
    "sctotalserverconnections":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/scpolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "scpolicy": [ {
    "name":<String_value>,
    "sctotalresponsesreceived":<Double_value>,
    "sccurrentwaitingtime":<Double_value>,
    "clienttransactionrate":<Double_value>,
    "sccurrentwaitingtimerate":<Double_value>,
    "sccurrentwaitingclients":<Double_value>,
    "sctotalrequestsreceived":<Double_value>,
    "screquestsreceivedrate":<Double_value>,
    "scresponsebytesrate":<Double_value>,
    "screquestbytesrate":<Double_value>,
    "sctotalresponsebytes":<Double_value>,
    "scserverconnectionsrate":<Double_value>,
    "sctotalservertransactions":<Double_value>,
    "sccurrentwaitingclientsrate":<Double_value>,
    "scaverageclientttlb":<Double_value>,
    "avgservertransactiontime":<Double_value>,
    "scphysicalserviceip":<String_value>,
    "sccurrentclientconnectionsrate":<Double_value>,
    "sccurrentclientconnections":<Double_value>,
    "openconnrate":<Double_value>,
    "scclientconnectionsrate":<Double_value>,
    "scaverageclientttlbrate":<Double_value>,
    "totclienttransaction":<Double_value>,
    "scservertransactionsrate":<Double_value>,
    "sctotalclientconnections":<Double_value>,
    "scphysicalserviceport":<Integer_value>,
    "totopenconn":<Double_value>,
    "avgservertransactiontimerate":<Double_value>,
    "scresponsesreceivedrate":<Double_value>,
    "sctotalrequestbytes":<Double_value>,
    "sctotalserverconnections":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SNMP

May 20, 2016

SNMP(Simple Network Management Protocol) statistics.

Name	Description
snmp	Statistics for snmp.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

snmp

May 20, 2016

Statistics for snmp.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
snmptotrxpkts	<Double>	Read-only	SNMP packets received.
snmprxpktsrate	<Double>	Read-only	Rate (/s) counter for snmptotrxpkts
snmptottxpkts	<Double>	Read-only	SNMP packets transmitted.
snmptxpktsrate	<Double>	Read-only	Rate (/s) counter for snmptottxpkts
snmptotgetreqs	<Double>	Read-only	SNMP Get-Request PDUs that have been accepted and processed.
snmpgetreqsrate	<Double>	Read-only	Rate (/s) counter for snmptotgetreqs
snmptotgetnextreqs	<Double>	Read-only	SNMP Get-Next PDUs that have been accepted and processed.
snmpgetnextreqsrate	<Double>	Read-only	Rate (/s) counter for snmptotgetnextreqs
snmptotgetbulkreqs	<Double>	Read-only	SNMP Get-Bulk PDUs that have been accepted and pro?cessed.
snmpgetbulkreqsrate	<Double>	Read-only	Rate (/s) counter for snmptotgetbulkreqs
snmptotresponses	<Double>	Read-only	SNMP Get-Response PDUs that have been generated by the NetScaler.
snmpresponsesrate	<Double>	Read-only	Rate (/s) counter for snmptotresponses
snmptottraps	<Double>	Read-only	SNMP Trap PDUs that have been generated by the NetScaler.
snmptoterrreqdropped	<Double>	Read-only	SNMP requests dropped.
snmptotparseerrs	<Double>	Read-only	Number of ASN.1 or BER errors encountered when decoding received SNMP Messages.
snmptotbadversions	<Double>	Read-only	Number of SNMP messages received, which were for an unsupported SNMP version.
snmptotbadcommname	<Double>	Read-only	SNMP messages received, which used an SNMP community name not known to the NetScaler.
snmptotbadcommuse	<Double>	Read-only	The total number of SNMP Messages received that represented an SNMP operation which was not allowed by the SNMP community named in the Message.
snmpunsupportedsecuritylevel	<Double>	Read-only	SNMP packets that were dropped because they requested a security level that was unknown to the NetScaler or otherwise unavailable.
snmpnotintimewindow	<Double>	Read-only	SNMP packets that were dropped because they appeared outside of the authoritative SNMP engine's window.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
snmpunknownengineids	<Double>	Read-only	SNMP packets that were dropped because they referenced an SNMP engine ID that was not known to the NetScaler.
snmpwrongdigests	<Double>	Read-only	SNMP packets that were dropped because they did not contain the expected digest value.
snmpdecryptionerrors	<Double>	Read-only	SNMP packets that were dropped because they could not be decrypted.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/snmp

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/snmp?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get snmp resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "snmp": [ {
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "snmpdecryptionerrors":<Double_value>,
    "snmptotresponses":<Double_value>,
    "snmptotbadcommuse":<Double_value>,
    "snmptoterrreqdropped":<Double_value>,
    "snmpgetnextreqsrate":<Double_value>,
    "snmptotrxpkts":<Double_value>,
    "snmptottxpkts":<Double_value>,
    "snmptotparseerrs":<Double_value>,
    "snmptottraps":<Double_value>,
    "snmptotbadversions":<Double_value>,
    "snmptxpktsrate":<Double_value>,
    "snmpresponsesrate":<Double_value>,
    "snmpgetreqsrate":<Double_value>,
    "snmptotbadcommname":<Double_value>,
    "snmptotgetnextreqs":<Double_value>,
    "snmpunknownengineids":<Double_value>,
    "snmpwrongdigests":<Double_value>,
    "snmpgetbulkreqsrate":<Double_value>,
    "snmpnotintimewindow":<Double_value>,
    "snmptotgetbulkreqs":<Double_value>,
    "snmpunknownusername":<Double_value>,
    "snmpunsupportedsecuritylevel":<Double_value>,
    "snmptotgetreqs":<Double_value>,
    "snmprxpktsrate":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Spillover

May 20, 2016

Spillover policies and actions.

Name	Description
spilloverpolicy	Statistics for Spillover policy resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

spilloverpolicy

May 20, 2016

Statistics for Spillover policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the spillover policy for which to show detailed statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/spilloverpolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/spilloverpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String**

Use this query-parameter to get spilloverpolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "spilloverpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/spilloverpolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "spilloverpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL

May 20, 2016

SSL statistics

Name	Description
ssl	Statistics for ssl.
sslvserver	Statistics for SSL virtual server resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ssl

May 20, 2016

Statistics for ssl.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sslcryptoutilizationstat	<Double>	Read-only	Utilization of the hardware crypto resource. Only valid values are 0-100%. Only works for platform with N3 chips
sslnumcardsup	<Double>	Read-only	Number of SSL cards that are UP. If the number of cards UP is lower than a threshold, a failover is initiated.
sslcardstatus	<Double>	Read-only	Status of the SSL card(s). The value should be interpreted in binary form, with each set bit indicates a card as UP.
sslcards	<Double>	Read-only	Number of SSL crypto cards present on the NetScaler appliance.
sslenginestatus	<Double>	Read-only	State of the SSL Engine (1=UP/0=DOWN). This state is decided based on SSL Feature/License status and minimum number of cards UP.
ssltotsessions	<Double>	Read-only	Number of SSL sessions on the NetScaler appliance.
sslsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsessions
ssltottransactions	<Double>	Read-only	Number of SSL transactions on the NetScaler appliance.
ssltransactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottransactions
ssltotsslv2transactions	<Double>	Read-only	Number of SSLv2 transactions on the NetScaler appliance.
sslsslv2transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv2transactions
ssltotsslv3transactions	<Double>	Read-only	Total number of SSLv3 transactions on the NetScaler appliance.
sslsslv3transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv3transactions
ssltottlsv1transactions	<Double>	Read-only	Number of TLSv1 transactions on the NetScaler appliance.
ssltlsv1transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv1transactions
ssltottlsv11transactions	<Double>	Read-only	Number of TLSv1.1 transactions on the NetScaler appliance.
ssltlsv11transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv11transactions
ssltottlsv12transactions	<Double>	Read-only	Number of TLSv1.2 transactions on the NetScaler appliance.
ssltlsv12transactionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv12transactions
ssltotsslv2sessions	<Double>	Read-only	Number of SSLv2 sessions on the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ssltotsslv3sessions	<Double>	Read-only	Number of SSLv3 sessions on the NetScaler appliance.
sslsslv3sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv3sessions
ssltottlsv1sessions	<Double>	Read-only	Number of TLSv1 sessions on the NetScaler appliance.
ssltlsv1sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv1sessions
ssltottlsv11sessions	<Double>	Read-only	Number of TLSv1.1 sessions on the NetScaler appliance.
ssltlsv11sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv11sessions
ssltottlsv12sessions	<Double>	Read-only	Number of TLSv1.2 sessions on the NetScaler appliance.
ssltlsv12sessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv12sessions
ssltotnewsessions	<Double>	Read-only	Number of new SSL sessions created on the NetScaler appliance.
sslnewsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotnewsessions
ssltotsessionmiss	<Double>	Read-only	Number of SSL session reuse misses on the NetScaler appliance.
sslsessionmissrate	<Double>	Read-only	Rate (/s) counter for ssltotsessionmiss
ssltotsessionhits	<Double>	Read-only	Number of SSL session reuse hits on the NetScaler appliance.
sslsessionhitsrate	<Double>	Read-only	Rate (/s) counter for ssltotsessionhits
sslbetotsessions	<Double>	Read-only	Number of back-end SSL sessions on the NetScaler appliance.
sslbesessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetotsessions
sslbetotsslv3sessions	<Double>	Read-only	Number of back-end SSLv3 sessions on the NetScaler appliance.
sslbesslv3sessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetotsslv3sessions
sslbetottlsv1sessions	<Double>	Read-only	Number of back-end TLSv1 sessions on the NetScaler appliance.
sslbetlsv1sessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv1sessions
sslbetottlsv11sessions	<Double>	Read-only	Number of back-end TLSv1.1 sessions on the NetScaler appliance.
sslbetlsv11sessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv11sessions
sslbetottlsv12sessions	<Double>	Read-only	Number of back-end TLSv1.2 sessions on the NetScaler appliance.
sslbetlsv12sessionsrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv12sessions
sslbetotsessionmultiplexattempts	<Double>	Read-only	Number of back-end SSL session multiplex attempts on the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sslbetotsessionmultiplexattemptsuccess	<Double>	Read-only	Number of back-end SSL session multiplex successes on the NetScaler appliance.
sslbesessionmultiplexattemptsuccessrate	<Double>	Read-only	Rate (/s) counter for sslbetotsessionmultiplexattemptsuccess
sslbetotsessionmultiplexattemptfails	<Double>	Read-only	Number of back-end SSL session multiplex failures on the NetScaler appliance.
sslbesessionmultiplexattemptfailsrate	<Double>	Read-only	Rate (/s) counter for sslbetotsessionmultiplexattemptfails
ssltotenc	<Double>	Read-only	Number of bytes encrypted on the NetScaler appliance.
sslencrate	<Double>	Read-only	Rate (/s) counter for ssltotenc
ssltotdec	<Double>	Read-only	Number of bytes decrypted on the NetScaler appliance.
ssldecrate	<Double>	Read-only	Rate (/s) counter for ssltotdec
ssltotrenegsessions	<Double>	Read-only	Number of SSL session renegotiations on the NetScaler appliance.
sslrenegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotrenegsessions
ssltotsslv3renegsessions	<Double>	Read-only	Number of session renegotiations done on SSLv3.
sslsslv3renegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv3renegsessions
ssltottlsv1renegsessions	<Double>	Read-only	Number of SSL session renegotiations done on TLSv1.
ssltlsv1renegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv1renegsessions
ssltottlsv11renegsessions	<Double>	Read-only	Number of SSL session renegotiations done on TLSv1.1.
ssltlsv11renegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv11renegsessions
ssltottlsv12renegsessions	<Double>	Read-only	Number of SSL session renegotiations done on TLSv1.2.
ssltlsv12renegsessionsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv12renegsessions
ssltotrsa512keyexchanges	<Double>	Read-only	Number of RSA 512-bit key exchanges on the NetScaler appliance.
sslrsa512keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotrsa512keyexchanges
ssltotrsa1024keyexchanges	<Double>	Read-only	Number of RSA 1024-bit key exchanges on the NetScaler appliance.
sslrsa1024keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotrsa1024keyexchanges
ssltotrsa2048keyexchanges	<Double>	Read-only	Number of RSA 2048-bit key exchanges on the NetScaler appliance.
sslrsa2048keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotrsa2048keyexchanges
ssltotrsa4096keyexchanges	<Double>	Read-only	Number of RSA 4096-bit key exchanges on the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ssltotdh512keyexchanges	<Double>	Read-only	Number of Diffie-Helman 512-bit key exchanges on the NetScaler appliance.
ssldh512keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotdh512keyexchanges
ssltotdh1024keyexchanges	<Double>	Read-only	Number of Diffie-Helman 1024-bit key exchanges on the NetScaler appliance.
ssldh1024keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotdh1024keyexchanges
ssltotdh2048keyexchanges	<Double>	Read-only	Number of Diffie-Helman 2048-bit key exchanges on the NetScaler appliance.
ssldh2048keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotdh2048keyexchanges
ssltotecdhe521keyexchanges	<Double>	Read-only	Number of 521 Elliptical Curve Diffie-Helman on the NetScaler appliance.
sslecdhe521keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotecdhe521keyexchanges
ssltotecdhe384keyexchanges	<Double>	Read-only	Number of 384 Elliptical Curve Diffie-Helman on the NetScaler appliance.
sslecdhe384keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotecdhe384keyexchanges
ssltotecdhe256keyexchanges	<Double>	Read-only	Number of 256 Elliptical Curve Diffie-Helman on the NetScaler appliance.
sslecdhe256keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotecdhe256keyexchanges
ssltotecdhe224keyexchanges	<Double>	Read-only	Number of 224 Elliptical Curve Diffie-Helman on the NetScaler appliance.
sslecdhe224keyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotecdhe224keyexchanges
ssltot40bitrc4ciphers	<Double>	Read-only	Number of RC4 40-bit cipher encryptions on the NetScaler appliance.
ssl40bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot40bitrc4ciphers
ssltot56bitrc4ciphers	<Double>	Read-only	Number of RC4 56-bit cipher encryptions on the NetScaler appliance.
ssl56bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot56bitrc4ciphers
ssltot64bitrc4ciphers	<Double>	Read-only	Number of RC4 64-bit cipher encryptions on the NetScaler appliance.
ssl64bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot64bitrc4ciphers
ssltot128bitrc4ciphers	<Double>	Read-only	Number of RC4 128-bit cipher encryptions on the NetScaler appliance.
ssl128bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot128bitrc4ciphers
ssltot40bitdesciphers	<Double>	Read-only	Number of DES 40-bit cipher encryptions on the NetScaler appliance.
ssl40bitdesciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot40bitdesciphers
ssltot56bitdesciphers	<Double>	Read-only	Number of DES 56-bit cipher encryptions on the NetScaler appliance.
ssl56bitdesciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot56bitdesciphers

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ssltot168bit3desciphers	<Double>	Read-only	Number of DES 168-bit cipher encryptions on the NetScaler appliance.
ssl168bit3desciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot168bit3desciphers
ssltotcipheraes128	<Double>	Read-only	Number of AES 128-bit cipher encryptions on the NetScaler appliance.
sslcipheraes128rate	<Double>	Read-only	Rate (/s) counter for ssltotcipheraes128
ssltotcipheraes256	<Double>	Read-only	Number of AES 256-bit cipher encryptions on the NetScaler appliance.
sslcipheraes256rate	<Double>	Read-only	Rate (/s) counter for ssltotcipheraes256
ssltot40bitrc2ciphers	<Double>	Read-only	Number of RC2 40-bit cipher encryptions on the NetScaler appliance.
ssl40bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot40bitrc2ciphers
ssltot56bitrc2ciphers	<Double>	Read-only	Number of RC2 56-bit cipher encryptions on the NetScaler appliance.
ssl56bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot56bitrc2ciphers
ssltot128bitrc2ciphers	<Double>	Read-only	Number of RC2 128-bit cipher encryptions on the NetScaler appliance.
ssl128bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot128bitrc2ciphers
ssltot128bitaesgcmciphers	<Double>	Read-only	Number of AEC-GCM 128-bit cipher encryptions on the NetScaler appliance.
ssl128bitaesgcmciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot128bitaesgcmciphers
ssltot256bitaesgcmciphers	<Double>	Read-only	Number of AEC-GCM 256-bit cipher encryptions on the NetScaler appliance.
ssl256bitaesgcmciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot256bitaesgcmciphers
ssltotnullciphers	<Double>	Read-only	Number of Null cipher encryptions on the NetScaler appliance.
sslnullciphersrate	<Double>	Read-only	Rate (/s) counter for ssltotnullciphers
ssltotmd5mac	<Double>	Read-only	Number of MD5 hashes on the NetScaler appliance.
sslmd5macrate	<Double>	Read-only	Rate (/s) counter for ssltotmd5mac
ssltotshamac	<Double>	Read-only	Number of SHA hashes on the NetScaler appliance.
sslshamacrate	<Double>	Read-only	Rate (/s) counter for ssltotshamac
ssltotsslv2handshakes	<Double>	Read-only	Number of handshakes on SSLv2 on the NetScaler appliance.
sslsslv2handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv2handshakes
ssltotsslv3handshakes	<Double>	Read-only	Number of handshakes on SSLv3 on the NetScaler appliance.
sslsslv3handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv3handshakes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ssltlsv1handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv1handshakes
ssltottlsv11handshakes	<Double>	Read-only	Number of SSL handshakes on TLSv1.1 on the NetScaler appliance.
ssltlsv11handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv11handshakes
ssltottlsv12handshakes	<Double>	Read-only	Number of SSL handshakes on TLSv1.2 on the NetScaler appliance.
ssltlsv12handshakesrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv12handshakes
ssltotsslv2clientauthentications	<Double>	Read-only	Number of client authentications done on SSLv2.
sslsslv2clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv2clientauthentications
ssltotsslv3clientauthentications	<Double>	Read-only	Number of client authentications done on SSLv3.
sslsslv3clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltotsslv3clientauthentications
ssltottlsv1clientauthentications	<Double>	Read-only	Number of client authentications done on TLSv1.
ssltlsv1clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv1clientauthentications
ssltottlsv11clientauthentications	<Double>	Read-only	Number of client authentications done on TLSv1.1.
ssltlsv11clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv11clientauthentications
ssltottlsv12clientauthentications	<Double>	Read-only	Number of client authentications done on TLSv1.2.
ssltlsv12clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for ssltottlsv12clientauthentications
ssltotrsaauthorizations	<Double>	Read-only	Number of RSA authentications on the NetScaler appliance.
sslrsaauthorizationsrate	<Double>	Read-only	Rate (/s) counter for ssltotrsaauthorizations
ssltotdhauthorizations	<Double>	Read-only	Number of Diffie-Helman authentications on the NetScaler appliance.
ssldhauthorizationsrate	<Double>	Read-only	Rate (/s) counter for ssltotdhauthorizations
ssltotdssauthorizations	<Double>	Read-only	Total number of times DSS authorization is used on the NetScaler appliance.
ssldssauthorizationsrate	<Double>	Read-only	Rate (/s) counter for ssltotdssauthorizations
ssltotnullauthorizations	<Double>	Read-only	Number of Null authentications on the NetScaler appliance.
sslnullauthorizationsrate	<Double>	Read-only	Rate (/s) counter for ssltotnullauthorizations
ssltotbkendsessionrenegotiate	<Double>	Read-only	Number of back-end SSL session renegotiations on the NetScaler appliance.
sslbkendsessionrenegotiaterate	<Double>	Read-only	Rate (/s) counter for ssltotbkendsessionrenegotiate

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sslbkendsslv3renegorate	<Double>	Read-only	Rate (/s) counter for ssltotbkendsslv3renego
ssltotbkendtlsvlrenego	<Double>	Read-only	Number of back-end TLSv1 session renegotiations on the NetScaler appliance.
sslbkendtlsvlrenegorate	<Double>	Read-only	Rate (/s) counter for ssltotbkendtlsvlrenego
ssltotbkendtlsv11renego	<Double>	Read-only	Number of back-end TLSv1.1 session renegotiations on the NetScaler appliance.
sslbkendtlsv11renegorate	<Double>	Read-only	Rate (/s) counter for ssltotbkendtlsv11renego
ssltotbkendtlsv12renego	<Double>	Read-only	Number of back-end TLSv1.2 session renegotiations on the NetScaler appliance.
sslbkendtlsv12renegorate	<Double>	Read-only	Rate (/s) counter for ssltotbkendtlsv12renego
sslbetotrsa512keyexchanges	<Double>	Read-only	Number of back-end RSA 512-bit key exchanges on the NetScaler appliance.
sslbersa512keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotrsa512keyexchanges
sslbetotrsa1024keyexchanges	<Double>	Read-only	Number of back-end RSA 1024-bit key exchanges on the NetScaler appliance.
sslbersa1024keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotrsa1024keyexchanges
sslbetotrsa2048keyexchanges	<Double>	Read-only	Number of back-end RSA 2048-bit key exchanges on the NetScaler appliance.
sslbersa2048keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotrsa2048keyexchanges
sslbetotdh512keyexchanges	<Double>	Read-only	Number of back-end DH 512-bit key exchanges on the NetScaler appliance.
sslbedh512keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotdh512keyexchanges
sslbetotdh1024keyexchanges	<Double>	Read-only	Number of back-end DH 1024-bit key exchanges on the NetScaler appliance.
sslbedh1024keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotdh1024keyexchanges
sslbetotdh2048keyexchanges	<Double>	Read-only	Number of back-end DH 2048-bit key exchanges on the NetScaler appliance.
sslbedh2048keyexchangesrate	<Double>	Read-only	Rate (/s) counter for sslbetotdh2048keyexchanges
sslbetotecdhecurve521	<Double>	Read-only	Number of back-end ECDHE 521 curve Key exchanges on the NetScaler appliance.
sslbeecdhecurve521rate	<Double>	Read-only	Rate (/s) counter for sslbetotecdhecurve521
sslbetotecdhecurve384	<Double>	Read-only	Number of back-end ECDHE 384 curve Key exchanges on the NetScaler appliance.
sslbeecdhecurve384rate	<Double>	Read-only	Rate (/s) counter for sslbetotecdhecurve384
sslbetotecdhecurve256	<Double>	Read-only	Number of back-end ECDHE 256 curve Key exchanges on the NetScaler appliance.
sslbeecdhecurve256rate	<Double>	Read-only	Rate (/s) counter for sslbetotecdhecurve256
sslbetotecdhecurve224	<Double>	Read-only	Number of back-end ECDHE 224 curve Key exchanges on the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sslbeecdhecurve224rate	<Double>	Read-only	Rate (/s) counter for sslbetotecdhecurve224
sslbetot40bitrc4ciphers	<Double>	Read-only	Number of back-end RC4 40-bit cipher encryptions on the NetScaler appliance.
sslbe40bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot40bitrc4ciphers
sslbetot56bitrc4ciphers	<Double>	Read-only	Number of back-end RC4 56-bit cipher encryptions on the NetScaler appliance.
sslbe56bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot56bitrc4ciphers
sslbetot64bitrc4ciphers	<Double>	Read-only	Number of back-end RC4 64-bit cipher encryptions on the NetScaler appliance.
sslbe64bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot64bitrc4ciphers
sslbetot128bitrc4ciphers	<Double>	Read-only	Number of back-end RC4 128-bit cipher encryptions on the NetScaler appliance.
sslbe128bitrc4ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot128bitrc4ciphers
sslbetot40bitdesciphers	<Double>	Read-only	Number of back-end DES 40-bit cipher encryptions on the NetScaler appliance.
sslbe40bitdesciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot40bitdesciphers
sslbetot56bitdesciphers	<Double>	Read-only	Number of back-end DES 56-bit cipher encryptions on the NetScaler appliance.
sslbe56bitdesciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot56bitdesciphers
sslbetot168bit3desciphers	<Double>	Read-only	Number of back-end 3DES 168-bit cipher encryptions on the NetScaler appliance.
sslbe168bit3desciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot168bit3desciphers
ssltotbkendcipheraes128	<Double>	Read-only	Back-end AES 128-bit cipher encryptions on the NetScaler appliance.
sslbkendcipheraes128rate	<Double>	Read-only	Rate (/s) counter for ssltotbkendcipheraes128
ssltotbkendcipheraes256	<Double>	Read-only	Back-end AES 256-bit cipher encryptions on the NetScaler appliance.
sslbkendcipheraes256rate	<Double>	Read-only	Rate (/s) counter for ssltotbkendcipheraes256
sslbetot40bitrc2ciphers	<Double>	Read-only	Number of back-end RC2 40-bit cipher encryptions on the NetScaler appliance.
sslbe40bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot40bitrc2ciphers
sslbetot56bitrc2ciphers	<Double>	Read-only	Number of back-end RC2 56-bit cipher encryptions on the NetScaler appliance.
sslbe56bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot56bitrc2ciphers
sslbetot128bitrc2ciphers	<Double>	Read-only	Number of back-end RC2 128-bit cipher encryptions on the NetScaler appliance.
sslbe128bitrc2ciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot128bitrc2ciphers
sslbetotnullciphers	<Double>	Read-only	Number of back-end null cipher encryptions on the NetScaler appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
sslbenullciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetotnullciphers
sslbetotmd5mac	<Double>	Read-only	Number of back-end MD5 hashes on the NetScaler appliance.
sslbemd5macrate	<Double>	Read-only	Rate (/s) counter for sslbetotmd5mac
sslbetotshamac	<Double>	Read-only	Number of back-end SHA hashes on the NetScaler appliance.
sslbeshamacrate	<Double>	Read-only	Rate (/s) counter for sslbetotshamac
sslbetotsslv3handshakes	<Double>	Read-only	Number of back-end SSLv3 handshakes on the NetScaler appliance.
sslbesslv3handshakesrate	<Double>	Read-only	Rate (/s) counter for sslbetotsslv3handshakes
sslbetottlsv1handshakes	<Double>	Read-only	Number of back-end TLSv1 handshakes on the NetScaler appliance.
sslbetlsv1handshakesrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv1handshakes
sslbetotsslv3clientauthentications	<Double>	Read-only	Number of back-end SSLv3 client authentications on the NetScaler appliance.
sslbesslv3clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotsslv3clientauthentications
sslbetottlsv1clientauthentications	<Double>	Read-only	Number of back-end TLSv1 client authentications on the NetScaler appliance.
sslbetlsv1clientauthenticationsrate	<Double>	Read-only	Rate (/s) counter for sslbetottlsv1clientauthentications
sslbetotrsaauthorizations	<Double>	Read-only	Number of back-end RSA authentications on the NetScaler appliance.
sslbersaauthorizationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotrsaauthorizations
sslbetotdhauthorizations	<Double>	Read-only	Number of back-end DH authentications on the NetScaler appliance.
sslbedhauthorizationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotdhauthorizations
sslbetotdssauthorizations	<Double>	Read-only	Number of back-end DSS authentications on the NetScaler appliance.
sslbedssauthorizationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotdssauthorizations
sslbetotnullauthorizations	<Double>	Read-only	Number of back-end null authentications on the NetScaler appliance.
sslbenullauthorizationsrate	<Double>	Read-only	Rate (/s) counter for sslbetotnullauthorizations
ssltotoffloadrsakeyexchanges	<Double>	Read-only	Number of RSA key exchanges offloaded to the cryptography card.
ssloffloadrsakeyexchangesrate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadrsakeyexchanges
ssltotoffloadsignrsa	<Double>	Read-only	Number of RSA sign operations offloaded to the cryptography card.
ssloffloadsignrsarate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadsignrsa
ssltotoffloaddhkeyexchanges	<Double>	Read-only	Number of DH key exchanges offloaded to the cryptography card.d

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ssltotoffloadbulkrc4	<Double>	Read-only	Number of RC4 encryptions offloaded to the cryptography card.
sslloffloadbulkrc4rate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadbulkrc4
ssltotoffloadbulkdes	<Double>	Read-only	Number of DES encryptions offloaded to the cryptography card.
sslloffloadbulkdesrate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadbulkdes
ssltotoffloadbulkaes	<Double>	Read-only	Number of AES encryptions offloaded to the cryptography card.
sslloffloadbulkaesrate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadbulkaes
ssltotoffloadbulkaesgcm128	<Double>	Read-only	Number of AES-GCM 128-bit encryptions offloaded to the cryptography card.
sslloffloadbulkaesgcm128rate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadbulkaesgcm128
ssltotoffloadbulkaesgcm256	<Double>	Read-only	Number of AES-GCM 256-bit encryptions offloaded to the cryptography card.
sslloffloadbulkaesgcm256rate	<Double>	Read-only	Rate (/s) counter for ssltotoffloadbulkaesgcm256
ssltotenchw	<Double>	Read-only	Number of bytes encrypted in hardware.
sslenchwrate	<Double>	Read-only	Rate (/s) counter for ssltotenchw
ssltotencsw	<Double>	Read-only	Number of bytes encrypted in software.
sslencswrate	<Double>	Read-only	Rate (/s) counter for ssltotencsw
ssltotencfe	<Double>	Read-only	Number of bytes encrypted on the front end.
sslencferate	<Double>	Read-only	Rate (/s) counter for ssltotencfe
ssltothwencfe	<Double>	Read-only	Number of bytes encrypted in hardware on the front end.
sslhwencferate	<Double>	Read-only	Rate (/s) counter for ssltothwencfe
ssltotswencfe	<Double>	Read-only	Number of bytes encrypted in software on the front end.
sslswencferate	<Double>	Read-only	Rate (/s) counter for ssltotswencfe
ssltotencbe	<Double>	Read-only	Number of bytes encrypted on the back end.
sslencberate	<Double>	Read-only	Rate (/s) counter for ssltotencbe
ssltothwencbe	<Double>	Read-only	Number of bytes encrypted in hardware on the back end.
sslhwencberate	<Double>	Read-only	Rate (/s) counter for ssltothwencbe
ssltotswencbe	<Double>	Read-only	Number of bytes encrypted in software on the back end.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
ssltotdechw	<Double>	Read-only	Number of bytes decrypted in hardware.
ssldechwrate	<Double>	Read-only	Rate (/s) counter for ssltotdechw
ssltotdecsw	<Double>	Read-only	Number of bytes decrypted in software.
ssldecswrate	<Double>	Read-only	Rate (/s) counter for ssltotdecsw
ssltotdecfe	<Double>	Read-only	Number of bytes decrypted on the front end.
ssldecferate	<Double>	Read-only	Rate (/s) counter for ssltotdecfe
ssltothwdecfe	<Double>	Read-only	Number of bytes decrypted in hardware on the front end.
sslhwdecferate	<Double>	Read-only	Rate (/s) counter for ssltothwdecfe
ssltotswdecfe	<Double>	Read-only	Number of bytes decrypted in software on the front end.
sslswdecferate	<Double>	Read-only	Rate (/s) counter for ssltotswdecfe
ssltotdecbe	<Double>	Read-only	Number of bytes decrypted on the back end.
ssldecberate	<Double>	Read-only	Rate (/s) counter for ssltotdecbe
ssltothwdecbe	<Double>	Read-only	Number of bytes decrypted in hardware on the back end.
sslhwdecberate	<Double>	Read-only	Rate (/s) counter for ssltothwdecbe
ssltotswdecbe	<Double>	Read-only	Number of bytes decrypted in software on back-end
sslswdecberate	<Double>	Read-only	Rate (/s) counter for ssltotswdecbe
sslbemaxmultiplexedsessions	<Double>	Read-only	Number of back-end SSL sessions reused on the NetScaler appliance.
sslbemaxmultiplexedsessionsrate	<Double>	Read-only	Rate (/s) counter for sslbemaxmultiplexedsessions
ssltot128bitideaciphers	<Double>	Read-only	Number of IDEA 128-bit cipher encryptions on the NetScaler appliance.
ssl128bitideaciphersrate	<Double>	Read-only	Rate (/s) counter for ssltot128bitideaciphers
sslbetot128bitideaciphers	<Double>	Read-only	Number of back-end IDEA 128-bit cipher encryptions on the NetScaler appliance.
sslbe128bitideaciphersrate	<Double>	Read-only	Rate (/s) counter for sslbetot128bitideaciphers

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/ssl

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/ssl?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get ssl resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "ssl": [ {
    "ssltlsv11handshakesrate":<Double_value>,
    "ssldh512keyexchangesrate":<Double_value>,
    "sslbetottlsv1handshakes":<Double_value>,
    "sslswdecberate":<Double_value>,
    "ssltottransactions":<Double_value>,
    "sslbetot40bitrc4ciphers":<Double_value>,
    "sslnumcardsup":<Double_value>,
    "sslbetlsv1clientauthenticationsrate":<Double_value>,
    "sslrenegsessionsrate":<Double_value>,
    "ssltottlsv11handshakes":<Double_value>,
    "sslbetotsessions":<Double_value>,
    "sslbetot64bitrc4ciphers":<Double_value>,
    "sslbersa2048keyexchangesrate":<Double_value>,
    "sslbetottlsv1clientauthentications":<Double_value>,
    "ssltothwencbe":<Double_value>,
    "sslbetotdh512keyexchanges":<Double_value>,
    "ssltotrsaauthorizations":<Double_value>,
    "sslecdhe384keyexchangesrate":<Double_value>,
    "sslcipheraes128rate":<Double_value>,
    "ssltotsslv2sessions":<Double_value>,
    "ssltothwdecfe":<Double_value>,
    "sslloffloadbulkrc4rate":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

"ssltotbkendtlsv11renego":<Double_value>,
"sslbe128bitrc4ciphersrate":<Double_value>,
"ssltotdecsw":<Double_value>,
"ssl128bitrc2ciphersrate":<Double_value>,
"ssloffloadbulkaesrate":<Double_value>,
"sslrsa512keyexchangesrate":<Double_value>,
"sslbe56bitrc2ciphersrate":<Double_value>,
"ssltotrsa4096keyexchanges":<Double_value>,
"sslbetotsslv3clientauthentications":<Double_value>,
"sslbeecdhecurve256rate":<Double_value>,
"ssltotsslv2clientauthentications":<Double_value>,
"sslbetlsv11sessionsrate":<Double_value>,
"sslbkendtlsv11renegorate":<Double_value>,
"ssldh1024keyexchangesrate":<Double_value>,
"ssltotmd5mac":<Double_value>,
"sslbetotnullauthorizations":<Double_value>,
"sslbesessionmultiplexattemptsrate":<Double_value>,
"sslsslv2handshakesrate":<Double_value>,
"ssldssauthorizationsrate":<Double_value>,
"sslbemaxmultiplexedsessionsrate":<Double_value>,
"sslbkendsessionrenegotiaterate":<Double_value>,
"sslbetotrsa512keyexchanges":<Double_value>,
"ssltlsv1renegsessionsrate":<Double_value>,
"ssltot168bit3desciphers":<Double_value>,
"ssltottlsv1handshakes":<Double_value>,
"sslbetot56bitdesciphers":<Double_value>,
"sslbe128bitrc2ciphersrate":<Double_value>,
"ssltottlsv12handshakes":<Double_value>,
"sslrsa2048keyexchangesrate":<Double_value>,
"ssltotsessionhits":<Double_value>,
"ssl40bitdesciphersrate":<Double_value>,
"ssltotencfe":<Double_value>,
"ssl40bitrc4ciphersrate":<Double_value>,
"sslbe56bitrc4ciphersrate":<Double_value>,
"ssltottlsv11renegsessions":<Double_value>,
"ssldecferate":<Double_value>,
"ssltottlsv12sessions":<Double_value>,
"ssltotrsa2048keyexchanges":<Double_value>,
"sslbetotsessionmultiplexattemptfails":<Double_value>,
"ssltotoffloadbulkdes":<Double_value>,
"ssltotnullciphers":<Double_value>,
"ssltlsv12sessionsrate":<Double_value>,
"ssltothwdecbe":<Double_value>,
"ssltotoffloadsignrsa":<Double_value>,
"sslswencferate":<Double_value>,
"sslbeecdhecurve224rate":<Double_value>,
"sslbetotshamac":<Double_value>,
"ssltotcipheraes128":<Double_value>,
"ssltotbkendsslv3renego":<Double_value>,
"ssltot128bitrc4ciphers":<Double_value>,
"sslbetotdssauthorizations":<Double_value>,
"sslbetot56bitrc4ciphers":<Double_value>,
"sslbetot40bitdesciphers":<Double_value>,
"ssltotsslv3renegsessions":<Double_value>,
"sslbersaauthorizationsrate":<Double_value>,
"ssltot128bitaesgcmciphers":<Double_value>,
"ssltotsslv2transactions":<Double_value>,
"sslbedh1024keyexchangesrate":<Double_value>,
"ssl64bitrc4ciphersrate":<Double_value>,
"sslsslv3clientauthenticationsrate":<Double_value>,
"sslbetottlsv11sessions":<Double_value>,
"sslsessionmissrate":<Double_value>,
"sslbetotdhauthorizations":<Double_value>,
"sslbemd5macrate":<Double_value>,
"sslecdhe256keyexchangesrate":<Double_value>,
"sslbetotmd5mac":<Double_value>,
"ssldhauthorizationsrate":<Double_value>,
"ssloffloadbulkaesgcm128rate":<Double_value>,
"sslshamacrate":<Double_value>,
"sslhwdecberate":<Double_value>,
"sslnullauthorizationsrate":<Double_value>,
"sslbetot128bitrc2ciphers":<Double_value>,
"sslsslv3sessionsrate":<Double_value>,
"ssltotdh1024keyexchanges":<Double_value>,
"ssltotbkendsessionrenegotiate":<Double_value>,
"sslbkendtlsv12renegorate":<Double_value>,
"sslsessionhitsrate":<Double_value>,
"sslsslv3renegsessionsrate":<Double_value>,
"ssldechwrate":<Double_value>,
"ssltotsslv3clientauthentications":<Double_value>,
"ssldecrate":<Double_value>,
"ssltlsv11clientauthenticationsrate":<Double_value>,
"ssldh2048keyexchangesrate":<Double_value>,
"sslbesslv3sessionsrate":<Double_value>,
"ssltotoffloadbulkaesgcm256":<Double_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

"ssl40bitrc2ciphersrate":<Double_value>,
"sslbe40bitdesciphersrate":<Double_value>,
"sslbetotsessionmultiplexattemptsuccess":<Double_value>,
"ssloffloadrsakeyexchangesrate":<Double_value>,
"sslbetotdh2048keyexchanges":<Double_value>,
"sslbesessionmultiplexattemptsuccessrate":<Double_value>,
"sslbe40bitrc4ciphersrate":<Double_value>,
"ssltotoffloadbulkaes":<Double_value>,
"sslbetot168bit3desciphers":<Double_value>,
"ssltottlsv12renegsessions":<Double_value>,
"ssltotrenegsessions":<Double_value>,
"sslbkendsslv3renegorate":<Double_value>,
"sslsslv3transactionsrate":<Double_value>,
"sslecdhe521keyexchangesrate":<Double_value>,
"ssltottlsv11clientauthentications":<Double_value>,
"sslsslv2clientauthenticationsrate":<Double_value>,
"sslbetotrsa2048keyexchanges":<Double_value>,
"sslrsaauthorizationsrate":<Double_value>,
"ssltot40bitrc2ciphers":<Double_value>,
"ssltot56bitrc2ciphers":<Double_value>,
"ssl128bitrc4ciphersrate":<Double_value>,
"ssltothwencfe":<Double_value>,
"ssltot40bitdesciphers":<Double_value>,
"sslbeshamacrate":<Double_value>,
"sslbkendcipheraes128rate":<Double_value>,
"ssltottlsv1transactions":<Double_value>,
"ssltotrsa512keyexchanges":<Double_value>,
"ssltlsv12renegsessionsrate":<Double_value>,
"sslbetotecdhecurve224":<Double_value>,
"sslbetotnullciphers":<Double_value>,
"sslencrate":<Double_value>,
"ssltotencsw":<Double_value>,
"ssltottlsv1renegsessions":<Double_value>,
"ssltotbkendcipheraes128":<Double_value>,
"ssltlsv11transactionsrate":<Double_value>,
"ssltlsv12clientauthenticationsrate":<Double_value>,
"ssltotcipheraes256":<Double_value>,
"ssltotsslv3transactions":<Double_value>,
"ssltotdecbe":<Double_value>,
"ssloffloadbulkdesrate":<Double_value>,
"ssltotbkendtlsv12renego":<Double_value>,
"sslbesessionsrate":<Double_value>,
"ssltot128bitideaciphers":<Double_value>,
"ssltotdec":<Double_value>,
"ssl56bitdesciphersrate":<Double_value>,
"sslbetot56bitrc2ciphers":<Double_value>,
"sslbkendcipheraes256rate":<Double_value>,
"sslbetotrsaauthorizations":<Double_value>,
"sslbedh2048keyexchangesrate":<Double_value>,
"sslbe168bit3desciphersrate":<Double_value>,
"sslsslv2transactionsrate":<Double_value>,
"sslrsa1024keyexchangesrate":<Double_value>,
"sslbedh512keyexchangesrate":<Double_value>,
"ssltottlsv1clientauthentications":<Double_value>,
"sslbetot40bitrc2ciphers":<Double_value>,
"ssl168bit3desciphersrate":<Double_value>,
"ssltot64bitrc4ciphers":<Double_value>,
"ssltotswencfe":<Double_value>,
"ssldecswrate":<Double_value>,
"ssltot256bitaesgcmciphers":<Double_value>,
"ssltotdhauthorizations":<Double_value>,
"ssltotsessionmiss":<Double_value>,
"ssltotdecfe":<Double_value>,
"sslbesslv3handshakesrate":<Double_value>,
"sslbeecdhecurve521rate":<Double_value>,
"sslencswrate":<Double_value>,
"sslhwencferate":<Double_value>,
"sslbesessionmultiplexattemptfailsrate":<Double_value>,
"sslbe128bitideaciphersrate":<Double_value>,
"ssltotecdhe224keyexchanges":<Double_value>,
"ssltotoffloaddhkeyexchanges":<Double_value>,
"ssltlsv1clientauthenticationsrate":<Double_value>,
"ssltot56bitrc4ciphers":<Double_value>,
"sslbe40bitrc2ciphersrate":<Double_value>,
"sslenchwrate":<Double_value>,
"sslbetlsv1handshakesrate":<Double_value>,
"ssltotsslv3handshakes":<Double_value>,
"sslbenullciphersrate":<Double_value>,
"sslsessionsrate":<Double_value>,
"ssltottlsv12transactions":<Double_value>,
"ssltottlsv11sessions":<Double_value>,
"ssl56bitrc4ciphersrate":<Double_value>,
"ssldecberate":<Double_value>,
"sslbedssauthorizationsrate":<Double_value>,
"ssltot56bitdesciphers":<Double_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

"sslbkendtlsvlrenegorate":<Double_value>,
"sslbenullauthorizationsrate":<Double_value>,
"ssltotenchw":<Double_value>,
"ssltlsv11sessionsrate":<Double_value>,
"sslbetotecdhecurve256":<Double_value>,
"ssltotdh512keyexchanges":<Double_value>,
"ssltlsv1sessionsrate":<Double_value>,
"ssltotencbe":<Double_value>,
"sslcryptoutilizationstat":<Double_value>,
"ssltotecdhe384keyexchanges":<Double_value>,
"sslcipheraes256rate":<Double_value>,
"ssltotsslv2handshakes":<Double_value>,
"ssltotenc":<Double_value>,
"sslbetotsslv3sessions":<Double_value>,
"sslencberate":<Double_value>,
"ssltottlsv11transactions":<Double_value>,
"sslbemaxmultiplexedsessions":<Double_value>,
"ssl128bitaesgcmciphersrate":<Double_value>,
"sslbetotecdhecurve384":<Double_value>,
"ssltotsessions":<Double_value>,
"ssltot40bitrc4ciphers":<Double_value>,
"sslofffloadbulkaesgcm256rate":<Double_value>,
"ssltotecdhe256keyexchanges":<Double_value>,
"sslcardstatus":<Double_value>,
"sslnewsessionsrate":<Double_value>,
"sslbetot128bitideaciphers":<Double_value>,
"ssltransactionsrate":<Double_value>,
"ssltotdechw":<Double_value>,
"sslhwdecferate":<Double_value>,
"ssltotbkendtlsvlrenego":<Double_value>,
"sslbetlsv1sessionsrate":<Double_value>,
"sslbetotsslv3handshakes":<Double_value>,
"ssltlsv12transactionsrate":<Double_value>,
"sslbetlsv12sessionsrate":<Double_value>,
"sslsslv3handshakesrate":<Double_value>,
"sslbetotsessionmultiplexattempts":<Double_value>,
"ssltotdssauthorizations":<Double_value>,
"ssltotshamac":<Double_value>,
"sslrsa4096keyexchangesrate":<Double_value>,
"ssltotswdecfe":<Double_value>,
"ssltotrsa1024keyexchanges":<Double_value>,
"ssltotnullauthorizations":<Double_value>,
"ssltotofffloadrsakeyexchanges":<Double_value>,
"sslbedhauthorizationsrate":<Double_value>,
"ssltotecdhe521keyexchanges":<Double_value>,
"sslmd5macrate":<Double_value>,
"sslswdecferate":<Double_value>,
"sslcards":<Double_value>,
"sslhwencberate":<Double_value>,
"ssltotofffloadbulkaesgcm128":<Double_value>,
"sslnullciphersrate":<Double_value>,
"sslbetot128bitrc4ciphers":<Double_value>,
"ssltotsslv3sessions":<Double_value>,
"ssltotdh2048keyexchanges":<Double_value>,
"ssltlsv1handshakesrate":<Double_value>,
"sslbetotdh1024keyexchanges":<Double_value>,
"ssltotswdecbe":<Double_value>,
"ssltotnewsessions":<Double_value>,
"sslecdhe224keyexchangesrate":<Double_value>,
"ssltottlsv1sessions":<Double_value>,
"ssltot128bitrc2ciphers":<Double_value>,
"sslbe64bitrc4ciphersrate":<Double_value>,
"sslbetotrsa1024keyexchanges":<Double_value>,
"sslbetottlsv1sessions":<Double_value>,
"sslbersa1024keyexchangesrate":<Double_value>,
"ssltotbkendcipheraes256":<Double_value>,
"sslenginestatus":<Double_value>,
"ssltotswencbe":<Double_value>,
"sslbersa512keyexchangesrate":<Double_value>,
"sslbetottlsv12sessions":<Double_value>,
"ssltlsv1transactionsrate":<Double_value>,
"sslbetotecdhecurve521":<Double_value>,
"sslbe56bitdesciphersrate":<Double_value>,
"ssltlsv11renegsessionsrate":<Double_value>,
"ssltlsv12handshakesrate":<Double_value>,
"sslbeecdhecurve384rate":<Double_value>,
"ssl56bitrc2ciphersrate":<Double_value>,
"sslofffloaddhkeyexchangesrate":<Double_value>,
"sslsslv2sessionsrate":<Double_value>,
"sslswencberate":<Double_value>,
"ssl256bitaesgcmciphersrate":<Double_value>,
"ssltottlsv12clientauthentications":<Double_value>,
"sslofffloadsignrsarate":<Double_value>,
"sslencferate":<Double_value>,
"sslbesslv3clientauthenticationsrate":<Double_value>,

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "ssltotoffloadbulkrc4":<Double_value>,
    "ssl128bitideaciphersrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

sslvserver

May 20, 2016

Statistics for SSL virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
vservername	<String>	Read-write	Name of the SSL virtual server for which to show detailed statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
vslbhealth	<Double>	Read-only	Health of the vserver. This gives percentage of UP services bound to this vserver.
primaryipaddress	<String>	Read-only	IP address of the vserver
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
actsvcs	<Double>	Read-only	number of ACTIVE services bound to a vserver
ssltotclientauthsuccess	<Double>	Read-only	Number of successful client authentication on this vserver
ssltotclientauthfailure	<Double>	Read-only	Number of failure client authentication on this vserver

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/sslvserver

Query-parameters:

args

http://<netscaler-ip-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

address>/nitro/v1/stat/sslvserver?**args=vservername:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<St**

Use this query-parameter to get sslvserver resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslvserver": [ {
    "vservername":<String_value>,
    "state":<String_value>,
    "vslbhealth":<Double_value>,
    "actsvcs":<Double_value>,
    "primaryipaddress":<String_value>,
    "type":<String_value>,
    "primaryport":<Integer_value>,
    "ssltotclientauthfailure":<Double_value>,
    "ssltotclientauthsuccess":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/sslvserver/vservername_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "sslvserver": [ {
    "vservername":<String_value>,
    "state":<String_value>,
    "vslbhealth":<Double_value>,
    "actsvcs":<Double_value>,
    "primaryipaddress":<String_value>,
    "type":<String_value>,
    "primaryport":<Integer_value>,
    "ssltotclientauthfailure":<Double_value>,
    "ssltotclientauthsuccess":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Stream

May 20, 2016

Stream statistics.

Name	Description
streamidentifier	Statistics for identifier resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

streamidentifier

May 20, 2016

Statistics for identifier resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the stream identifier. Minimum length = 1
pattern	<String[]>	Read-write	Values on which grouping is performed are displayed in the output as row titles. If grouping is performed on two or more fields, their values are separated by a question mark in the row title. For example, consider a selector that contains the expressions HTTP.REQ.URL and CLIENT.IP.SRC (in that order), on an appliance that has accumulated records of a number of requests for two URLs, example.com/page1.html and example.com/page2.html, from two client IP addresses, 192.0.2.10 and 192.0.2.11. With a pattern of ? ?, the appliance performs grouping on both fields and displays statistics for the following: * Requests for example.com/abc.html from 192.0.2.10, with a row title of example.com/abc.html?192.0.2.10. * Requests for example.com/abc.html from 192.0.2.11, with a row title of example.com/abc.html?192.0.2.11. * Requests for example.com/def.html from 192.0.2.10, with a row title of example.com/def.html?192.0.2.10. * Requests for example.com/def.html from 192.0.2.11, with a row title of example.com/def.html?192.0.2.11. With a pattern of * ?, the appliance performs grouping on only the client IP address values and displays statistics for the following requests: * All requests from 192.0.2.10, with the IP address as the row title. * All requests from 192.0.2.11, with the IP address as the row title. With a pattern of ? *, the appliance performs grouping on only the URL values and displays statistics for the following requests: * All requests for example.com/abc.html, with the URL as the row title. * All requests for example.com/def.html, with the URL as the row title. With a pattern of * *, the appliance displays one set of collective statistics for all the requests received, with no row title. With a pattern of example.com/abc.html ?, the appliance displays statistics for requests for example.com/abc.html from each unique client IP address. With a pattern of * 192.0.2.11, the appliance displays statistics for all requests from 192.0.2.11.
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
sortby	<String>	Read-write	use this argument to sort by specific key. Possible values =
sortorder	<String>	Read-write	use this argument to specify sort order. Default value: SORT_DESCENDING Possible values = ascending, descending
streamobjreq	<Double>	Read-only	Total number of Stream Requests recieved.
streamobjbandw	<Double>	Read-only	Total Bandwidth consumed.
streamobjresptime	<Double>	Read-only	Average response time of the stream session.
streamobjconn	<Double>	Read-only	Current connections on the stream session.
streamobjbreachcnt	<Double>	Read-only	Total Breaching Transactions in configured interval.

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

X-NITRO-USER:*<username>*

X-NITRO-PASS:*<password>*

Note: In such cases, make sure that the request header DOES not include the following:

Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/streamidentifier

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/streamidentifier?**args=name:<String_value>,pattern:<String[]_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>,sortby:<Strin**

Use this query-parameter to get streamidentifier resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "streamidentifier": [ {
    "name":<String_value>,
    "streamobjconn":<Double_value>,
    "streamobjbreachcnt":<Double_value>,
    "streamobjbandw":<Double_value>,
    "streamobjresptime":<Double_value>,
    "streamobjreq":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/streamidentifier/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "streamidentifier": [ {
    "name":<String_value>,
    "streamobjconn":<Double_value>,
    "streamobjbreachcnt":<Double_value>,
    "streamobjbandw":<Double_value>,
    "streamobjresptime":<Double_value>,
    "streamobjreq":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

System

May 20, 2016

System statistics.

Name	Description
system	Statistics for system.
systembw	Statistics for bw resource.
systemcpu	Statistics for cpu resource.
systemmemory	Statistics for Global system memory stats resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

system

May 20, 2016

Statistics for system.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
voltagev12n	<Double>	Read-only	Power supply -12V output. Acceptable range is -13.20 through -10.80 volts. 9800 and 9960 platforms display standard value of -12.0V.
voltagev5n	<Double>	Read-only	Power supply -5V output. Acceptable range is -5.50 through -4.50 volts. 9800 and 9960 platforms display standard value of -5.0V.
cpuusage	<Double>	Read-only	CPU utilization percentage.
rescpuusage	<Double>	Read-only	Shows average CPU utilization percentage if more than 1 CPU is present.
slavecpuusage	<Double>	Read-only	CPU 1 (currently the slave CPU) utilization, as percentage of capacity. Not applicable for a single-CPU system.
mastercpuusage	<Double>	Read-only	CPU 0 (currently the master CPU) utilization, as percentage of capacity.
auxvolt7	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 7.
auxvolt6	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 6.
auxvolt5	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 5.
auxvolt4	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 4.
auxvolt3	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 3.
auxvolt2	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 2.
auxvolt1	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 1.
auxvolt0	<Double>	Read-only	Voltage of a device connected to health monitoring chip through pin 0.
voltagevsen2	<Double>	Read-only	Voltage Sensor 2 Input. Currently only 13k Platforms will have valid value for this counter and for older platforms this will be 0.
voltagev5sb	<Double>	Read-only	Power Supply 5V Standby Voltage. Currently only 13k Platforms will have valid value for this counter and for older platforms this will be 0.
voltagevtt	<Double>	Read-only	Intel CPU Vtt power. Currently only 13k Platforms will have valid value for this counter and for older platforms this will be 0.
voltagevbat	<Double>	Read-only	Onboard battery power supply output. 9800 and 9950 platforms display standard value of 5.0V.
voltagev12p	<Double>	Read-only	Power supply +12V output. Acceptable range is 10.80 through 13.20 volts.
voltagev5p	<Double>	Read-only	Power supply +5V output. Acceptable range is 4.50 through 5.50 volts.
			Standby power supply +3.3V output. Acceptable range is 2.970 through 3.630 volts. 9800 and 9960 platforms display standard value of 3.3V. You

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
voltagev33main	<Double>	Read-only	Main power supply +3.3V output. Acceptable range is 2.970 through 3.630 volts. This is a critical counter. You can configure Main 3.3V Supply Voltage, by using the Set snmp alarm VOLTAGE-LOW command to set the lower limit and the Set snmp alarm VOLTAGE-HIGH command to set the upper limit.
voltagevcc1	<Double>	Read-only	CPU core 1 voltage. Acceptable range is 1.080 through 1.650 volts. If CPU 1 is not connected to the health monitoring chip, display shows voltage of CPU 0.
voltagevcc0	<Double>	Read-only	CPU core 0 voltage. Acceptable range is 1.080 through 1.650 volts.
numcpus	<Double>	Read-only	The number of CPUs on the NetScaler appliance.
memusagepcnt	<Double>	Read-only	Percentage of memory utilization on NetScaler.
memuseinmb	<Double>	Read-only	Main memory currently in use, in megabytes.
mgmtcpuusagepcnt	<Double>	Read-only	Management CPU utilization percentage.
pktcpuusagepcnt	<Double>	Read-only	Average CPU utilization percentage for all packet engines excluding management PE.
cpuusagepcnt	<Double>	Read-only	CPU utilization percentage.
rescpuusagepcnt	<Double>	Read-only	Average CPU utilization percentage. Not applicable for a single-CPU system.
starttime	<String>	Read-only	Time when the NetScaler appliance was last started.
disk0perusage	<Double>	Read-only	Used space in /flash partition of the disk, as a percentage. This is a critical counter. You can configure /flash Used (%) by using the Set snmp alarm DISK-USAGE-HIGH command.
disk1perusage	<Double>	Read-only	Used space in /var partition of the disk, as a percentage. This is a critical counter. You can configure /var Used (%) by using the Set snmp alarm DISK-USAGE-HIGH command.
cpufan0speed	<Double>	Read-only	CPU Fan 0 speed. Acceptable range is 3000 through 6000 RPM. This is a critical counter. You can configure CPU Fan 0 Speed by using the Set snmp alarm FAN-SPEED-LOW command to set the lower limit.
cpufan1speed	<Double>	Read-only	CPU Fan 1 speed. Acceptable range is 3000 through 6000 RPM. 7000 platform displays speed of CPU fan 0. This is a critical counter. You can configure CPU Fan 1 Speed by using the Set snmp alarm FAN-SPEED-LOW command to set the lower limit.
systemfanspeed	<Double>	Read-only	System fan speed. Acceptable range is 3000 through 6000 RPM. This is a critical counter. You can configure System Fan Speed by using the Set snmp alarm FAN-SPEED-LOW command to set the lower limit.
fan0speed	<Double>	Read-only	System fan 1 speed. For new platforms associated pin is connected to CPU supporting fans. For platforms in which it is not connected, it will point to System Fan.
fanspeed	<Double>	Read-only	System fan 2 speed. For new platforms associated pin is connected to CPU supporting fans. For platforms in which it is not connected, it will point to System Fan
cpu0temp	<Double>	Read-only	CPU 0 temperature. 9800 and 9960 platforms display internal chip temperature. This is a critical counter. You can configure CPU 0 Temperature by using the Set snmp alarm TEMPERATURE-HIGH command to set the upper limit.
cpu1temp	<Double>	Read-only	CPU 1 temperature. 9800 and 9960 platforms display internal chip temperature. 7000, 9010 and 10010 platforms display CPU 0 temperature. This is a critical counter. You can configure CPU 1 Temperature by using the Set snmp alarm TEMPERATURE-HIGH command to set the upper limit.
internaltemp	<Double>	Read-only	Internal temperature of health monitoring chip. This is a critical counter. You can configure Internal Temperature by using the Set snmp alarm TEMPERATURE-HIGH command to set the upper limit.
powersupply1status	<String>	Read-only	Power supply 1 failure status.
powersupply2status	<String>	Read-only	Power supply 2 failure status.
powersupply3status	<String>	Read-only	Power supply 3 failure status.
powersupply4status	<String>	Read-only	Power supply 4 failure status.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
disk0used	<Double>	Read-only	Used space in /flash partition of the hard disk.
disk0avail	<Double>	Read-only	Available space in /flash partition of the hard disk.
disk1size	<Double>	Read-only	Size of /var partition of the hard disk.
disk1used	<Double>	Read-only	Used space in /var partition of the hard disk.
disk1avail	<Double>	Read-only	Available space in /var partition of the hard disk.
fan2speed	<Double>	Read-only	Speed of Fan 0 if associated pin is connected to health monitoring chip.
fan3speed	<Double>	Read-only	Speed of Fan 1 if associated pin is connected to health monitoring chip.
fan4speed	<Double>	Read-only	Speed of Fan 2 if associated pin is connected to health monitoring chip.
fan5speed	<Double>	Read-only	Speed of Fan 3 if associated pin is connected to health monitoring chip.
auxtemp0	<Double>	Read-only	Temperature of a device connected to health monitoring chip through pin 0.
auxtemp1	<Double>	Read-only	Temperature of a device connected to health monitoring chip through pin 1.
auxtemp2	<Double>	Read-only	Temperature of a device connected to health monitoring chip through pin 2.
auxtemp3	<Double>	Read-only	Temperature of a device connected to health monitoring chip through pin 3.
timesincestart	<String>	Read-only	Seconds since the NetScaler appliance started.
memsizemb	<Double>	Read-only	Total amount of system memory, in megabytes.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/system

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/system?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get system resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "system": [ {
    "voltagevbat":<Double_value>,
    "auxvolt2":<Double_value>,
    "voltagev5n":<Double_value>,
    "auxvolt7":<Double_value>,
    "cpu0temp":<Double_value>,
    "fan0speed":<Double_value>,
    "disk1perusage":<Double_value>,
    "rescpuusagepcnt":<Double_value>,
    "disk1used":<Double_value>,
    "disk1avail":<Double_value>,
    "fan4speed":<Double_value>,
    "rescpuusage":<Double_value>,
    "powersupply3status":<String_value>,
    "auxvolt5":<Double_value>,
    "auxvolt3":<Double_value>,
    "disk0perusage":<Double_value>,
    "fan2speed":<Double_value>,
    "powersupply4status":<String_value>,
    "auxvolt1":<Double_value>,
    "fanspeed":<Double_value>,
    "fan5speed":<Double_value>,
    "disk0size":<Double_value>,
    "mgmtcpuusagepcnt":<Double_value>,
    "cpuusage":<Double_value>,
    "voltagev5sb":<Double_value>,
    "disk0used":<Double_value>,
    "powersupply1status":<String_value>,
    "cpufan0speed":<Double_value>,
    "disk1size":<Double_value>,
    "auxtemp1":<Double_value>,
    "numcpus":<Double_value>,
    "pktcpuusagepcnt":<Double_value>,
    "voltagev5p":<Double_value>,
    "voltagevsen2":<Double_value>,
    "auxvolt0":<Double_value>,
    "auxtemp2":<Double_value>,
    "memsizemb":<Double_value>,
    "voltagev33main":<Double_value>,
    "cpu1temp":<Double_value>,
    "voltagev12n":<Double_value>,
    "memuseinmb":<Double_value>,
    "auxtemp3":<Double_value>,
    "internaltemp":<Double_value>,
    "voltagev12p":<Double_value>,
    "disk0avail":<Double_value>,
    "voltagev33stby":<Double_value>,
    "voltagevcc1":<Double_value>,
    "fan3speed":<Double_value>,
    "voltagevtt":<Double_value>,
    "auxtemp0":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "cpufan1speed":<Double_value>,
    "voltagevcc0":<Double_value>,
    "auxvolt4":<Double_value>,
    "starttime":<String_value>,
    "systemfanspeed":<Double_value>,
    "cpuusagepcnt":<Double_value>,
    "mastercpuusage":<Double_value>,
    "timesincestart":<String_value>,
    "auxvolt6":<Double_value>,
    "slavecpuusage":<Double_value>,
    "memusagepcnt":<Double_value>,
    "powersupply2status":<String_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systembw

Jul 07, 2017
Statistics for bw resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
httpcltpoolinactive	<Double>	Read-only	No of requests sent from BW client.
httpcltpooloutactive	<Double>	Read-only	No of responses Received.
httpsvr200okresp	<Double>	Read-only	Number of 200 Ok response sent from the BW appliance.
httpsvr200okresprate	<Double>	Read-only	Rate (/s) counter for httpsvr200okresp
httpsvr404notfound	<Double>	Read-only	Number of 404 Not Found responses sent
httpsvr404notfoundrate	<Double>	Read-only	Rate (/s) counter for httpsvr404notfound
httpclterrstray	<Double>	Read-only	Number of stray packets received from server without HTTP request
httpclterrstrayrate	<Double>	Read-only	Rate (/s) counter for httpclterrstray
httpclttfplwm	<Double>	Read-only	Number of Responses Falling on LWM for TTFP.
httpclttfplwmrate	<Double>	Read-only	Rate (/s) counter for httpclttfplwm
httpclttfp_0	<Double>	Read-only	Number of Responses Falling on Band-0 for TTFP.
httpclttfp_0rate	<Double>	Read-only	Rate (/s) counter for httpclttfp_0
httpclttfp_1	<Double>	Read-only	Number of Responses Falling on Band-1 for TTFP.
httpclttfp_1rate	<Double>	Read-only	Rate (/s) counter for httpclttfp_1
httpclttfp_2	<Double>	Read-only	Number of Responses Falling on Band-2 for TTFP.
httpclttfp_2rate	<Double>	Read-only	Rate (/s) counter for httpclttfp_2
httpclttfp_3	<Double>	Read-only	Number of Responses Falling on Band-3 for TTFP.
httpclttfp_3rate	<Double>	Read-only	Rate (/s) counter for httpclttfp_3
httpclttfp_4	<Double>	Read-only	Number of Responses Falling on Band-4 for TTFP.
httpclttfp_4rate	<Double>	Read-only	Rate (/s) counter for httpclttfp_4
httpclttfp_5	<Double>	Read-only	Number of Responses Falling on Band-5 for TTFP.
httpclttfp_5rate	<Double>	Read-only	Rate (/s) counter for httpclttfp_5

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
httpclttfp_6	<Double>	Read-only	Number of Responses Falling on Band-6 for TTFP.
httpclttfp_6rate	<Double>	Read-only	Rate (/s) counter for httpclttfp_6
httpclttfp_7	<Double>	Read-only	Number of Responses Falling on Band-7 for TTFP.
httpclttfp_7rate	<Double>	Read-only	Rate (/s) counter for httpclttfp_7
httpclttfphwm	<Double>	Read-only	Number of Responses Falling on HWM for TTFP.
httpclttfphwmrate	<Double>	Read-only	Rate (/s) counter for httpclttfphwm
httpclttfpmax	<Double>	Read-only	Peak RTT observed for Time to First response packet.
httpclttlplwm	<Double>	Read-only	Number of Responses Falling on LWM for TTLP.
httpclttlplwmrate	<Double>	Read-only	Rate (/s) counter for httpclttlplwm
httpclttlp_0	<Double>	Read-only	Number of Responses Falling on Band-0 for TTLP.
httpclttlp_0rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_0
httpclttlp_1	<Double>	Read-only	Number of Responses Falling on Band-1 for TTLP.
httpclttlp_1rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_1
httpclttlp_2	<Double>	Read-only	Number of Responses Falling on Band-2 for TTLP.
httpclttlp_2rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_2
httpclttlp_3	<Double>	Read-only	Number of Responses Falling on Band-3 for TTLP.
httpclttlp_3rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_3
httpclttlp_4	<Double>	Read-only	Number of Responses Falling on Band-4 for TTLP.
httpclttlp_4rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_4
httpclttlp_5	<Double>	Read-only	Number of Responses Falling on Band-5 for TTLP.
httpclttlp_5rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_5
httpclttlp_6	<Double>	Read-only	Number of Responses Falling on Band-6 for TTLP.
httpclttlp_6rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_6
httpclttlp_7	<Double>	Read-only	Number of Responses Falling on Band-7 for TTLP.
httpclttlp_7rate	<Double>	Read-only	Rate (/s) counter for httpclttlp_7
httpclttlphwm	<Double>	Read-only	Number of Responses Falling on HWM for TTLP.
httpclttlphwmrate	<Double>	Read-only	Rate (/s) counter for httpclttlphwm

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
[redacted]	[redacted]	[redacted]	[redacted]

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/systembw

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/systembw?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get systembw resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systembw": [ {
    "httpcltttfp_4rate":<Double_value>,
    "httpcltttlp_5rate":<Double_value>,
    "httpcltttfp_2rate":<Double_value>,
    "httpcltttfp_7rate":<Double_value>,
    "httpcltttlphwmrate":<Double_value>,
    "httpcltttfp_0rate":<Double_value>,
    "httpcltttlplwmrate":<Double_value>,
    "httpcltttfp_4":<Double_value>,
    "httpcltpooloutactive":<Double_value>,
    "httpcltttlplwm":<Double_value>,
    "httpcltttlp_1":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "httpsvr404notfoundrate":<Double_value>,
    "httpclttlp_6":<Double_value>,
    "httpclttlphwm":<Double_value>,
    "httpclttlp_1rate":<Double_value>,
    "httpcltpoolinactive":<Double_value>,
    "httpclttlp_3":<Double_value>,
    "httpclttlp_5":<Double_value>,
    "httpclttlp_6rate":<Double_value>,
    "httpsvr200okresprate":<Double_value>,
    "httpclttlfp_1rate":<Double_value>,
    "httpclttlfp_1":<Double_value>,
    "httpclttlfphwmrate":<Double_value>,
    "httpclttlfp_6":<Double_value>,
    "httpclttlfpmax":<Double_value>,
    "httpclttlfp_5":<Double_value>,
    "httpclttlpmax":<Double_value>,
    "httpclttlp_4rate":<Double_value>,
    "httpclttlp_7rate":<Double_value>,
    "httpclttlfp_0":<Double_value>,
    "httpclttlfp_3":<Double_value>,
    "httpclttlfphwm":<Double_value>,
    "httpclterrstray":<Double_value>,
    "httpclttlp_0rate":<Double_value>,
    "httpclttlfplwm":<Double_value>,
    "httpclttlp_0":<Double_value>,
    "httpclterrstrayrate":<Double_value>,
    "httpclttlfplwmrate":<Double_value>,
    "httpclttlp_2":<Double_value>,
    "httpclttlfp_2":<Double_value>,
    "httpclttlp_2rate":<Double_value>,
    "httpclttlp_4":<Double_value>,
    "httpclttlfp_7":<Double_value>,
    "httpsvr200okresp":<Double_value>,
    "httpsvr404notfound":<Double_value>,
    "httpclttlp_3rate":<Double_value>,
    "httpclttlfp_6rate":<Double_value>,
    "httpclttlfp_5rate":<Double_value>,
    "httpclttlfp_3rate":<Double_value>,
    "httpclttlp_7":<Double_value>
}}}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemcpu

May 20, 2016

Statistics for cpu resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
id	<Double>	Read-write	ID of the CPU for which to display statistics. Default value: 65535 Minimum value = 0 Maximum value = 65534
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
percpuuse	<Double>	Read-only	CPU utilization percentage.

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/systemcpu

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/systemcpu?**args=id:<Double_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_valu**

Use this query-parameter to get systemcpu resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "systemcpu": [ {
    "id":<Double_value>,
    "percpuuse":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/systemcpu/id_value><Double>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemcpu": [ {
    "id":<Double_value>,
    "percpuuse":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

systemmemory

May 20, 2016

Statistics for Global system memory stats resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
shmemallocpcnt	<Double>	Read-only	Shared memory insue percent.
shmemallocinmb	<Double>	Read-only	Shared memory insue, in megabytes.
shmemtotinmb	<Double>	Read-only	Total shared memory allowed to allocate, in megabytes.
memtotfree	<Double>	Read-only	Total Free PE Memory in the System.
memusagepcnt	<Double>	Read-only	Percentage of memory utilization on NetScaler.
memtotuseinmb	<Double>	Read-only	Total NetScaler Memory in use, in megabytes.
memtotallocpcnt	<Double>	Read-only	Currently allocated memory in percent.
memtotallocmb	<Double>	Read-only	Currently allocated memory, in megabytes.
memtotinmb	<Double>	Read-only	Total memory available (grabbed) for use by packet engine (PE), in megabytes.
memtotavail	<Double>	Read-only	Total system memory available for PE to grab from the system.

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URL: http://<netscaler-ip-address>/nitro/v1/stat/systemmemory

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/systemmemory?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get systemmemory resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "systemmemory": [ {
    "memtotallocpcnt":<Double_value>,
    "memtotfree":<Double_value>,
    "shmemallocpcnt":<Double_value>,
    "shmemallocinmb":<Double_value>,
    "memtotuseinmb":<Double_value>,
    "memusagepcnt":<Double_value>,
    "memtotinmb":<Double_value>,
    "memtotallocmb":<Double_value>,
    "memtotavail":<Double_value>,
    "shmemtotinmb":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Traffic Management

May 20, 2016

TM session/policy statistics.

Name	Description
tmsessionpolicy	Statistics for TM session policy resource.
tmtrafficpolicy	Statistics for TM traffic policy resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmsessionpolicy

May 20, 2016

Statistics for TM session policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the advanced tmsession policy for which to display statistics. If no name is specified, statistics for all advanced tmsession polices are shown. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/tmsessionpolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/tmsessionpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<Stri**

Use this query-parameter to get tmsessionpolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/tmsessionpolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmsessionpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

tmtrafficpolicy

May 20, 2016

Statistics for TM traffic policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	The name of the TM traffic policy for which statistics will be displayed. If not given statistics are shown for all policies. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/tmtrafficpolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/tmtrafficpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<Strin**

Use this query-parameter to get tmtrafficpolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "tmtrafficpolicy": [ {
    "name":<String_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/tmtrafficpolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "tmtrafficpolicy": [ {
    "name":<String_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Transform

May 20, 2016

URL Transformation statistics.

Name	Description
transformpolicy	Statistics for URL Transformation policy resource.
transformpolicylabel	Statistics for transform policy label resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicy

May 20, 2016

Statistics for URL Transformation policy resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the policy. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicyhits	<Double>	Read-only	Number of hits on the policy
pipolicyhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyhits
pipolicyundefhits	<Double>	Read-only	Number of undef hits on the policy
pipolicyundefhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicyundefhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/transformpolicy

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/transformpolicy?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<Stri**

Use this query-parameter to get transformpolicy resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

---

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/transformpolicy/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicy": [ {
    "name":<String_value>,
    "pipolicyundefhitsrate":<Double_value>,
    "pipolicyundefhits":<Double_value>,
    "pipolicyhitsrate":<Double_value>,
    "pipolicyhits":<Double_value>
}]}
```

© 1999-2017 Citrix Systems, Inc. All rights reserved.

transformpolicylabel

May 20, 2016

Statistics for transform policy label resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
labelname	<String>	Read-write	The name of the URL Transformation policy label. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
pipolicylabelhits	<Double>	Read-only	Number of times policy label was invoked.
pipolicylabelhitsrate	<Double>	Read-only	Rate (/s) counter for pipolicylabelhits

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/transformpolicylabel

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/transformpolicylabel?**args=labelname:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearst**

Use this query-parameter to get transformpolicylabel resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicylabel": [ {
    "labelname":<String_value>,
    "pipolicylabelhitsrate":<Double_value>,
    "pipolicylabelhits":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/transformpolicylabel/labelname_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "transformpolicylabel": [ {
    "labelname":<String_value>,
    "pipolicylabelhitsrate":<Double_value>,
    "pipolicylabelhits":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL VPN

May 20, 2016

Virtual Private Network statistics.

Name	Description
vpn	Statistics for vpn.
vpnvserver	Statistics for VPN virtual server resource.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpn

May 20, 2016
Statistics for vpn.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
indexhtmlhit	<Double>	Read-only	Number of requests for VPN login page.
indexhtmlnoserved	<Double>	Read-only	Number of failures to display VPN login page.
cfghtmlserved	<Double>	Read-only	Number of client configuration requests received by VPN server.
cfghtmlservedrate	<Double>	Read-only	Rate (/s) counter for cfghtmlserved
dnsreqhit	<Double>	Read-only	Number of DNS queries resolved by VPN server.
dnsreqhitrate	<Double>	Read-only	Rate (/s) counter for dnsreqhit
winsrequesthit	<Double>	Read-only	Number of WINS queries resolved by VPN server.
winsrequesthitrate	<Double>	Read-only	Rate (/s) counter for winsrequesthit
csrequesthit	<Double>	Read-only	Number of SSL VPN tunnels formed between VPN server and client.
csrequesthitrate	<Double>	Read-only	Rate (/s) counter for csrequesthit
csnonhttpprobehit	<Double>	Read-only	Number of probes from VPN to back-end non-HTTP servers that have been accessed by the VPN client.
csnonhttpprobehitrate	<Double>	Read-only	Rate (/s) counter for csnonhttpprobehit
cshttpprobehit	<Double>	Read-only	Number of probes from VPN to back-end HTTP servers that have been accessed by the VPN client.
cshttpprobehitrate	<Double>	Read-only	Rate (/s) counter for cshttpprobehit
totalcsconnsucc	<Double>	Read-only	Number of successful probes to all back-end servers.
csconnsuccrate	<Double>	Read-only	Rate (/s) counter for totalcsconnsucc
totalfsrequest	<Double>	Read-only	Number of file system requests received by VPN server.
fsrequestrate	<Double>	Read-only	Rate (/s) counter for totalfsrequest
iipdisabledmipused	<Double>	Read-only	Number of times MIP is used as IIP is disabled.
iipdisabledmipusedrate	<Double>	Read-only	Rate (/s) counter for iipdisabledmipused

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
iipfailedmipusedrate	<Double>	Read-only	Rate (/s) counter for iipfailedmipused
iipspillovermipused	<Double>	Read-only	Number of times MIP is used on IIP Spillover.
iipspillovermipusedrate	<Double>	Read-only	Rate (/s) counter for iipspillovermipused
iipdisabledmipdisabled	<Double>	Read-only	Both IIP and MIP is disabled.
iipdisabledmipdisabledrate	<Double>	Read-only	Rate (/s) counter for iipdisabledmipdisabled
iipfailedmipdisabled	<Double>	Read-only	Number of times IIP assignment failed and MIP is disabled.
iipfailedmipdisabledrate	<Double>	Read-only	Rate (/s) counter for iipfailedmipdisabled
socksmethreqrcvd	<Double>	Read-only	Number of received SOCKS method request.
socksmethreqrcvdrate	<Double>	Read-only	Rate (/s) counter for socksmethreqrcvd
socksmethreqsent	<Double>	Read-only	Number of sent SOCKS method request.
socksmethreqsentrate	<Double>	Read-only	Rate (/s) counter for socksmethreqsent
socksmethresprcvd	<Double>	Read-only	Number of received SOCKS method response.
socksmethresprcvdrate	<Double>	Read-only	Rate (/s) counter for socksmethresprcvd
socksmethrespsent	<Double>	Read-only	Number of sent SOCKS method response.
socksmethrespsentrate	<Double>	Read-only	Rate (/s) counter for socksmethrespsent
socksconnreqrcvd	<Double>	Read-only	Number of received SOCKS connect request.
socksconnreqrcvdrate	<Double>	Read-only	Rate (/s) counter for socksconnreqrcvd
socksconnreqsent	<Double>	Read-only	Number of sent SOCKS connect request.
socksconnreqsentrate	<Double>	Read-only	Rate (/s) counter for socksconnreqsent
socksconnresprcvd	<Double>	Read-only	Number of received SOCKS connect response.
socksconnresprcvdrate	<Double>	Read-only	Rate (/s) counter for socksconnresprcvd
socksconnrespsent	<Double>	Read-only	Number of sent SOCKS connect response.
socksconnrespsentrate	<Double>	Read-only	Rate (/s) counter for socksconnrespsent
socksservererror	<Double>	Read-only	Number of SOCKS server error.
socksservererrorrate	<Double>	Read-only	Rate (/s) counter for socksservererror
socksclienterror	<Double>	Read-only	Number of SOCKS client error.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Name	Data Type	Permissions	Description
staconnsuccess	<Double>	Read-only	Number of STA connection success.
staconnsuccessrate	<Double>	Read-only	Rate (/s) counter for staconnsuccess
staconnfailure	<Double>	Read-only	Number of STA connection failure.
staconnfailurerate	<Double>	Read-only	Rate (/s) counter for staconnfailure
cpsconnsuccess	<Double>	Read-only	Number of CPS connection success.
cpsconnsuccessrate	<Double>	Read-only	Rate (/s) counter for cpsconnsuccess
cpsconnfailure	<Double>	Read-only	Number of CPS connection failure.
cpsconnfailurerate	<Double>	Read-only	Rate (/s) counter for cpsconnfailure
starequestsent	<Double>	Read-only	Number of STA request sent.
starequestsentrate	<Double>	Read-only	Rate (/s) counter for starequestsent
staresponserecvd	<Double>	Read-only	Number of STA response received.
staresponserecvdrate	<Double>	Read-only	Rate (/s) counter for staresponserecvd
icalicensefailure	<Double>	Read-only	Number of ICA license failure.
icalicensefailurerate	<Double>	Read-only	Rate (/s) counter for icalicensefailure

Operations (click to see Properties)

GET (ALL)

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/vpn

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/vpn?**args=detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_value>**

Use this query-parameter to get vpn resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpn": [ {
    "staconnfailurerate":<Double_value>,
    "totalcsconnsucc":<Double_value>,
    "csnonhttpprobehitrate":<Double_value>,
    "socksmethreqrcvd":<Double_value>,
    "iipspillovermipusedrate":<Double_value>,
    "iipfailedmipdisabled":<Double_value>,
    "cshttpprobehitrate":<Double_value>,
    "socksconnrespsentrate":<Double_value>,
    "starequestsent":<Double_value>,
    "socksserverrorrate":<Double_value>,
    "winsrequesthitrate":<Double_value>,
    "socksserverror":<Double_value>,
    "socksmethresprcvdrate":<Double_value>,
    "socksconnreqrcvd":<Double_value>,
    "totalfsrequest":<Double_value>,
    "socksmethreqrcvdrate":<Double_value>,
    "socksmethreqsentrate":<Double_value>,
    "cpsconnfailurerate":<Double_value>,
    "iipdisabledmipdisabled":<Double_value>,
    "cshttpprobehit":<Double_value>,
    "iipspillovermipused":<Double_value>,
    "cpsconnsuccess":<Double_value>,
    "iipdisabledmipusedrate":<Double_value>,
    "socksmethrespsentrate":<Double_value>,
    "socksmethreqsent":<Double_value>,
    "icalicensefailurerate":<Double_value>,
    "staconnfailure":<Double_value>,
    "socksconnreqrcvdrate":<Double_value>,
    "socksclienterrorrate":<Double_value>,
    "csrequesthit":<Double_value>,
    "winsrequesthit":<Double_value>,
    "dnsreqhitrate":<Double_value>,
    "socksconnresprcvd":<Double_value>,
    "socksmethrespsent":<Double_value>,
    "csconnsuccrate":<Double_value>,
    "socksconnreqsent":<Double_value>,
    "dnsreqhit":<Double_value>,
    "staconnsuccessrate":<Double_value>,
    "iipfailedmipusedrate":<Double_value>,
    "staconnsuccess":<Double_value>,
    "cfghtmlservedrate":<Double_value>,
    "staresponserecvdrate":<Double_value>,
    "cpsconnfailure":<Double_value>,
    "csrequesthitrate":<Double_value>,
    "csnonhttpprobehit":<Double_value>,
    "iipfailedmipdisabledrate":<Double_value>,
    "cpsconnsuccessrate":<Double_value>,
    "socksconnresprcvdrate":<Double_value>,
    "indexhtmlnoserved":<Double_value>,
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
    "fsrequestrate":<Double_value>,
    "indexhtmlhit":<Double_value>,
    "cfghtmlserved":<Double_value>,
    "icalicensefailure":<Double_value>,
    "staresponserecvd":<Double_value>,
    "socksclienterror":<Double_value>,
    "iipdisabledmipdisabledrate":<Double_value>,
    "socksconnrespsent":<Double_value>,
    "socksmethresprcvd":<Double_value>,
    "iipdisabledmipused":<Double_value>,
    "iipfailedmipused":<Double_value>,
    "socksconnreqsentrate":<Double_value>,
    "starequestsentrate":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

vpnvserver

Jul 07, 2017

Statistics for VPN virtual server resource.

Properties (click to see Operations)

Name	Data Type	Permissions	Description
name	<String>	Read-write	Name of the virtual server for which to show detailed statistics. Minimum length = 1
clearstats	<String>	Read-write	Clear the statsistics / counters. Possible values = basic, full
primaryipaddress	<String>	Read-only	The IP address on which the service is running.
primaryport	<Integer>	Read-only	The port on which the service is running.
type	<String>	Read-only	Protocol associated with the vserver
state	<String>	Read-only	Current state of the server. Possible values are UP, DOWN, UNKNOWN, OFS(Out of Service), TROFS(Transition Out of Service), TROFS_DOWN(Down When going Out of Service)
totalrequests	<Double>	Read-only	Total number of requests received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
requestsrate	<Double>	Read-only	Rate (/s) counter for totalrequests
totalresponses	<Double>	Read-only	Number of responses received on this service or virtual server. (This applies to HTTP/SSL services and servers.)
responsesrate	<Double>	Read-only	Rate (/s) counter for totalresponses
totalrequestbytes	<Double>	Read-only	Total number of request bytes received on this service or virtual server.
requestbytesrate	<Double>	Read-only	Rate (/s) counter for totalrequestbytes
totalresponsebytes	<Double>	Read-only	Number of response bytes received by this service or virtual server.
responsebytesrate	<Double>	Read-only	Rate (/s) counter for totalresponsebytes

Operations (click to see Properties)

GET (ALL) | GET

Some options that you can use for each operations:

- **Getting warnings in response:** NITRO allows you to get warnings in an operation by specifying the "warning" query parameter as "yes". For example, to get warnings while connecting to the NetScaler appliance, the URL is as follows:

  http://*<netscaler-ip-address>*/nitro/v1/config/login?warning=yes

  If any, the warnings are displayed in the response payload with the HTTP code "209 X-NITRO-WARNING".

- **Authenticated access for individual NITRO operations:** NITRO allows you to logon to the NetScaler appliance to perform individual operations. You can use this option instead of creating a NITRO session (using the login object) and then using that session to perform all operations,

  To do this, you must specify the username and password in the request header of the NITRO request as follows:

  X-NITRO-USER:*<username>*

  X-NITRO-PASS:*<password>*

  Note: In such cases, make sure that the request header DOES not include the following:

  Cookie:NITRO_AUTH_TOKEN=*<tokenvalue>*

> **Note**
>
> Mandatory parameters are marked in red and placeholder content is marked in *<green>*.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

get (all)

URL: http://<netscaler-ip-address>/nitro/v1/stat/vpnvserver

Query-parameters:

args

http://<netscaler-ip-address>/nitro/v1/stat/vpnvserver?**args=name:<String_value>,detail:<Boolean_value>,fullvalues:<Boolean_value>,ntimes:<Double_value>,logfile:<String_value>,clearstats:<String_v**

Use this query-parameter to get vpnvserver resources based on additional properties.

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver": [ {
    "name":<String_value>,
    "state":<String_value>,
    "requestbytesrate":<Double_value>,
    "primaryipaddress":<String_value>,
    "totalrequests":<Double_value>,
    "type":<String_value>,
    "responsesrate":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totalresponses":<Double_value>,
    "totalrequestbytes":<Double_value>
}]}
```

get

URL: http://<netscaler-ip-address>/nitro/v1/stat/vpnvserver/name_value><String>

HTTP Method: GET

Request Headers:

```
Cookie:NITRO_AUTH_TOKEN=<tokenvalue>
Accept:application/json
```

Response:

HTTP Status Code on Success: 200 OK
HTTP Status Code on Failure: 4xx <string> (for general HTTP errors) or 5xx <string> (for NetScaler-specific errors). The response payload provides details of the error

Response Headers:

```
Content-Type:application/json
```

Response Payload:

```
{ "vpnvserver": [ {
    "name":<String_value>,
    "state":<String_value>,
    "requestbytesrate":<Double_value>,
    "primaryipaddress":<String_value>,
    "totalrequests":<Double_value>,
    "type":<String_value>,
    "responsesrate":<Double_value>,
    "totalresponsebytes":<Double_value>,
    "primaryport":<Integer_value>,
    "requestsrate":<Double_value>,
    "responsebytesrate":<Double_value>,
    "totalresponses":<Double_value>,
    "totalrequestbytes":<Double_value>
}]}
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Messages

Jul 06, 2017

- 0x300 ERRORS
- 0x850 ERRORS
- 0x900 ERRORS
- ACL ERRORS
- ACL6 ERRORS
- APPFLOW ERRORS
- AS ERRORS
- BASE ERRORS
- CACONF ERRORS
- CFE ERRORS
- CLI ERRORS
- CLUSTER ERRORS
- CR ERRORS
- CVPN ERRORS
- DSK CACHE ERRORS
- EV SUBSCRIBER ERRORS
- FWDSESSION ERRORS
- GENTOOL ERRORS
- GSLB ERRORS
- GSLB2 ERRORS
- HTML INJECTION ERRORS
- ICA CLIENT VERSION ERRORS
- INAT ERRORS
- INTERNAL NAMESERVER ERRORS
- INTF ERRORS
- LB ERRORS
- LB2 ERRORS
- LSN ERRORS
- NAT64 ERRORS
- NET ERRORS
- NETBRIDGE ERRORS
- NETPROFILE ERRORS
- NITRO ERRORS
- NNM ERRORS
- PARTITION ERRORS
- PBR ERRORS
- PBR6 ERRORS
- PI ERRORS
- POL ERRORS
- PQ ERRORS
- REWRITE ERRORS
- RISE ERRORS

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

- RNAT ERRORS
- SNMP ERRORS
- SSL ERRORS
- SSL2 ERRORS
- SSLVPN AAA ERRORS
- SST ERRORS
- SVPN ERRORS
- TD ERRORS
- URLTRANS ERRORS
- VLAN ERRORS
- VMAC ERRORS
- VPATH ERRORS
- VXLAN ERRORS
- WARNING ERRORS

© 1999-2017 Citrix Systems, Inc. All rights reserved.

0x300 ERRORS

Jul 07, 2017

0x300 ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_RNAT_INV	769	0x301	Reverse NAT not applicable for default route.
NSERR_INVALID_IF	770	0x302	Invalid interface name/number.
NSERR_MGRLIMIT_REACHED	771	0x303	Maximum manager limit reached.
NSERR_SP_INVALD_TABLE	772	0x304	SP table entries should be in increasing order.
NSERR_RNAT_NATIP_EXISTS	773	0x305	RNAT to the target network with specified NAT IP already exists.
NSERR_RNAT_EXISTS	774	0x306	RNAT to the target network already exists.
NSERR_RNAT_NOT_EXISTS	775	0x307	RNAT to the target network does not exist.
NSERR_RNAT_NATIP_NOT_EXISTS	776	0x308	RNAT to the target network with specified NAT IP doesn't exist.
NSERR_RNAT_INVALID_NATIP	777	0x309	NAT IP is not valid.
NSERR_RNAT_XACLWITHTTL	778	0x30a	Acl with ttl can not be used in RNAT/RNAT6/ForwardingSession/LSN rule.
NSERR_ARP_DISABLED	784	0x310	IP has arp disabled.
NSERR_ARP_SEC_NOT_OWNEDIP	785	0x311	Secondary can not arp for this IP.
NSERR_CPE_RULE_INVAL	786	0x312	Invalid rule.
NSERR_INVAL_FLOWTYPE	787	0x313	Only authorization, audit, VPN session and traffic policies can be bound to aaa user or group.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CPE_RULE_ACTION_INVAL	789	0x315	Request action is valid only for request rule.
NSERR_CPE_DEF_SET_INVAL	790	0x316	Default policy cannot be set.
NSERR_INVAL_FORCECLEANUP	791	0x317	Invalid forcecleanup value.
NSERR_INVAL_AAA_GROUP	792	0x318	Invalid authorizationgroup value.
NSERR_INVAL_PROXY	793	0x319	Invalid allprotocolproxy value.
NSERR_INVAL_HTTPPROXY	800	0x320	Invalid HTTP proxy value.
NSERR_INVAL_FTPPROXY	801	0x321	Invalid FTP proxy value.
NSERR_INVAL_SOCKPROXY	802	0x322	Invalid SOCKS proxy value.
NSERR_INVAL_GOPHERPROXY	803	0x323	Invalid GOPHER proxy value.
NSERR_INVAL_SSLPROXY	804	0x324	Invalid SSL proxy value.
NSERR_INVAL_AAAGRP_MAX	805	0x325	Max 5 groups can be specified in authorizationgroup.
NSERR_INVAL_MAX_PORT_NUM	806	0x326	Maximum 16 ports can be specified in httpport.
NSERR_INVAL_HTTPPORT	807	0x327	Invalid port.
NSERR_INVAL_VPNVSERER_POLTYPE	808	0x328	Only authentication and traffic policies can be bound to a VPN vserver.
NSERR_INVAL_VPNGLOBAL_POLTYPE	809	0x329	Only authentication and traffic policies can be bound to VPN global.
NSERR_CPE_REM_INUSE	810	0x32a	Bound policy cannot be removed.
NSERR_PROXY_CONFLICT	811	0x32b	Proxy server for all protocols already configured.
NSERR_PROXY_INVAL	812	0x32c	Domain names allowed only if proxy type is browser

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PXYEXCPT_INVAL	813	0x32d	Proxy exception allowed only if proxy type is browser
NSERR_CPE_POLTYPE_NO_CSE	814	0x32e	Policy type does not support client security expressions in rule
NSERR_SESSACT_CSE_INCOMPATIBLE	815	0x32f	Session action and rule are incompatible
NSERR_NOMEM_CSE	816	0x330	Not enough memory while adding client security expressions
NSERR_INCOMPAT_FS_RULE	817	0x331	File system expressions supported in authorization policy only
NSERR_INCOMPAT_FS_MIX	818	0x332	Incompatible expressions mixed with file system expressions in rule
NSERR_DR_ENABLE	819	0x333	Dynamic routing can be enabled on only one IP per subnet
NSERR_MAX_DISTANCE	820	0x334	Only null interface routes can have distance equal to 255
NSERR_NULL_ROUTE_DISTANCE	821	0x335	It is not possible to set the administrative distance/cost metric for a null interface route
NSERR_BAD_ACTION_TCP_PROFILE_TYPE	822	0x336	TCP profile cannot be set to this service type
NSERR_BAD_ACTION_HTTP_PROFILE_TYPE	823	0x337	HTTP profile cannot be set to this service type
NSERR_SP_INVALID_THRESHOLD	824	0x338	Invalid base threshold value
NSERR_VIP_ROUTE_EXISTS	825	0x339	VIP exists for this host route
NSERR_ACTION_BOUND	832	0x340	Action bound to policy can not be deleted.
NSERR_MAX_IPLIMIT	833	0x341	Maximum limit for bound IP to this resource record reached, remaining IP's destined for this resource record will be discarded.
NSERR_BAD_PREFIX_LEN	826	0x33a	Invalid rnat/lsn network prefix len it must be 0-128
NSERR_INV_NETADDR	827	0x33b	Invalid IPv6 network address
NSERR_MIN_DISTANCE	828	0x33c	Static route cannot have distance less than 1

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_IPSEC_INVALID_PROFILE_NAME	830	0x33e	Invalid profile name.
NSERR_SPOTTED	831	0x33f	Spotted IP cannot be used here
NSERR_IPSEC_PBR_ONLY_IP_SUPPORTED	834	0x342	Only IP range is supported
NSERR_IPSEC_IP_NOT_CONTIGUOUS	835	0x343	Invalid IP range. The IP range does not cover all the IP's for a given subnet mask
NSERR_INVAL_USER_ACCOUNTING	836	0x344	Invalid userAccounting value. userAccounting value must be a radiusPolicy.
NSERR_RDP_PROFILE_INUSE	837	0x345	Profile is in use.
NSERR_RDP_INVALID_PROFILE_NAME	838	0x346	Invalid profile name.
NSERR_RDP_INVALID_RDP_FILE_NAME	839	0x347	RDP file name should have .rdp extension and some special characters for RDP file name are not supported.
NSERR_RDP_INVALID_SERVERPROFILE_NAME	840	0x348	Invalid server profile name.
NSERR_RDP_SERVERPROFILE_ALREADY_IN_USE	841	0x349	Server profile is already in use.
NSERR_RDP_INVALID_HOST_LEN	848	0x350	Invalid Host length. Atleast 3 characters expected
NSERR_RDP_NEED_IP	849	0x351	RDP IP is needed
NSERR_IPSEC_INV_PROF	850	0x352	For IPSEC tunnels, profile name cannot be 'none'
NSERR_RDP_NEED_PSK	854	0x356	PSK is needed

© 1999-2017 Citrix Systems, Inc. All rights reserved.

0x850 ERRORS

May 20, 2016

0x850 ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_MONITOR_INTERVAL	2128	0x850	Monitor interval must be greater than response timeout
NSERR_MONITOR_DESTIP	2129	0x851	Destination IP must be specified for transparent, tunneled or tos monitors
NSERR_MONITOR_CODES	2130	0x852	Too many response codes, only 16 allowed
NSERR_MONITOR_REF	2131	0x853	Monitor must be unbound before it can be deleted
NSERR_MONITOR_BUILTIN	2132	0x854	Built-in monitors cannot be deleted
NSERR_MONITOR_BOUND	2133	0x855	The monitor is already bound to the service
NSERR_MONITOR_TYPE	2134	0x856	Invalid monitor type
NSERR_MONITOR_LOCAL	2135	0x857	Monitor object cannot be bound to local service
NSERR_TIMEOUT_RANGE	2136	0x858	Timeout value out of range; enter a value between 2 minutes and 1440 minutes
NSERR_INVALIDHASHLEN	2137	0x859	Hashlength should be between 1 and 4096 (inclusive)
NSERR_NOTAUTHORIZED	2138	0x85A	Not authorized to execute this command
NSERR_MONITOR_DEFAULT	2139	0x85B	No operations allowed with the default monitor
NSERR_MONITOR_LDNS_ADD_PERM	2140	0x85C	Cannot add monitor of ldns type
NSERR_MONITOR_LDNS_BIND_PERM	2141	0x85D	Cannot bind ldns monitor

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_BACKUP_LOOP	2144	0x860	The VIP is already a backup in the chain
NSERR_SECURE_UDP	2154	0x86A	Secure mode not supported for this protocol
NSERR_MONITOR_WRONG_TYPE	2155	0x86B	Existing monitor is of different type than given
NSERR_LRTM_PERM	2156	0x86C	Enabling LRTM on this monitor type is not permitted
NSERR_MONITOR_SCRIPTNAME	2157	0x86D	Invalid script name for user monitor
NSERR_MONITOR_DISPATCHERIP	2159	0x86F	Invalid dispatcher IP for user monitor
NSERR_MONITOR_USERPERM	2160	0x870	Operation not permitted for a user monitor
NSERR_MONITOR_NOCODES	2161	0x871	A response code is required for this monitor type
NSERR_INVAL_MON	2162	0x872	Attempt to bind invalid monitor type
NSERR_TYPE_EXISTS	2163	0x873	Only one monitor of this type can be bound to a service
NSERR_MONITOR_UNBIND_DEFAULT	2164	0x874	Default monitor cannot be unbound from a service
NSERR_MONITOR_BIND_DEFAULT	2165	0x875	Default monitor cannot be bound explicitly to a service
NSERR_MONITOR_DISABLE_DEFAULT	2166	0x876	Default monitor cannot be disabled or enabled
NSERR_DRTM_PERM	2167	0x877	Dynamic response timeout is not permitted on this monitor type
NSERR_MON_DTRM_DEVIATION	2168	0x878	Response timeout of the monitor has been changed to interval - 1
NSERR_MONITOR_INVALID_VALUE	2169	0x879	Time parameter must be a multiple of 10
NSERR_MONITOR_NOT_BOUND	2170	0x87A	Monitor not bound to service
NSERR_INVALID_MONITOR	2171	0x87B	Monitor does not exist

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_MONITOR_SCRIPT_ARG_SIZE	2173	0x87D	Error message argument size too long
NSERR_METRICTABLE_NOENT	2174	0x87E	Metric table does not exist
NSERR_METRICTABLE_EXIST	2175	0x87F	Metric table exists
NSERR_DEL_METRICTABLE_PERMANENT	2176	0x880	Permanent metric table cannot be deleted
NSERR_MAX_METRIC_BINDING	2177	0x881	Maximum of 10 SNMP type metric can be bound to the monitor
NSERR_METRIC_NOENT	2178	0x882	Metric does not exist
NSERR_METRIC_EXISTS	2179	0x883	Metric exists
NSERR_OID_EXIST	2180	0x884	SNMP OID exists
NSERR_SNMP_OID_INVAL	2181	0x885	SNMP OID is invalid
NSERR_METRICTABLE_RDONLY	2182	0x886	No operations allowed on read only metric table
NSERR_THRESHOLD_ZERO	2183	0x887	This metric will not be used for CUSTOM LB as its threshold value is zero
NSERR_LDAP_MONITOR_INCOMPLETE	2184	0x888	bindDN or baseDN must be specified before LDAP monitor can be used
NSERR_MYSQL_MONITOR_INCOMPLETE	2185	0x889	userName or database must be specified before MYSQL monitor can be used
NSERR_POP3_MONITOR_INCOMPLETE	2186	0x88A	userName must be specified before POP3 monitor can be used
NSERR_NNTP_MONITOR_INCOMPLETE	2187	0x88B	group must be specified before NNTP monitor can be used
NSERR_FTPEXTENDED_MONITOR_INCOMPLETE	2188	0x88C	fileName must be specified before FTP-EXTENDED monitor can be used
NSERR_SNMP_MONITOR_INCOMPLETE	2189	0x88D	snmpOID or snmpCommunity must be specified before SNMP monitor can be used
			application must be specified before

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CITRIX_WEB_INTERFACE	2191	0x88F	sitePath must be specified before CITRIX-WEB-INTERFACE monitor can be used
NSERR_LDNS_MON_CANT_DISABLE	2192	0x890	Atleast one ldns monitor needs to be enabled
NSERR_RES_RETRY_ON_IP_SVR	2193	0x891	Resolve retry can be set only on domain based servers
NSERR_IP_ON_DBS_SVR	2194	0x892	IP Address cannot be set on a domain based server
NSERR_MONITOR_ALERTRETRIES	2195	0x893	Monitor retries must be greater than SNMP alert retries
NSERR_MONITOR_FAILURERETRIES	2196	0x894	Monitor retries must be greater than monitor failureRetries
NSERR_MONITOR_ALERTFAILURERETRIES	2197	0x895	Monitor failureRetries must be greater than SNMP alertRetries
NSERR_MONITOR_IP	2198	0x896	The server/monitor destination ip address is not appropriate for this monitor
NSERR_MONITOR_SUBNET	2199	0x897	The server/monitor destination ip address should be in a directly connected subnet for this monitor
NSERR_MONITOR_STATIC	2200	0x898	A static/permanent MAC entry is configured for the server/monitor destination ip address
NSERR_TOSID_NOT_SET	2201	0x899	Tos id must be specified for tos enabled vserver/monitor
NSERR_FAILURERETRIES_NOT_SUPPORTED	2202	0x89A	The failure retries is not supported for this monitor type. Resetting the value of failure retries to 0(Disabled).
NSERR_SUCCESSRETRIES_NOT_SUPPORTED	2203	0x89B	The success retries is not supported for this monitor type. Resetting the value of success entries to 1(Disabled).
NSERR_SETVS_INVAL_MYSQLPARAMS	2204	0x89C	This parameter is only applicable to a MYSQL vserver.
NSERR_SETVS_INVAL_PROTOCOL_PARAMS	2205	0x89D	The parameters are only applicable to HTTP or SSL vservers.
NSERR_SETVS_INVAL_PROTOCOLS_PARAMS	2206	0x89E	The parameters are only applicable to HTTP, SSL or RTSP vservers.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			required for Xen Desktop monitor \n
NSERR_CITRIX_WI_EXTENDED_MONITOR_INCOMPLETE	2208	0x8A0	username, password, sitepath and domain must be specified before CITRIX-WI-EXTENDED monitor can be used
NSERR_SETVS_INVAL_MSSQLPARAMS	2209	0x8A1	This parameter is only applicable to a MSSQL vserver.
NSERR_DRTM_MON_BOUNDED_SVC	2210	0x8A2	In a cluster system, deviation can't be set/unset on any already bound monitor.
NSERR_MONITOR_RESP_TO_THRESH	2211	0X8A3	Response timeout threshold is rounded off to nearest multiple of 10 less than the given value
NSERR_PASSIVE_ON_LOAD_ONLY	2212	0x8A4	PASSIVE option can be set only on LOAD monitors.
NSERR_SETVS_INVAL_ORACLEPARAMS	2213	0x8A5	This parameter is only applicable to a ORACLE vserver.
NSERR_RADIUS_ACCOUNTING_MONITOR_INCOMPLETE	2214	0x8A6	Secret key must be specified before RADIUS ACCOUNTING monitor can be used
NSERR_MONITOR_SCRIPT_ARG_CNT	2215	0x8A7	Too many script arguments
NSERR_INVALID_DBSLB_BIND	2216	0x8A8	More than one bind not possible on one service for DBSLB
NSERR_USER_NONETPROF	2217	0x8A9	Netprofile cannot be set on scriptable monitors, so ignoring the configured netprofile for this monitor.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

0x900 ERRORS

Jul 07, 2017

0x900 ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SVCPORTTYPE	2305	0x901	Service exists with the same port and service type
NSERR_VIP_IN_GROUP	2306	0x902	vserver is bound to a group. Set persistence parameters on group to change them for all vservers in group or unbind vserver from group and set on individual vserver.
NSERR_UNSUPPORTED_BKP	2307	0x903	This backup/primary persistence combination is not supported
NSERR_NONHTTPSSL_VIPINGRP	2308	0x904	Cookie persistence cannot be applied - group has non HTTP/SSL type of vserver
NSERR_NONHTTP_VIPINGRP	2309	0x905	Cookie persistence cannot be applied - group has non-HTTP type of vserver
NSERR_USE_PROPER_RM_CMD	2310	0x906	Use remove IP option instead
NSERR_LBMETHOD_NOT_SUPPORTED	2321	0x911	LB method not supported for LLB/PBR
NSERR_PERSISTENCE_NOT_SUPPORTED	2322	0x912	Persistence not supported for LLB/PBR
NSERR_VSERVER_PARAMETERS	2323	0x913	Vserver arguments not valid for LLB/PBR VIP
NSERR_SERVICE_PARAMETERS	2324	0x914	Service parameters are invalid for LLB/PBR VIP
NSERR_LBVIP_DELETE	2325	0x915	Vserver cannot be removed with out removing LB/PBR route
NSERR_CACHEPOLICY_RESPACTION_INVAL	2326	0x916	Integrated caching action cannot be applied on a response
NSERR_CACHELICENSE_FAILED	2327	0x917	The license for Integrated Caching feature was not enabled due to an internal error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_LBVIP_MULTIROUTES	2328	0x918	LB/PBR routes
NSERR_LBMAC_INVAL	2329	0x919	MAC/IPTUNNEL mode can be set only for a VIP with wildcard IP or with service type ANY or for a sessionless VIP
NSERR_CACHESTATS_OBJ_NOTPRESENT	2336	0x920	No object in cache matching the specified attributes
NSERR_CACHEPOLICY_NOTACTIVE	2337	0x921	Integrated caching policy is not active
NSERR_CACHE_BUILTINS_NOT_SOURCED	2338	0x922	Failed sourcing cache builtins, Disabling IC
NSERR_CACHEPOLICY_PRIORITY_INVAL	2339	0x923	Invalid priority
NSERR_ROUTE6_DEFAULT_ONLY	2340	0x924	Only default route configuration supported
NSERR_ROUTE6_DEFAULT_EXISTS	2341	0x925	Configured route already exists
NSERR_ROUTE6_INVALID_GATEWAY	2342	0x926	Invalid gateway
NSERR_ROUTE6_MAX	2343	0x927	Already maximum number of routes configured
NSERR_ROUTE6_NOT_EXIST	2344	0x928	Route does not exist
NSERR_IPV6_INVALID_ADDR	2345	0x929	Incorrect IPv6 address type
NSERR_LBVIP_IPV6TOV4	2352	0x930	Changing IP from IPv6 to IPv4 not permitted
NSERR_IPV6_NSIPTOVIP	2353	0x931	Configuring NSIP as VIP not permitted
NSERR_IPV6_VIPTONSIP	2354	0x932	Configuring VIP as NSIP not permitted
NSERR_IPV6_SCOPE	2355	0x933	Incorrect IPv6 address scope. (Default: global)
NSERR_LBVIP_IPV4TOV6	2356	0x934	Changing IP from IPv4 to IPv6 not permitted
NSERR_IPV6_LINKLOCALTOVIP	2357	0x935	Configuring Link-Local address as VIP not permitted

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			NSIP (ignoring mapped ip)
NSERR_SECUREIPPORTADDRINUSE	2359	0x937	Internal secure service exists with the same port and service type. Please use this for secure access
NSERR_IPV6_INVALID_PREFIX	2360	0x938	Link-local prefix length is not equal to 64
NSERR_ND6_LINKLOCAL_VLAN	2361	0x939	Vlan is necessary with IPv6 Link-local address
NSERR_INVALID_IPV6_PREFIXLFT	2362	0x93A	Invalid IPv6 prefix life time. prefix valid life time must be greater than or equal to preferred life time
NSERR_IPV6_ND_TIMEOUT	2363	0x93B	Can not SET/UNSET ndtimeouts, when ra_learning is enabled.
NSERR_ND6_VLAN_INTF	2368	0x940	Interface/channel not a member of given vlan
NSERR_PERSIST_TIMEOUT_TO_DEFAULT	2369	0x941	Unsetting Persistency, changing timeout to default
NSERR_LB_GROUP_NOT_EXIST	2370	0x942	LB group does not exist
NSERR_LB_VSERVER_ALREADY_BOUND	2371	0x943	Vserver is already bound to a LB group
NSERR_ROUTE6_INVALID_LINKLOCAL_GATEWAY_VLAN	2372	0x944	Vlan ID must be specified for linklocal gateway
NSERR_ROUTE6_NO_GWORIF	2373	0x945	Either Gateway or Interface (vlan) should be specified for route
NSERR_ROUTE6_VLAN_MONITOR	2374	0x946	Monitor should not be given for vlan route
NSERR_LBMETHOD_LEASTREQ_NOT_SUPPORTED	2375	0x947	LeastRequest LB method not supported for this service type
NSERR_IPV6_MULTIPLE_LLIP	2376	0x948	Multiple link-local IPs to a Vlan is not allowed
NSERR_IPV6_TOOMANY_PREFIXES	2377	0x949	Too many prefixes to a Vlan, maxmimum 5 prefixes are allowed
NSERR_RAPREFIX_BOUNDTOVLAN	2378	0x94A	Router Advertisement prefix is bound to vlan
NSERR_LB_VSERVER_NOT_BOUND	2379	0x94B	Given vserver is not bound to this LB group

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_VSERVER_BOUND_TO_NG	2380	0x94C	Cannot delete vserver that is bound to a nodegroup
NSERR_VPATH_DESTIP_NOTDIRECT	2381	0x94D	Destination IPaddress should be directly connected
NSERR_LRONLACPONLY	2382	0x94E	Link Redundancy can be set only on LACP Channels
NSERR_CACHE_POLICY_CONTENTGRPMISMATCH	2383	0x94F	Type of Content Group and Policy does not Match.
NSERR_NONHTTPNOSQL_VS	2384	0x950	Specified policy can be bound only to HTTP/SSL/MYSQL/MSSQL vserver.
NSERR_SMPP_CUSTOM_SVRID_LEN_NOT_SUPPORTED	2385	0x951	CustomServerID length for SMPP protocol must be 2
NSERR_SMPP_CUSTOM_SVRID_NOT_EXIST	2386	0x952	CustomServerID is mandatory for SMPP protocol
NSERR_SMPP_CUSTOM_SVRID_EXIST	2387	0x953	CustomServerID already exists, it should be unique for SMPP protocol
NSERR_SIP_DUPLICATE_SRC_PORT	2388	0x954	RNAT source port and RNAT secure source port must be different
NSERR_SIP_DUPLICATE_DST_PORT	2389	0x955	RNAT destination port and RNAT secure destination port must be different
NSERR_IP_ADDR_ICMP_DIS_NOT_ALLOWED	2390	0x956	Disabling ICMP for an IP(v6) address is not allowed
NSERR_IP_ADDR_ND6_DIS_NOT_ALLOWED	2391	0x957	disabling neighbor discovery for an ipv6 address is not allowed
NSERR_IP_ADDR_STATE_DIS_NOT_ALLOWED	2392	0x958	state disable is not allowed for an IP(v6) address
NSERR_ND6_LOOP_LL_NOT_ALLOW	2393	0x959	Loopback/self or link-local IP addresses not allowed

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ACL ERRORS

Jul 07, 2017

ACL ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ACL_NOT_EXISTS	864	0x360	ACL rule does not exist
NSERR_ACL_EXISTS	865	0x361	IP address has existing ACL rule
NSERR_ACLPIP_WOSRCDST	866	0x362	Peer IP can't be given without src/dst flag
NSERR_ACL_SAMEIP_PIP	867	0x363	IP address and peer IP can't be same
NSERR_ACL_INVAL_PEERIP	868	0x364	Invalid peer IP
NSERR_ACL_IPPIP_EXISTS	869	0x365	ACL with identical parameter specification already exists
NSERR_XACLDELERROR	870	0x366	ACL has already been removed
NSERR_XACLADDERROR	871	0x367	Port can be specified only if protocol is TCP (6) or UDP (17)
NSERR_XACL_PRIORITY_EXISTS	872	0x368	ACL with this priority already exists
NSERR_XACL_ICMP_REQD	873	0x369	ICMP type / code can be specified only if protocol is ICMP(1)
NSERR_NOLOOPBACK	874	0x36a	ACL cannot be configured on the loopback interface
NSERR_INVICMPTYPE	875	0x36b	Invalid ICMP type
NSERR_INVICMPCODE	876	0x36c	Invalid ICMP code
NSERR_XACLRNATDEL	877	0x36d	ACL is bounded to RNAT/LSN, cannot be removed

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_XACLFWDSESSIONDEL	879	0x36f	ACL is bounded to FORWARDSESSION, cannot be removed

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ACL6 ERRORS

Jul 07, 2017

ACL6 ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ACL6_IPPIP_EXISTS	896	0x380	ACL6 with identical parameter specification already exists
NSERR_ACL6_DELERROR	897	0x381	ACL6 has already been removed
NSERR_ACL6_ADDERROR	898	0x382	Port can be specified only if protocol is TCP (6) or UDP (17)
NSERR_ACL6_PRIORITY_EXISTS	899	0x383	ACL6 with this priority already exists
NSERR_ACL6_ICMP_REQD	900	0x384	ICMPv6 type/code can be specified only if protocol is ICMPv6 (58)
NSERR_ACL6_UNSPECADDR	901	0x385	unspecified address (::) can not be configured in ACL6
NSERR_ACL6_MODCFGINFO	902	0x386	ACL6 modified, apply ACLs6 to activate change
NSERR_ACL6_PREFIXLEN	903	0x387	Prefix length should not be configured in ACL6. (Use range instead)
NSERR_ACL6_RNAT_NOT_ALLOWED	904	0x388	Action must be ALLOW to bind ACL6 to RNAT
NSERR_ACL6RNAT_DELERROR	905	0x389	This ACL6 is associated with rnat6/nat64/forwardingsession/lsn config, Remove rnat6/nat64/forwardingsession/lsn first
NSERR_RNAT6_INVALID_REDPORT	906	0x38a	Invalid option for SET/UNSET, redirect port can be specified only for acl6 based RNAT
NSERR_ACL6_RNAT_EXISTS	907	0x38b	ACL6 based rnat is already exist for this acl6(Duplicate config)
NSERR_NET_RNAT6_EXISTS	908	0x38c	network based rnat is already exist for this network address(Duplicate config)
NSERR_RNAT_VIP_PORT_ALLOC_FAIL	909	0x38d	ERROR : Port allocation for VIP v6 address Failed

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ACL6_RNAT_NAT64_NOT_ALLOWED	910	0x38e	ERROR: Acl6 already bound to nat64 config
NSERR_ACL_SRCMAC_MASK_LENGTH	911	0x38f	SrcMacMask length should be 12
NSERR_ACL_INVALID_NAME	1444	0x5a4	ERROR: ACL name length should be >=1 and <=128
NSERR_ACL_SRCMAC_MASK_PAIR	1455	0x5af	SrcMacMask should Contain consecutive 1s or 0s per octet and pair of 0s should not be present after pair of 1s

© 1999-2017 Citrix Systems, Inc. All rights reserved.

APPFLOW ERRORS

Jul 07, 2017

APPFLOW ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_APPFLOW_INVALIDPORT	3936	0xf60	Invalid collector port
NSERR_NO_SUCH_COLLECTOR	3938	0xf62	No matching collector
NSERR_APPFLOW_NCORE_ONLY	3939	0xf63	The AppFlow feature is available only on Citrix NetScaler nCore.
NSERR_APPFLOW_NONZERO_ACTION_REFCOUNT	3940	0xf64	The specified AppFlow collector is being used in an AppFlow action.
NSERR_APPFLOW_COLLECTOR_NAME_INUSE	3941	0xf65	AppFlow collector name already in use.
NSERR_APPFLOW_ACT_INVAL	3942	0xf66	No such AppFlow action exists.
NSERR_APPFLOW_NETPROF_REQD	3472	0xd90	NetProfile required for AppFlow collector within partition.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AS ERRORS

May 20, 2016

AS ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AS_NOSTARTURL	3120	0xc30	No such StartURL check
NSERR_AS_EXIST_STARTURL	3121	0xc31	The StartURL check is already in use
NSERR_AS_NODENYURL	3122	0xc32	No such DenyURL check
NSERR_AS_EXIST_DENYURL	3123	0xc33	The DenyURL check is already in use
NSERR_AS_NOCOOKIECONSISTENCY	3124	0xc34	No such CookieConsistency check
NSERR_AS_EXIST_COOKIECONSISTENCY	3125	0xc35	The CookieConsistency check is already in use
NSERR_AS_NOFIELDCONSISTENCY	3126	0xc36	No such FieldConsistency check
NSERR_AS_EXIST_FIELDCONSISTENCY	3127	0xc37	The FieldConsistency check is already in use
NSERR_AS_NOXSS	3128	0xc38	No such CrossSiteScripting check
NSERR_AS_EXIST_XSS	3129	0xc39	The CrossSiteScripting check is already in use
NSERR_AS_NOSQL	3130	0xc3a	No such SQLInjection check
NSERR_AS_EXIST_SQL	3131	0xc3b	The SQLInjection check is already in use
NSERR_AS_NOFIELDFORMAT	3132	0xc3c	No such FieldFormat check
NSERR_AS_EXIST_FIELDFORMAT	3133	0xc3d	The FieldFormat check is already in use

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AS_NOOBJECTEXPRESSION	3134	0xc3e	No such SafeObject check
NSERR_AS_EXIST_OBJECTEXPRESSION	3135	0xc3f	The SafeObject check is already in use
NSERR_AS_NOFIELDTYPE	3136	0xc40	No such FieldType. See /var/log/ns.log for more details.
NSERR_AS_NOT_SUPPORTED_VS	3137	0xc41	Specified policy can be bound only to HTTP/SSL vserver
NSERR_AS_BAD_ACTION_STARTURL	3138	0xc42	Invalid StartURL Action
NSERR_AS_BAD_ACTION_DENYURL	3139	0xc43	Invalid DenyURL Action
NSERR_AS_BAD_ACTION_COOKIECONSISTENCY	3140	0xc44	Invalid CookieConsistency Action
NSERR_AS_BAD_ACTION_FIELDCONSISTENCY	3141	0xc45	Invalid FieldConsistency Action
NSERR_AS_BAD_ACTION_XSS	3142	0xc46	Invalid CrossSiteScripting Action
NSERR_AS_BAD_ACTION_SQL	3143	0xc47	Invalid SQLInjection Action
NSERR_AS_BAD_ACTION_FIELDFORMAT	3144	0xc48	Invalid FieldFormat Action
NSERR_AS_BAD_ACTION_OBJECTEXPRESSION	3145	0xc49	Invalid SafeObject Action
NSERR_AS_BAD_ACTION_BUFFEROVERFLOW	3146	0xc4a	Invalid BufferOverflow Action
NSERR_AS_BAD_ACTION_CCARD	3147	0xc4b	Invalid CreditCard Action
NSERR_AS_FIELDFORMAT_MIN_GT_MAX	3148	0xc4c	Minimum FieldFormat length cannot exceed maximum FieldFormat length
NSERR_AS_FIELDTYPE_BAD_NAME_LEN	3149	0xc4d	Invalid FieldType name length
NSERR_AS_BAD_COMMENT_LEN	3150	0xc4e	Invalid comment length
NSERR_AS_FIELDTYPE_BAD_REGEX_LEN	3151	0xc4f	Invalid regex length

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code (Decimal)	Error Code (Hex)	Error Message
NSERR_AS_FIELDTYPE_BUILTIN	3153	0xc51	Built-in field types cannot be modified or deleted
NSERR_AS_BAD_ACTION_CONTENT_TYPE	3154	0xc52	Invalid ContentType Action
NSERR_AS_BAD_DEFAULT_CHARSET	3155	0xc53	Invalid default character set
NSERR_AS_BAD_COOKIECONSISTENCY_NAME	3156	0xc54	Invalid Cookie name
NSERR_AS_BAD_ERRORURL	3157	0xc55	Invalid ErrorURL
NSERR_AS_BAD_ENCODING_URL	3158	0xc56	Invalid encoding for URL
NSERR_AS_BAD_ENCODING_FIELDNAME	3159	0xc57	Invalid encoding for field name
NSERR_AS_BAD_ENCODING_COOKIENAME	3160	0xc58	Invalid encoding for cookie name
NSERR_AS_BAD_ENCODING_OBJECTEXPRESSION_NAME	3161	0xc59	Invalid encoding for SafeObject name
NSERR_AS_BAD_ENCODING_EXPRESSION	3162	0xc5a	Invalid encoding for expression
NSERR_AS_BAD_ENCODING_REGEX	3163	0xc5b	Invalid encoding for regex
NSERR_AS_MODIFIED_URL	3164	0xc5c	Replaced character that is not printable ASCII with escaped equivalent in URL
NSERR_AS_MODIFIED_OBJECTEXPRESSION	3165	0xc5d	Replaced character that is not printable ASCII with escaped equivalent in expression
NSERR_AS_CFFIELD_BAD_FIELDNAME_LEN	3166	0xc5e	Invalid confidential form field name length
NSERR_AS_CFFIELD_BAD_URL_LEN	3167	0xc5f	Invalid confidential form field URL length
NSERR_AS_NOCFFIELD	3168	0xc60	No such confidential form field
NSERR_AS_BAD_CFF_NAME	3169	0xc61	Name may not contain leading/trailing spaces.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			leading/trailing spaces.
NSERR_AS_CFF_DUP	3171	0xc63	The confidential field is already in use.
NSERR_AS_BAD_ACTION_XDOS	3172	0xc64	Invalid XML Dos Action
NSERR_AS_EXIST_XML_DOS_URL	3173	0xc65	The XML DoS URL check is already in use.
NSERR_XML_URL_NOT_SUPPORTED	3174	0xc66	Only .* is supported for XML checks.
NSERR_AS_NO_XDOS_URL	3175	0xc67	No such URL exist for XDOS check.
NSERR_AS_INVALID_XML_DOS_CONF	3176	0xc68	Invalid configuration: xmlMaxFileSize can not be less than xmlMinFileSize when both checks are enabled.
NSERR_AS_BAD_ACTION_XML_SQLINJECTION	3177	0xc69	Invalid XML - SQLInjection Action.
NSERR_AS_BAD_ACTION_XML_XSS	3178	0xc6a	Invalid XML - CrossSiteScripting Action.
NSERR_AS_BAD_ACTION_XML_WELLFORMEDNESS	3179	0xc6b	Invalid XML - Format Action.
NSERR_AS_BAD_ACTION_PROFILE_TYPE	3180	0xc6c	Invalid Appsecure Profile Type.
NSERR_AS_MODIFIED_FIELDNAME	3181	0xc6d	Replaced character that is not printable ASCII with escaped equivalent in form field name
NSERR_AS_BAD_ACTION_WSI	3182	0xc6e	Invalid XML WS-I Action.
NSERR_AS_EXIST_XML_WSI_URL	3183	0xc6f	The XML WS-I URL check is already in use.
NSERR_AS_NO_WSI_URL	3184	0xc70	No such URL exist for WS-I check.
NSERR_AS_INVALID_XML_WSI_CONF	3185	0xc71	Invalid WS-I rule id in the list.
NSERR_AS_OBJECTNAME_TOO_BIG	3186	0xc72	Object too big.
			Imported file does not exist

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			use]. See /var/log/ns.log for more details.
NSERR_AS_SERVER_NAME_TOO_BIG	3188	0xc74	Server name too big.
NSERR_AS_OBJECT_NOT_READABLE	3189	0xc75	Object not readable [Please make sure it exists].
NSERR_AS_NOPROFILE	3190	0xc76	No such profile.
NSERR_AS_BAD_ACTION_XML_ATTACHMENT	3191	0xc77	Invalid XML - Attachment Action
NSERR_AS_BAD_ACTION_MSGVAL	3192	0xc78	Invalid XML Message Validation Action.
NSERR_AS_EXIST_XML_MSGVAL_URL	3193	0xc79	The XML MSGVAL URL check is already in use.
NSERR_AS_NO_MSGVAL_URL	3194	0xc7a	No such URL exist for MSGVAL check.
NSERR_AS_INVALID_XML_MSGVAL_CONF	3195	0xc7b	Invalid MsgVal configuration.
NSERR_AS_BIND_XML_MSGVAL_CONF	3196	0xc7c	Error in message validation binding. For More details see log messages.
NSERR_AS_IMPORT_FAILED	3197	0xc7d	Importing the resource failed
NSERR_AS_OBJECT_EXIST	3198	0xc7e	Object already exists
NSERR_AS_INVALID_OPTION	3199	0xc7f	Invalid option
NSERR_AS_REMOVE_FAILED	3200	0xc80	Problem in removing resource
NSERR_AS_NO_RESOURCE	3201	0xc81	No such resource. Object you are trying to update or remove does not exist.
NSERR_AS_SHOW_FAILED	3202	0xc82	Problem in showing object
NSERR_AS_DEPENDENCY_FAILED	3203	0xc83	Problem in downloading dependencies
			Problem compiling object. For

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INVALID_OBJECT_NAME	3205	0xc85	Invalid name; names must begin with an alphanumeric character or underscore and must contain only alphanumerics, '_', '#', '.', ':', '@', '=' or '-'
NSERR_AS_IMPORT_DOWNLOAD_FAILED	3206	0xc86	Problem in importing the object. Please check the DNS NameServer/Route settings and try again. For more details see /var/log/ns.log file
NSERR_AS_RESOURCE_INUSE	3207	0xc87	Can not remove resource. Resource is in use
NSERR_AS_BAD_MAPPING_DATA	3208	0xc88	Mapping Data Corrupted
NSERR_AS_PROFILE_CHANGE_HTML	3209	0xc89	HTML checks will not be applicable when profile type is XML
NSERR_AS_PROFILE_CHANGE_XML	3210	0xc8a	XML checks will not be applicable when profile type is HTML
NSERR_AS_IMPORT_ALREADY_INPROCESS	3211	0xc8b	Import failed. Another resource with the same name being processed
NSERR_AS_INVALID_XML_ERROR_OBJECT	3212	0xc8c	Invalid XML error object
NSERR_AS_XML_WELLFORMEDNESS_DISABLED	3213	0xc8d	XML Security checks can not be performed, once an XML message is found not wellformed.
NSERR_AS_INDIVIDUAL_IMPORT_LIMIT_EXCEEDED	3214	0xc8e	Import failed - importing file size greater than configured size limit
NSERR_AS_TOTAL_IMPORT_LIMIT_EXCEEDED	3215	0xc8f	Import failed - exceeding the configured total size limit on the imported objects
NSERR_AS_LEARNING_BUSY	3216	0xc90	Please wait for the learning database to finish updating
NSERR_AS_DEPRECATED_XML_WSI_RULE_R4003	3217	0xc91	WS-I Rule R4003 has been deprecated, it will be removed from the WS-I binding list.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			is already in use.
NSERR_AS_NO_XML_ATTACHMENT_URL	3219	0xc93	No such URL exist for XML Attachment check.
NSERR_AS_BAD_ACTION_SOAP_FAULT	3220	0xc94	Invalid SOAP Fault Filtering Action.
NSERR_AS_XML_DOS_VALUE_OUT_OF_RANGE	3221	0xc95	XML DoS check value is not within the allowed range.
NSERR_AS_XML_MSGVAL_CLI_ERROR	3222	0xc96	-XMLValidateResponse requires one of [-XMLValidateSOAPEnvelope, -XMLWSDL, -XMLRequestSchema, -XMLResponseSchema]
NSERR_AS_INVALID_CUSTOM_SETTINGS_OBJECT	3223	0xc97	Import failed. Please check syntax of signature object
NSERR_AS_SESSION_TIMEOUT_LIFETIME_CONFLICT	3224	0xc98	The session lifetime cannot be less than the session timeout
NSERR_AS_CLASSIC_POLICY_ALREADY_BOUND	3225	0xc99	Failed to bind advanced policy because a classic one is already bound globally or to a vserver. Binding classic and advanced policies at the same time is not allowed
NSERR_AS_ADVANCED_POLICY_ALREADY_BOUND	3226	0xc9a	Failed to bind classic policy because an advanced one is already bound globally or to a vserver or to a user defined policylabel. Binding classic and advanced policies at the same time is not allowed
NSERR_AS_CUSTOM_SETTINGS_NO_INJECTION_TYPE	3227	0xc9b	Failed to set signature object. One of the
NSERR_AS_NOCSRF_TAG	3228	0xc9c	No such CrossSiteRequestForgery check
NSERR_AS_EXIST_CSRF_TAG	3229	0xc9d	The CrossSiteRequestForgery check is already in use
NSERR_AS_BAD_ACTION_CSRF_TAG	3230	0xc9e	Invalid CSRF Tag Action
NSERR_AS_TURNING_TAGGING_OFF_CSRF_TAG_ON	3231	0xc9f	Must set CSRF tagging check to 'none' before disabling form tagging.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AS_TURNING_CSRF_TAG_ON_TAGGING_OFF	3232	0xca0	Form tagging must be enabled before turning on CSRF tag checks.
NSERR_AS_IMPORT_LIMIT_LESS_THAN_IMPORT_DIR_SIZE	3233	0xca1	Import limit should be greater than the current total imported objects size.
NSERR_AS_UNSUPPORTED_IMPORT_PROTOCOL	3234	0xca2	Unsupported import source protocol. Supported protocols are http, https and ftp
NSERR_AS_UNSUPPORTED_EXPORT_PROTOCOL	3235	0xca3	Unsupported export target protocol. Supported protocols are http and https
NSERR_AS_EXPORT_FAILED	3236	0xca4	Export failed. Check /var/log/ns.log for details.
NSERR_AS_BAD_LOCAL_FILE	3237	0xca5	Local file cannot use .. to backtrack up a directory.
NSERR_AS_INVALID_REGEX_CUSTOM_OBJECT	3238	0xca6	Invalid regular expression in custom object. Reverting to default settings.
NSERR_AS_BUILTIN_INVALID_OP	3239	0xca7	Invalid operation for built-in profile
NSERR_AS_INVALID_ADV_POLICY_STATE	3240	0xca8	Cannot bind advanced policy with state set to DISABLED
NSERR_AS_BUILTIN_NAMES_IN_CONF	3241	0xca9	Entities with names reserved for built-in profiles have been discarded
NSERR_AS_DEFAULT_PROFILE_IS_BYPASS	3242	0xcaa	The default profile is set to bypass traffic. It can be set using the 'set appfw settings' command.
NSERR_AS_NO_LEARN_DENYURL	3243	0xcab	Learn is not supported as a denyURLAction.
NSERR_AS_INVALID_XML_FILE_SIZE	3244	0xcac	Invalid configuration: xmlMinFileSize / xmlMaxFileSize can not be less than 4 or greater than 1000000000.
NSERR_AS_NO_LEARN_BUFFEROVERFLOW	3246	0xcae	Learn is not supported as a bufferOvewflowAction.
NSERR_AS_NO_LEARN_XML_FORMAT	3247	0xcaf	Learn is not supported as an

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AS_NO_LEARN_XML_SQLINJECTION	3251	0xcb3	Learn is not supported as an XMLSQLInjectionAction.
NSERR_AS_NO_LEARN_XML_XSS	3252	0xcb4	Learn is not supported as an XMLXSSAction.
NSERR_AS_NO_LEARN_XML_MSGVAL	3253	0xcb5	Learn is not supported as an XMLValidationAction.
NSERR_AS_NO_LEARN_XML_SOAP_FAULT	3254	0xcb6	Learn is not supported as an XMLSOAPFaultAction.
NSERR_AS_IMPORT_INTERNAL_ERROR	3255	0xcb7	Internal error while importing resource.
NSERR_AS_NETSVC_CONN_FAILED	3256	0xcb8	Critical internal error. please retry after some time.
NSERR_AS_VALIDATION_FAILED	3257	0xcb9	Problem validating Imported object against specification. For more details see /var/log/ns.log file.
NSERR_AS_CKI_TRANSFORM_DISABLED	3258	0xcba	This setting will not take effect until cookieTransforms is turned ON.
NSERR_AS_NOXMLXSS	3259	0xcbb	No such XMLXSS check
NSERR_AS_EXIST_XMLXSS	3260	0xcbc	The XMLXSS check is already in use
NSERR_AS_NOXMLSQL	3261	0xcbd	No such XMLSQLInjection check
NSERR_AS_EXIST_XMLSQL	3262	0xcbe	The XMLSQLInjection check is already in use
NSERR_AS_NO_ENDTAG	3263	0xcbf	Missing end tag in the imported signature file.
NSERR_AS_SIG_INVALID_RULEID	3264	0xcc0	Invalid rule id in imported signature.
NSERR_AS_SIG_INVALID_RULE_VERSION	3265	0xcc1	Invalid version in imported signature.
NSERR_AS_SIG_PARSE_ERROR	3266	0xcc2	Error parsing imported signatures.
NSERR_AS_SIG_INVALID_ATTRIBUTE	3267	0xcc3	Imported signatures have invalid attributes

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AS_SIG_MULTIPLE_LOCATIONS	3268	0xcc4	multiple locations
NSERR_AS_SIG_INVALID_STATE	3269	0xcc5	Error parsing imported signatures. Invalid parse state.
NSERR_AS_CKI_ENCRYPT_METHOD_IS_NONE	3270	0xcc6	For security purposes, please set the NS encryptions parameter using the \"set ns encryptionParams\" command.
NSERR_AS_UPDATE_FAILED	3271	0xcc7	Updating the resource failed
NSERR_AS_OBJECT_NOT_USED	3272	0xcc8	The specified object is not used
NSERR_AS_SIG_MULTIPLE_URLS	3273	0xcc9	Imported signature pattern location has multiple URLs
NSERR_AS_SIG_MULTIPLE_FIELD_NAMES	3274	0xcca	Imported signature pattern location has multiple field names
NSERR_AS_SIG_NO_FASTMATCH	3275	0xccb	A signature rule has no fastmatch pattern. See log for details.
NSERR_AS_PE_POLICY_NOTSUPPORTED	3276	0xccc	Not supported for classic policy.
NSERR_AS_AUDITLOG_POLICY_NOTSUPPORTED	3277	0xccd	Not supported for auditlog policy.
NSERR_AS_SIG_MULTIPLE_CREDIT_CARD_RULES	3278	0xcce	Imported signatures has multiple credit card patterns in Rule.
NSERR_AS_TURNING_TAGGING_OFF_SESSIONLESS_FFC_ON	3279	0xccf	Must disable sessionless field consistency checks before disabling form tagging.
NSERR_AS_TURNING_SESSIONLESS_FFC_ON_TAGGING_OFF	3280	0xcd0	Form tagging must be enabled before enabling sessionless field consistency checks.
NSERR_AS_SIG_INVALID_CREDIT_CARD_PATTERN	3281	0xcd1	Imported Signatures has invalid Credit Card pattern.
NSERR_AS_SIG_INVALID_SAFE_OBJECT_PATTERN	3282	0xcd2	Imported Signatures has invalid Safe Object pattern.
NSERR_AS_IMPORT_INVALID_TAR_ARCHIVE	3283	0xcd3	Imported archive is invalid. Please provide valid tar.gz archive.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code (Decimal)	Error Code (Hex)	Error Message
			SQL injection attributes.
NSERR_AS_SPLSTRING_INVALID_ATTRIBUTE	3285	0xcd5	Imported signatures has invalid special string attributes.
NSERR_AS_SQLINJECTION_KEYWORD_LITERAL_EXCEED_MAXLEN	3286	0xcd6	Maximum allowed length for type \"LITERAL\" for keyword is 255.
NSERR_AS_SPLSTRING_LITERAL_EXCEED_MAXLEN	3287	0xcd7	Maximum allowed length for type \"LITERAL\" for special string is 255.
NSERR_AS_XSLT_TRANFORM_LATEST	3288	0xcd8	Problem converting imported signature to the latest version. For more details see /var/log/ns.log file.
NSERR_AS_XSLT_TRANFORM_USER	3289	0xcd9	Problem converting imported signature with the given xslt file. For more details see /var/log/ns.log file.
NSERR_AS_SIG_RESP_BODY_EXPR_ERROR	3290	0xcda	Match type EXPRESSION is not supported in response rules.
NSERR_AS_NOTRUSTEDLEARNINGCLIENT	3291	0xcdb	No Such Trusted Learning Client.
NSERR_AS_EXIST_TRUSTEDLEARNINGCLIENT	3292	0xcdc	The Trusted Learning Client is already in use.
NSERR_AS_POLICY_ALLOWED_BINDPOINT_NONE	3293	0xcdd	Only bindpoint none is allowed for this policy.
NSERR_AS_XSS_DENIED_PATTERN_INVALID_ATTRIBUTE	3294	0xcde	Imported signatures has invalid denied pattern attribute.
NSERR_AS_XSS_DENIED_PATTERN_EXCEED_MAXLEN	3295	0xcdf	Maximum allowed length for type \"LITERAL\" for denied pattern is 255.
NSERR_AS_XSS_ALLOWED_PATTERN_INVALID_ATTRIBUTE	3296	0xce0	Imported signatures has invalid allowed pattern attribute.
NSERR_AS_XSS_ALLOWED_PATTERN_EXCEED_MAXLEN	3297	0xce1	Maximum allowed length for type \"LITERAL\" for allowed pattern is 255.
NSERR_AS_SIG_MULTIPLE_SUB_LOCATIONS	3298	0xce2	Imported signature pattern location has multiple sub locations
			Internal error when configuring

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AS_SIG_URL_TAG_ERROR	3300	0xce4	Signature file has
NSERR_AS_SIG_HEADER_NAME_TAG_ERROR	3301	0xce5	Signature file has
NSERR_AS_SIG_COOKIE_NAME_TAG_ERROR	3302	0xce6	Signature file has
NSERR_AS_SIG_FIELD_NAME_TAG_ERROR	3303	0xce7	Signature file has
NSERR_AS_SIG_PATTERN_ADD_ERROR	3304	0xce8	Problem constructing signature pattern from file
NSERR_AS_SIG_NON_LITERAL_FASTMATCH	3305	0xce9	Signature file has \"fastmatch\" on a non-literal match. See /var/log/ns.log for more details
NSERR_AS_SIG_NEGATED_LITERAL_FASTMATCH	3306	0xcea	Signature file has \"fastmatch\" on a negated literal match.
NSERR_AS_SIG_MEM_ALLOC_FAILED	3307	0xceb	Memory allocation failed while loading signatures.
NSERR_AS_SIG_TOO_MANY_PATTERNS	3308	0xcec	Signature file has a rule with too many patterns.
NSERR_AS_SIG_LITERAL_HEX_PARSE_ERROR	3309	0xced	Signature file has a pattern with hex byte syntax error.
NSERR_AS_SIG_PCRE_COMPILE_ERROR	3310	0xcee	Signature file has an invalid PCRE.
NSERR_INVALID_SIGNATURE_NAME	3311	0xcef	Invalid signature name; it must begin with an alphanumeric character or underscore and must contain only alphanumerics, '_', '.', or '-'
NSERR_AS_SIGNATURE_MERGE	3312	0xcf0	Problem merging imported signature rules with the existing signature rules. For more details see /var/log/ns.log file.
NSERR_AS_RECORDER_MEM_ALLOC_ERROR	3313	0xcf1	Cannot allocate memory for requested recorder size.
NSERR_AS_SIG_INVALID_FILE_VERSION	3314	0xcf2	Signature file has an invalid SignaturesFile 'version' value.
NSERR_AS_AUTO_SIGNATURE_UPDATED_FAILED	3315	0xcf3	Updating of signatures failed. For more details see /var/log/ns.log

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AS_NO_CONTENT_TYPE	3316	0xcf4	No Such content-type.
NSERR_AS_POLICY_BINDING_NOTALLOWED	3317	0xcf5	Appfw policy can only be bound to spotted vservers that are associated to nodegroup which is bound to one/zero node.
NSERR_AS_POLICY_SET_NOTALLOWED	3318	0xcf6	Set disallowed as advance profile can only be associated to policy that is either not bound or bound to spotted vservers.
NSERR_AS_SIGNATURE_URL_NOT_ACCESSIBLE	3319	0xcf7	Signature URL set in 'set appfw settings' is not accessible. Please check the DNS NameServer/Route settings and try again.
NSERR_AS_NODEGROUP_UNBIND_VSERVER_NOTALLOWED	3320	0xcf8	Unbinding vserver from nodegroup disallowed as vserver has appfw policy bound to it.
NSERR_AS_NODEGROUP_BIND_NODE_NOTALLOWED	3321	0xcf9	Nodegroup could be associated with only one node if it has a vserver that has appfw policy bound to it.
NSERR_AS_BUILT_IN_OBJECT_ERROR	3322	0xcfa	Cannot add or remove built in objects.
NSERR_AS_SIGNATURE_VALIDATE_FAILED	3323	0xcfb	Imported signature is failed to validate against sha1 file. For more details see /var/log/ns.log file.
NSERR_AS_INVALID_PARAMETER	3325	0xcfd	Sha1 parameter is allowed to use only with Import Appfw Signatures command
NSERR_AS_LEARNING_DB_ERROR	3326	0xcfe	Learning database is either corrupt or does not exist.
NSERR_AS_VALIDATION_SIGNATURE_MAPPING_FAILED	3327	0xcff	SignatureMapping.xml is invalid. Failed to validate SignatureMapping.xml against its schema file.
NSERR_AS_EXIST_CONTENT_TYPE	3332	0xd04	Content-type is already in use.
			Signature file has a rule with a

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			the required maxMatchLength attribute
NSERR_AS_NOT_ALLOWED_CONTENT_TYPE	3334	0xd06	Content-types 'application/x-www-form-urlencoded' or 'multipart/form-data' are allowed by default and do not need to be explicitly configured in the allowed list
NSERR_AS_TOTAL_IMPORT_SIZE_LIMIT_EXCEEDED	3335	0xd07	Update is successful but update operation of this import object exceeded total import size limit. Now total import size limit is changed to newer value.
NSERR_AS_MAX_TOTAL_IMPORT_LIMIT_EXCEEDED	3336	0xd08	Update operation of this import object exceeded maximum total import size limit. Please remove the object.
NSERR_INVALID_DEFAULT_SIGNATURE_NAME	3337	0xd09	Signature name must only be '*Default Signatures' or '*Xpath Injection Patterns'
NSERR_AS_SIG_OFFSET_GTE_DEPTH	3338	0xd0a	Signature file has a pattern with offset attribute greater than or equal to the depth attribute. See /var/log/ns.log for more details
NSERR_AS_SQLWILDCHAR_INVALID_ATTRIBUTE	3339	0xd0b	Imported signatures has invalid wild char attributes.
NSERR_AS_WILDCHAR_LITERAL_EXCEED_MAXLEN	3340	0xd0c	Maximum allowed length for type \"LITERAL\" for wild char exceeded.
NSERR_AS_SIG_EXCLAMATION_LITERAL	3341	0xd0d	Cannot have an exclamation point alone as a literal
NSERR_AS_REMOVE_FAILED_RENAME_FAILED	3342	0xd0e	Problem in removing resource - file rename failed
NSERR_AS_ARCHIVE_PROFILE_FAILED	3343	0xd0f	Some commands failed during archive profile. Please see /var/log/ns.log for more details.
NSERR_AS_RESTORE_PROFILE_FAILED	3345	0xd11	Some commands failed during restore profile. Please see /var/log/ns.log for more details.
NSERR_AS_EMPTY_CONTENT_TYPE	3352	0xd18	Empty content type

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AS_SIG_FASTMATCH_BAD_LOCATION	3359	0xd1f	pattern in a bad location. See log for details.
NSERR_AS_SESSIONLESS_FFC_POST_ONLY	3360	0xd20	'POST ONLY' is same as option 'ON' in streaming
NSERR_AS_TURNING_TAGGING_OFF_FFC_ON	3361	0xd21	Form-tagging cannot be disabled and remains enabled as long as field consistency check is enabled.
NSERR_AS_NO_CREDIT_CARD_NUMBER	3363	0xd23	No such credit card number.
NSERR_AS_EXIST_CREDIT_CARD_NUMBER	3364	0xd24	Credit card number is already in use.
NSERR_AS_NO_SUCH_IMPORTED_OBJECT	3365	0xd25	No such imported object.
NSERR_AS_LEARNING_SQL_ERROR	3366	0xd26	SQL Error when performing learning operation
NSERR_AS_IMPORT_FILE_IS_EMPTY	3367	0xd27	Imported file is empty
NSERR_AS_IMPORT_OVERWRITE_OPTION_NOT_SUPPORTED	3368	0xd28	Overwrite option is not supported.
NSERR_AS_BAD_INSPECT_CONTENT_TYPE	3369	0xd29	Invalid inspection content type
NSERR_AS_GEO_LOCATION_LOGGING_ENABLED	3376	0xd30	GeoLocationLogging only works when CEFLogging is enabled
NSERR_AS_NO_SUCH_WSDL_OBJECT	3377	0xd31	WSDL does not exist
NSERR_AS_NO_SUCH_XMLSCHEMA_OBJECT	3378	0xd32	XMLSchema does not exist
NSERR_AS_NO_SUCH_XMLERRORPAGE_OBJECT	3379	0xd33	XMLErrorPage does not exist
NSERR_AS_NO_SUCH_SIGNATURE_OBJECT	3380	0xd34	Signature does not exist
NSERR_AS_NO_SUCH_ARCHIVE_OBJECT	3381	0xd35	Archive does not exist
NSERR_AS_NO_SUCH_HTMLERRORPAGE_OBJECT	3382	0xd36	HTMLErrorPage does not exist
NSERR_AS_XSS_ALLOWED_TAG_INVALID_ATTRIBUTE	3393	0xd41	Xss allowed Tag has invalid

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AS_XSS_ALLOWED_ATTR_INVALID_ATTRIBUTE	3394	0xd42	Xss allowed attribute has invalid attribute
NSERR_AS_SQL_TRANSFORM_INVALID_ATTRIBUTE	3395	0xd43	SQL transform has invalid attribute
NSERR_AS_SRC_NOT_DEFAULT	3403	0xd4b	?
NSERR_AS_SRC_INVALID	3404	0xd4c	Invalid source
NSERR_AS_IMPORT_NOT_DEFAULT_WITH_SHA1	3405	0xd4d	Name should be *Default Signatures when -sha1 is specified

© 1999-2017 Citrix Systems, Inc. All rights reserved.

BASE ERRORS

Jul 07, 2017

BASE ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NOTRUNNING	256	0x100	NetScaler not running
NSERR_PERM	257	0x101	Operation not permitted
NSERR_NOENT	258	0x102	No such resource
NSERR_SRCH	259	0x103	No such process
NSERR_INTR	260	0x104	Interrupted system call
NSERR_IO	261	0x105	Input/output error
NSERR_NXIO	262	0x106	Device not configured
NSERR_2BIG	263	0x107	Argument list too long
NSERR_NOEXEC	264	0x108	Exec format error
NSERR_BADF	265	0x109	Bad file descriptor
NSERR_CHILD	266	0x10A	No child processes
NSERR_DEADLK	267	0x10B	Resource deadlock avoided
NSERR_NOMEM	268	0x10C	Cannot allocate memory
NSERR_ACCES	269	0x10D	Permission denied
NSERR_FAULT	270	0x10E	Bad address

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NOTBLK	271	0x10F	Block device required
NSERR_BUSY	272	0x110	Device busy
NSERR_EXIST	273	0x111	Resource already exists
NSERR_XDEV	274	0x112	Cross-device link
NSERR_NODEV	275	0x113	Operation not supported by device
NSERR_NOTDIR	276	0x114	Not a directory
NSERR_ISDIR	277	0x115	Is a directory
NSERR_INVAL	278	0x116	Invalid argument
NSERR_NFILE	279	0x117	Too many open files in system
NSERR_MFILE	280	0x118	Too many open files
NSERR_NOTTY	281	0x119	Inappropriate ioctl for device
NSERR_TXTBSY	282	0x11A	Text file busy
NSERR_FBIG	283	0x11B	File too large
NSERR_NOSPACE	284	0x11C	No space left on device
NSERR_SPIPE	285	0x11D	Illegal seek
NSERR_ROFS	286	0x11E	Read-only file system
NSERR_MLINK	287	0x11F	Too many links
NSERR_PIPE	288	0x120	Broken pipe
NSERR_DOM	289	0x121	Numerical argument out of domain

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_RANGE	290	0x122	Result too large
NSERR_AGAIN	291	0x123	Resource temporarily unavailable
NSERR_INPROGRESS	292	0x124	Operation now in progress
NSERR_ALREADY	293	0x125	Operation already in progress
NSERR_NOTSOCK	294	0x126	Socket operation on non-socket
NSERR_DESTADDRREQ	295	0x127	Destination address required
NSERR_MSGSIZE	296	0x128	Message too long
NSERR_PROTOTYPE	297	0x129	Protocol wrong type for socket
NSERR_NOPROTOOPT	298	0x12A	Protocol not available
NSERR_PROTONOSUPPORT	299	0x12B	Protocol not supported
NSERR_SOCKTNOSUPPORT	300	0x12C	Socket type not supported
NSERR_OPNOTSUPP	301	0x12D	Operation not supported
NSERR_PFNOSUPPORT	302	0x12E	Protocol family not supported
NSERR_AFNOSUPPORT	303	0x12F	Address family not supported by protocol family
NSERR_ADDRINUSE	304	0x130	Address already in use
NSERR_ADDRNOTAVAIL	305	0x131	Can't assign requested address
NSERR_NETDOWN	306	0x132	Network is down
NSERR_NETUNREACH	307	0x133	Network is unreachable
NSERR_NETRESET	308	0x134	Network dropped connection on reset

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CONNABORTED	309	0x135	Software caused connection abort
NSERR_CONNRESET	310	0x136	Connection reset by peer
NSERR_NOBUFS	311	0x137	No buffer space available
NSERR_ISCONN	312	0x138	Socket is already connected
NSERR_NOTCONN	313	0x139	Socket is not connected
NSERR_SHUTDOWN	314	0x13A	Can't send after socket shutdown
NSERR_TOOMANYREFS	315	0x13B	Resource in use
NSERR_TIMEDOUT	316	0x13C	Operation timed out
NSERR_CONNREFUSED	317	0x13D	Connection refused
NSERR_LOOP	318	0x13E	Too many levels of symbolic links
NSERR_NAMETOOLONG	319	0x13F	File name too long
NSERR_HOSTDOWN	320	0x140	Host is down
NSERR_HOSTUNREACH	321	0x141	No route to host
NSERR_NOTEMPTY	322	0x142	Directory not empty
NSERR_PROCLIM	323	0x143	Too many processes
NSERR_USERS	324	0x144	Too many users
NSERR_DQUOT	325	0x145	Disc quota exceeded
NSERR_STALE	326	0x146	Stale NFS file handle
NSERR_REMOTE	327	0x147	Too many levels of remote in path

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_BADRPC	328	0x148	RPC struct is bad
NSERR_RPCMISMATCH	329	0x149	RPC version wrong
NSERR_PROGUNAVAIL	330	0x14A	RPC prog. not avail
NSERR_PROGMISMATCH	331	0x14B	Program version wrong
NSERR_PROCUNAVAIL	332	0x14C	Bad procedure for program
NSERR_NOLCK	333	0x14D	No locks available
NSERR_NOSYS	334	0x14E	System Call not supported; Possible Reason: NetScaler is not running
NSERR_FTYPE	335	0x14F	Inappropriate file type or format
NSERR_AUTH	336	0x150	Authentication error
NSERR_NEEDAUTH	337	0x151	Need authenticator
NSERR_WOULDBLOCK	338	0x152	Operation would block
NSERR_NOCODE	339	0x153	Feature is not implemented
NSERR_NOTSUSER	340	0x154	Not super-user
NSERR_BIGDATA	341	0x155	Data size is too big
NSERR_SMALLDATA	342	0x156	Data size is too small
NSERR_NOMORENT	343	0x157	No more entry in table
NSERR_NOSERVICE	344	0x158	No Service
NSERR_OSERROR	345	0x159	Operating system error
NSERR_NONEXPCMD	346	0x15A	Unexpected command

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CMDPROPFAIL	347	0x15B	Command propagation failed
NSERR_TOOMANYNODES	348	0x15C	Too many nodes
NSERR_SECONDARYFAIL	349	0x15D	Command failed on secondary node, but succeeded on primary node. Configuration will be synchronized to ensure secondary and primary have same configuration.
NSERR_INVALBACKUP	350	0x15E	Invalid backup vserver
NSERR_NOSERVER	351	0x15F	No such server
NSERR_LOGINREQD	352	0x160	Login is required
NSERR_RPCINVAL	353	0x161	RPC invalid argument
NSERR_NOUSER	354	0x162	Invalid username or password
NSERR_INVALPASSWD	355	0x163	Invalid username or password
NSERR_LICENSE	356	0x164	Feature(s) not licensed
NSERR_DEFERRED	357	0x165	The command is stored for later execution
NSERR_PROPAUTHFAIL	358	0x166	Command propagation failed due to user authentication problems
NSERR_NODELSUSER	359	0x167	The superuser cannot be removed
NSERR_NOMODSUSER	360	0x168	Permissions for this superuser cannot be modified
NSERR_INVALNODEID	362	0x16A	Invalid node ID specified
NSERR_NOTOPHA	363	0x16B	Operation not permitted on stand-alone node
NSERR_NOOPPEERBAD	364	0x16c	Operation not possible due to invalid peer state. Rectify and retry.
			Operation not possible as system state is

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			information.
NSERR_NOPNOW	366	0x16e	Operation not possible now. Please wait for system to stabilize before retrying.
NSERR_NOOPPRI	367	0x16f	Operation permitted only on secondary node.
NSERR_NOOPSEC	368	0x170	Operation permitted only on primary node.
NSERR_REDIRECT	369	0x171	Redirect request (VIP down)
NSERR_BUFOVERFLOW	370	0x172	Buffer overflow occurred
NSERR_NOUSERPOLICY	371	0x173	No command policy defined, permission denied
NSERR_NOSYSGROUP	372	0x174	System group does not exist
NSERR_NOSYSCMDPOL	373	0x175	System command policy does not exist
NSERR_HAIPV6PT	374	0x176	IPV6 Feature cannot be disabled while HA is running over IPv6.
NSERR_HANSIPV6	375	0x177	IPv6 NSIP cannot be changed while HA is running over IPv6.
NSERR_NSIPV6ACTIVE	376	0x178	IPv6 NSIP is not in ACTIVE state, rectify and retry.
NSERR_RTMON_STANDALONE	377	0x179	Unbind all route monitors before making node standalone.
NSERR_IPV6FEAT_DISABLED	378	0x17a	IPv6 Feature Disabled
NSERR_DHCP_PBR_DEPENDENT	379	0x17b	PBR rule present with nexthop dependent on released DHCP lease
NSERR_IPSECPROF_PERM	380	0x17c	Tunnel protocol IPSEC must have ipsec profile set to 'none'.
NSERR_IFPAIR_BOUND	381	0x17d	Interface(s) already member of other interface pair
NSERR_IFPAIR_NOTPAIR	382	0x17e	Number of member interfaces need to be exactly 2
NSERR_IPRANGENOTALLOWD	394	0x18A	Range not allowed for wildcard IP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_IVALIDIPRANGE	395	0x18B	Invalid range value
NSERR_IPRANGEMAXLIMIT	396	0x18C	Range value greater than maximum limit 254
NSERR_IP_NOT_EXIST	397	0x18D	IP does not exist for default or specified td
NSERR_TOOMANYRULES	398	0x18E	Maximum number of ACLs on the system has been exceeded
NSERR_PASSWD_LEN_MIN_8	399	0x18F	Password too short - minimum length is 8 characters
NSERR_SYNC_DISABLED	410	0x19A	Synchronization disabled
NSERR_NODE_DISABLED	411	0x19B	Node disabled
NSERR_SYNC_PROGRESS	412	0x19C	Synchronization is in progress
NSERR_ADNS_PERM	413	0x19D	Operation not permitted on ADNS service
NSERR_NOINC_IPSAMESUBNET	414	0x19E	Peer IP should be on the NSIP subnet if INC mode is disabled
NSERR_INVALID_PEERIP	415	0x19F	Not a valid peer IP
NSERR_REDIRECT307	416	0x1A0	Redirect request (send 307 temporary redirect
NSERR_INVALHOSTNAME	417	0x1A1	Invalid hostname
NSERR_REWRITE_NOT_SUPPORTED	418	0x1A2	Port rewrite not supported on this vserver type
NSERR_IPCHG_DENY	419	0x1A3	IP address change not permitted on this entity
NSERR_IPCHG_GSLB	420	0x1A4	GSLB local service IP cannot be changed. Please change the corresponding vserver's IP
NSERR_GSLB_IPCHG	421	0x1A5	corresponding GSLB service IP has also been changed.
NSERR_NOAUDITSRVC	422	0x1A6	No audit service running on the specified port

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_LACPKEY_NOTSET	423	0x1A7	LACP key not set
NSERR_CHANNEL_INUSEBYLACP	424	0x1A8	Channel in use by LACP
NSERR_LACPENABLED	425	0x1A9	LACP enabled interface cannot be bound manually to a channel
NSERR_IFACEMANUALLYBOUND	426	0x1AA	LACP cannot be enabled, interface manually bound to a channel
NSERR_INTRECINUSE	427	0x1AB	Domain name for the reverse domain name already exists
NSERR_IPPORT_VIP_CONFLICT	428	0x1AC	IP, port conflict with another entity bound to the same vserver
NSERR_IPPORT_VIP_BOUND	429	0x1AD	IP, port conflict with an entity already bound to the vserver
NSERR_NO_BACKUP_VIP	430	0x1AE	No such backup vserver
NSERR_REQ_SET_ARGS	431	0x1AF	No arguments to set
NSERR_SVCGRP_MEMBER_NAMECONFLICT	432	0x1B0	Name conflicts with an existing service or service group member name
NSERR_SERVER_NAME_EXIST	433	0x1B1	Server name already exists
NSERR_MAX_SERVICE_BINDING_ON_VSERVER	434	0x1B2	Maximum services bound to vserver exceeded
NSERR_MAX_SVC_ENTITY_BINDING_ON_SVCGROUP	435	0x1B3	Maximum services bound to service group exceeded
NSERR_FSMALL	436	0x1B4	File too small
NSERR_INTOFLOW	437	0x1B5	Integer overflow
NSERR_ASYNC_BLOCKED	438	0x1B6	PI has blocked for body accumulation
NSERR_SACL_CLEARPENDING	439	0x1B7	Clear in progress
NSERR_SACL_NAME_EXISTS	440	0x1B8	An ACL with the same name exists
NSERR_SACL_SUPERSET_EXISTS	441	0x1B9	Operation not permitted. (An overlapping

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code (Decimal)	Error Code (Hex)	Error Message
NSERR_SACL_SUBSET_EXISTS	442	0x1BA	Operation not permitted. (A rule which is subset of this rule already exists)
NSERR_NOINC_ROUTEMONITOR	443	0x1BB	Route monitors should be added only if INC mode is enabled
NSERR_SESSION_EXPIRED	444	0x1BC	Session expired or killed. Please login again
NSERR_SESSION_EXCEEDED	445	0x1BD	Session limit exceeded
NSERR_CFE_CONN_EXCEEDED	446	0x1BE	Connection limit to CFE exceeded
NSERR_CFE_KILLSELF	449	0x1C1	Invalid session id. Current session cannot be killed
NSERR_CFE_INCOMPLETESESSION	450	0x1C2	Invalid session. Response/challenge is incomplete
NSERR_ACL_ALREADYINSTATE	451	0x1C3	Entity already in this state
NSERR_CFE_SESSION_NOEXIST	452	0x1C4	Session does not exist
NSERR_SYSGROUP_USER_EXISTS	453	0x1C5	User already bound to system group
NSERR_SYSGROUP_POLICY_EXISTS	454	0x1C6	Policy already bound to system group
NSERR_SLEEP	455	0x1C7	Sleep process sleep
NSERR_PPEDIE	456	0x1C8	Your PPE is no more
NSERR_NOCONN_CMDPROPFAIL	457	0x1C9	Unable to establish connection with the secondary. Command propagation failed
NSERR_TIMEOUT_SECONDARY	458	0x1CA	There is no response from the secondary. Propagation timed out
NSERR_RPC_TIMEOUT	459	0x1CB	RPC timeout
NSERR_NOT_SUPPORTED	460	0x1CC	Feature not supported in this release
NSERR_NOENT_VSERVER	461	0x1CD	No such Vserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NOENT_SVC	462	0x1CE	No such service
NSERR_NOENT_SVC_SVCGRP	463	0x1CF	No such service/serviceGroup
NSERR_HA_NOV4_NETMASK	464	0x1D0	IPv4 netmask is not required for IPv6
NSERR_MAX_VSEVRVER_BINDINGS_TO_SERVICE	465	0x1D1	Maximum vservers bound to a service exceeded
NSERR_SACL6_CLEARPENDING	466	0x1D2	Simple ACL6 Clear in progress
NSERR_SACL6_NAME_EXISTS	467	0x1D3	Simple ACL6 with the same name exists
NSERR_SACL6_SUPERSET_EXISTS	468	0x1D4	Operation not permitted. (An overlapping rule already exists)
NSERR_SACL6_SUBSET_EXISTS	469	0x1D5	Operation not permitted. (A rule which is subset of this rule already exists)
NSERR_NO_BACKUP_VIP_BOUND	470	0x1D6	No backupVserver bound
NSERR_SESSION_EXPIRED_REDIRECT	471	0x1D7	Session expired, Please login again
NSERR_NOENT_VLAN	472	0x1D8	Vlan does not exist
NSERR_MAX_SVC_ENTITY	473	0x1D9	Maximum service limit reached
NSERR_CL_EXIST	474	0x1DA	Resource already exists in cluster
NSERR_OWNERNODE_NOTALLOWED	475	0x1DB	Can not specify owner node.
NSERR_OWNERNODE_NOT_SPECIFIED	476	0x1DC	Owner node is not specified.
NSERR_NOT_SUPPINCLUSTER	477	0x1DD	Operation not supported in Cluster.
NSERR_VSERVER_TYPE_MISMATCH	478	0x1DE	Vserver type mismatch.
NSERR_ITER_END	479	0x1DF	End of iteration reached.
NSERR_CONNECT_TO_CLUSTER_IP	1382	0x566	Please connect to cluster IP and run showtechsupport on that node.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_LB_AUDIT_IP_DENY	1383	0x567	Changing IP address of LB vserver not permitted when Audit server also referring to same server_info .
NSERR_IC_OFF_DISABLING_FEO	1384	0x568	IC is disabled, but FEO is enabled.
NSERR_SYNC_VLAN_EXIST	1385	0x569	Please unset sync vlan, before removing the HA node.
NSERR_CMDPROP_DISABLED	1450	0x5AA	Command propagation disabled.
NSERR_NOTHEME	1451	0x5AB	Portal theme doesn't exist
NSERR_THEMEEXISTS	1452	0x5AC	Portal theme already exists
NSERR_THEME_INUSE	1453	0x5AD	Portal theme in use
NSERR_NO_CONTENT	1454	0x5AE	204 No Content
NSERR_ARP_IFACE_NODE_MISMATCH	1440	0x5a0	Interface and node mismatch
NSERR_HA_BOTH_PRIMARY	1441	0x5a1	Node priority cannot be primary for both the nodes in HA setup
NSERR_PROP_DISABLED	1448	0x5a8	Propagation disabled
NSERR_HA_FAILOVER_PERM	1449	0x5a9	Failover not permitted as one node is up and another node is down
NSERR_NOEULA	938	0x3AA	Eula name doesn't exist
NSERR_EULAEXISTS	939	0x3AB	Eula with same name already exists
NSERR_EULA_INUSE	940	0x3AC	Eula name in use
NSERR_SYSLOG_TCP_CONFIG_ERR	1459	0x5B3	tcpProfileName/maxLogDataSizeToHold cannot be set when transport is UDP
NSERR_SYSLOG_LOOPBACK_OR_LOCAL_IP_NOTALLOWED	1460	0x5B4	TCP transport is not allowed for local logging(when serverIP is configured as Loopback IP or NetScaler's local IP)
NSERR_MAX_CONC_SAVECONFIG	1461	0x5B5	Maximum number of concurrent save configs are in progress. Try again later.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SYSLOG_LBMETHOD_NOTALLOWED	1462	0x5B6	LB method RR or LC is allowed only for service type SYSLOGTCP/SYSLOGUDP
NSERR_SYSLOG_PERSISTENCE_NOTSUPPORTED	1463	0x5B7	Persistency is not supported for SYSLOGTCP and SYSLOGUDP service types
NSERR_SYSLOG_HASHLBMETHODS_NOTALLOWED	1464	0x5B8	Hash based LB methods which depend on packet parameters cannot be set for SYSLOGTCP/SYSLOGUDP service types
NSERR_SYSLOG_LEASTCONN_NOTSUPPORTED	1467	0x5BB	LEASTCONNECTION algorithm is not supported for SYSLOGTCP/SYSLOGUDP service types
NSERR_SYSLOG_LBVSERVER_IPPORT_NOTALLOWED	1468	0x5BC	IP address and port cannot be configured for LB vserver of type SYSLOGUDP/SYSLOGTCP
NSERR_SYSLOG_NOLBVIP	1469	0x5BD	No such LB vserver exists
NSERR_SYSLOG_LBPROTOCOL_NOTALLOWED	1470	0x5BE	LB vserver of type other than SYSLOGUDP/SYSLOGTCP cannot be set to syslogaction
NSERR_SYSLOG_TRANSPORT_UNSET_NOTALLOWED	1471	0x5BF	Transport cannot be unset when LB vserver name is configured
NSERR_SYSLOG_TRANSPORT_MISMATCH	1472	0x5C0	Mismatch in transport type of LB vserver and syslog action
NSERR_SYSLOG_LBVSRVR_INUSE_BY_SYSACT	1473	0x5C1	The LB vserver is being accessed by syslog action
NSERR_SYSLOG_LBVSRVR_RENAME_NOTSUPPORTED	1474	0x5C2	Renaming LB vserver is not supported for SYSLOGUDP/SYSLOGTCP service types
NSERR_SYSLOG_INVALID_SERVIP	1475	0x5C3	ServerIP cannot be configured as 0.0.0.0 or 0 for syslog action
NSERR_AUDITLOG_PORT_SET_UNSET_NOTALLOWED	1476	0x5C4	Port cannot be set/unset when LB vserver name/Server domain name is configured
NSERR_SYSLOG_TCPPROFILE_NOTALLOWED	1477	0x5C5	TCP profile name cannot be set when LB vserver name is configured
NSERR_SYSLOG_CANNOT_SET_LBVSNAME	1478	0x5C6	Unset the TCP profile name and then set the LB vserver name
NSERR_PORTALBINDING_NOT_SUPPORTED	1479	0x5C7	Operation not supported. Unset VPN uitheme parameter first and then try

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PORTALTHEME_NAME_NOT_ALLOWED	1480	0x5C8	\'EULA\' and \'Custom\' keywords are reserved. Please try with any other name
NSERR_AUDIT_VERSION_MISMATCH	1481	0x5C9	Audit version mismatch
NSERR_RADIUS_LISTENER_PERM	1482	0x5CA	Operation not permitted on RADIUS LISTENER service
NSERR_SYSLOG_NETPROFILE_NOTALLOWED	1483	0x5CB	Network profile name cannot be set when LB vserver name is configured
NSERR_AUDITLOG_DMNRESOLVRETRY	1484	0x5CC	Domain resolve retry can be set only when server domain name is configured
NSERR_AUDITLOG_ARGUMENT_MISSING	1485	0x5CD	One of the arguments \"Server IP/Server domain name/LB vserver name\" is required for auditlog action
NSERR_AUDITLOG_DBS_SERVER_EXIST	1486	0x5CE	Auditlog internal DBS server with the same domain name exists
NSERR_ICAMODE_STATUS_NOT_OK	1487	0x5CF	vpn icamode status not okay
NSERR_SYSLOG_LBVSERVER_NOTALLOWED	1488	0x5D0	LB vserver name cannot be set when network profile is configured. Unset the netProfile and set LB vserver name
NSERR_AUDITLOG_SVC_EXISTS	1490	0x5D2	Auditlog service exists with this IP address

© 1999-2017 Citrix Systems, Inc. All rights reserved.

CACONF ERRORS

May 20, 2016

CACONF ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CACONF_CNFL_HEUREXP_RELEXP	480	0x1E0	Conflicting arguments, heurExpiryParam and relExpiry
NSERR_CACONF_CNFL_HEUREXP_RELEXPMILI	481	0x1E1	Conflicting arguments, heurExpiryParam and relExpiryMilliSec
NSERR_CACONF_CNFL_RELEXP_RELEXPMILI	482	0x1E2	Conflicting arguments, relExpiry and relExpiryMiliSec
NSERR_CACONF_CNFL_ABSEXP_HEUREXP	483	0x1E3	Conflicting arguments, absExpiry and heurExpiryParam
NSERR_CACONF_CNFL_ABSEXP_RELEXPMILI	484	0x1E4	Conflicting arguments, absExpiry and relExpiryMilliSec
NSERR_CACONF_CNFL_ABSEXPGMT_HERUEXP	485	0x1E5	Conflicting arguments, absExpiryGMT and heurExpiryParam
NSERR_CACONF_CNFL_ABSEXP_ABSEXPGMT	486	0x1E6	Conflicting arguments, absExpiry and absExpiryGMT
NSERR_CACONF_CNFL_ABSEXPGMT_RELEXPMILI	487	0x1E7	Conflicting arguments, absExpiryGMT and relExpiryMilliSec
NSERR_CACONF_CNFL_HITPARAMS_HITSLCTR	488	0x1E8	Conflicting arguments, hitParams and hitSelector
NSERR_CACONF_CNFL_INVALPARAMS_HITSLCTR	489	0x1E9	Conflicting arguments, invalParams and hitSelector
NSERR_CACONF_CNFL_HITPARAMS_INVLSLCTR	490	0x1EA	Conflicting arguments, hitParams and invalSelector
NSERR_CACONF_CNFL_INVALPARAMS_INVLSLCTR	491	0x1EB	Conflicting arguments, invalParams and invalSelector
NSERR_CACONF_CNFL_HITSLCTR_MATCHCOOKY	492	0x1EC	Conflicting arguments, hitSelector and matchCookies
NSERR_CACONF_CNFL_HITSLCTR_INVALREST2HOST	493	0x1ED	Conflicting arguments, hitSelector and invalRestrictedToHost

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CACONF_CNFL_HITSLCTR_IGNRPARAMVALCASE	494	0x1EE	Conflicting arguments, hitSelector and ignoreParamValueCase
NSERR_CACONF_CNFL_INVALSLCTR_MATCHCOOKY	495	0x1EF	Conflicting arguments, invalSelector and matchCookies
NSERR_CACONF_CNFL_INVALSLCTR_INVALREST2HOST	496	0x1F0	Conflicting arguments, invalSelector and invalRestrictedToHost
NSERR_CACONF_CNFL_INVALSLCTR_IGNRPARAMVALCASE	497	0x1F1	Conflicting arguments, invalSelector and ignoreParamValueCase
NSERR_CACONF_CNFL_POSRELEXP_RELEXP	498	0x1F2	Conflicting arguments, weekPosRelExpiry and relExpiry
NSERR_CACONF_CNFL_POSRELEXP_RELEXPMILI	499	0x1F3	Conflicting arguments, weekPosRelExpiry and relExpiryMiliSec
NSERR_CACONF_CNFL_ABSEXP_POSRELEXP	500	0x1F4	Conflicting arguments, absExpiry and weekPosRelExpiry
NSERR_CACONF_CNFL_ABSEXPGMT_POSRELEXP	501	0x1F5	Conflicting arguments, absExpiryGMT and weekPosRelExpiry
NSERR_CACONF_CNFL_IGNRPARAMVALCASE_HITPARAMS	502	0x1F6	Invalid arguments, ignoreParamValueCase is set while hitParams is 0
NSERR_CACONF_CNFL_HIT_INVALPARAMS_MATCHCUKY	503	0x1F7	Invalid arguments, matchCookies is set while both hitParams and invalParams are 0
NSERR_CACONF_CNFL_INVALREST2HOST_INVALPARAM	504	0x1F8	Invalid arguments, invalRestrictedToHost is set while invalParams is 0
NSERR_CACONF_CNFL_PREFETCH_PREFETCHSEC	505	0x1F9	Invalid arguments, prefetchPeriod is set while prefetch is disabled
NSERR_CACONF_CNFL_PREFETCH_PREFETCHMILI	506	0x1FA	Invalid arguments, prefetchPeriodMilliSec is set while prefetch is disabled
NSERR_CACONF_CNFL_PREFETCHMILI_PREFETCHSEC	507	0x1FB	Conflicting arguments, prefetchPeriodMilliSec and prefetchPeriod
NSERR_CACONF_ARG_LE_MIN_VAL	508	0x1FC	Invalid arguments, given value is less than the minimum value
NSERR_CACONF_ARG_GE_MAX_VAL	509	0x1FD	Invalid arguments, given value is more than the maximum value

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CACHEHA_OBJPERSIST_NOTSET	511	0x1FF	minHit Cache Object Sync to Secondary is not set, use set cache parameter to enable.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

CFE ERRORS

Jul 07, 2017

CFE ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NOSUCHIOCTL	1184	0x4a0	Command not implemented on server
NSERR_NOTARGETS	1185	0x4a1	No configured targets
NSERR_CANTRECOVER	1186	0x4a2	Configuration possibly inconsistent. Please check with the \"show configstatus\" command or reboot.
NSERR_IGNOREDIOCTL	1187	0x4a3	The command was ignored.
NSERR_REMOTECLOSE	1188	0x4a4	The remote side closed the connection.
NSERR_INVALID_TARGET	1189	0x4a5	The specified target does not exist
NSERR_FILE_ERROR	1190	0x4a6	File operation failed
NSERR_COMMENT_DROPPED	1191	0x4a7	Failed to retain all comments
NSERR_AGGREQ_TIMEOUT	1192	0x4a8	Request to Aggregator timed out
NSERR_AGGREAD	1193	0x4a9	Failed to read data from Aggregator
NSERR_RPC_CMD_DUP	1194	0x4aa	Found unexpected RPC duplicate command
NSERR_RPC_CMD_NONDUP	1195	0x4ab	Found unexpected RPC command
NSERR_CFE_PE_COMM	1196	0x4ac	Communication error with the packet engine
NSERR_AGG_CONFAIL	1197	0x4ad	Failed to connect to the aggregator

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AGG_INVALIDRESPONSE	1199	0x4af	Invalid response from the aggregator
NSERR_NONTPSVR	1200	0x4b0	Cannot enable ntpd when there is no ntp server configured
NSERR_AGG_SENDFAIL	1201	0x4b1	Failed to send to aggregator
NSERR_CFE_ASLEARN_COMM	1202	0x4b2	Communication error with aslearn
NSERR_READONLY_SESSION	1203	0x4b3	Session is read-only; connect to the Cluster IP address to modify the configuration
NSERR_WRONG_PE_STATE	1204	0x4b4	The packet engine is in the wrong state to receive this command
NSERR_CFE_VALIDATION_FAILED	1205	0x4b5	Config validation failed
NSERR_XPATH_INTERNAL	1206	0x4b6	XPATH Internal Error
NSERR_CFE_CALLHOME_COMM	1207	0x4b7	Communication error with callhome
NSERR_BIND_NOENT	1208	0x4b8	No such bind resource
NSERR_PPNETMASK	1209	0x4b9	Netmask length 31 supports only point to point communications
NSERR_CFE_CLFSYNCD_COMM	1210	0X4ba	Communication error with cluster filesync server
NSERR_DB_INTERNAL	1211	0x4bb	Config DB Internal Error
NSERR_MIPDEPRECATED	1212	0x4bc	MIP is deprecated. Please use SNIP
NSERR_KEKEXISTS	1213	0x4bd	Cannot create KEK. KEK already exists.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

CLI ERRORS

Jul 07, 2017

CLI ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_TCPCONNFAIL	1024	0x400	Connection failed
NSERR_LOGINFAIL	1025	0x401	Login failed
NSERR_NOLOGIN	1026	0x402	Not logged in
NSERR_AUTHTIMEOUT	1027	0x403	Not logged in or connection timed out
NSERR_NOT_PRIMARY	1028	0x404	You are connected to a secondary node; configuration changes made in this session will not be propagated to, or saved on, other nodes
NSERR_REMOTEOP	1029	0x405	Operation cannot be performed from remote login
NSERR_CONNLOST	1030	0x406	Connection with NetScaler lost
NSERR_RPCDATAMISMATCH	1031	0x407	Communication error (RPC data-size mismatch)
NSERR_RPCBADREPLY	1032	0x408	Communication error (bad RPC reply)
NSERR_UNABLETOPROMPT	1033	0x409	Unable to display secondary prompt
NSERR_LOGIN_CHALLENGED	1034	0X40A	Login challenged
NSERR_LOGOUT_SAVE	1035	0X40B	You are leaving without saving changes. Save changes now (Y/N)? [N]:
NSERR_CLEAR_CONFIG	1036	0X40C	Are you sure you want to clear the configuration(Y/N)? [N]:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			(Y/N)? [N]:
NSERR_SHUTDOWNNS	1038	0X40E	Are you sure you want to completely stop NetScaler (Y/N)? [N]:
NSERR_REBOOT_SAVE	1039	0X40F	Configuration not Saved. Are you sure you want to restart NetScaler (Y/N)? [N]:
NSERR_USERABORT	1040	0x410	User requested abort
NSERR_EOF	1041	0x411	EOF
NSERR_INTERRUPT	1042	0x412	Interrupted
NSERR_INTERNAL	1043	0x413	Internal error
NSERR_NOT_CLIP	1044	0x414	You are connected to NSIP of a cluster node; configuration changes made in this session will not be propagated to, or saved on, other cluster nodes
NSERR_EULA_NOTACCEPTED	1045	0x415	End User License Agreement not accepted. Login from CLI/GUI to accept it.
NSERR_STRMAXLEN_255	1048	0x418	String length exceeds maximum. Allowed maximum length is 255 bytes.
NSERR_STRMAXLEN_32	1049	0x419	String length exceeds maximum. Allowed maximum length is 32.
NSERR_NOPREFIX_LENGTH	1050	0x41A	Prefix length must be given with network
NSERR_INVALIDPOLLVAL	1051	0x41B	Minpoll value should not be greater than maxpoll
NSERR_SHELL_ACCESS	1052	0x41C	User has shell access
NSERR_NORESPONSE	1056	0x420	No response from NetScaler
NSERR_IOERROR	1057	0x421	I/O error
NSERR_ENV	1058	0x422	Environment error
NSERR_CMDSFAILED	1059	0x423	Some commands failed

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INVALID_TCP_OPTION_TYPE	1061	0x425	Invalid tcp option type value. Allowed type values are 0 to 254
NSERR_LICEXPIRED	1066	0x42A	Features(s) license expired
NSERR_FEATDISABLED	1067	0x42B	Feature(s) not enabled
NSERR_MAXLIMIT	1072	0x430	Maximum resource limit reached
NSERR_SET_NOSUPPORT	1073	0x431	Set operation not supported
NSERR_INVALIDVALUE	1074	0x432	Invalid value
NSERR_INVALIDNAME	1075	0x433	Invalid name; names must begin with an alphanumeric character or underscore and must contain only alphanumerics, '_', '#', '.', ' ', ':', '@', '=' or '-'
NSERR_INVALIDPASSWORD	1076	0x434	Invalid password. Password must contain atleast one lower case character, one upper case character, one numeric character and one special character when strong password is enabled.
NSERR_INVALIDMINPASSWORDLENGTH	1077	0x435	Minimum password length should be greater than or equal to 4 when strong password is enabled
NSERR_ENABLINGPASSWORDCOMPLEXITY	1078	0x436	Strong Password now enabled. Please ensure all the existing user passwords adhere to this restriction. Minimum Password Length is set to 4 as default.
NSERR_INVALIDPASSWORDLENGTH	1079	0x437	Password length should adhere to minimum password length value in system parameter settings.
NSERR_CANNOTSETMINPASSWORDLENGTH	1080	0x438	Strong Password needs to be enabled to set the Minimum Password Length value.
NSERR_NOSUCHCMD	1088	0x440	No such command
NSERR_CMDAMBIGUOUS	1089	0x441	Ambiguous command name

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ARGVALMISSING	1091	0x443	Required argument value missing
NSERR_ARGSMUTEX	1092	0x444	Arguments cannot both be specified
NSERR_ARGPREREQ	1093	0x445	Argument pre-requisite missing
NSERR_ARGSTOOFEW	1094	0x446	Too few arguments
NSERR_ARGMISSING	1095	0x447	Required argument missing
NSERR_ARGORDER	1096	0x448	Argument(s) out of order
NSERR_ARGVALBAD	1097	0x449	Invalid argument value
NSERR_ARGVALSEQ	1098	0x44A	Arguments cannot have the same value
NSERR_ARGAMBIGUOUS	1099	0x44B	Ambiguous argument name
NSERR_SYNCGSLBCONFIG	1100	0x44C	Please confirm whether you want to sync-config (Y/N)? [N]:
NSERR_SYNCGSLBCONFIG_WARN	1101	0x44D	Syncing config may cause configuration loss on other site.
NSERR_ARGVALSNEQ	1104	0x450	Arguments must have the same value
NSERR_ARGVALAMBIGUOUS	1105	0x451	Ambiguous argument value
NSERR_STRMAXLEN	1106	0x452	String length exceeds maximum
NSERR_STRMINLEN	1107	0x453	String too short
NSERR_INTMAXVAL	1108	0x454	Integer value exceeds maximum
NSERR_INTMINVAL	1109	0x455	Integer value below minimum
NSERR_INVALIDIP	1110	0x456	Invalid IP address

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_TOOMANYVALS	1111	0x458	Argument has too many values
NSERR_BADRANGE	1113	0x459	Range minimum exceeds range maximum
NSERR_EXPRQUOTES	1114	0x45A	Expression must be enclosed in quotes
NSERR_BADQUOTES	1115	0x45B	Unmatched quote
NSERR_INVALIDRANGE	1116	0x45C	Invalid range specification
NSERR_MISMATCHRANGES	1117	0x45D	Ranges are different sizes
NSERR_MULTIRANGES	1118	0x45E	Only one range allowed per argument
NSERR_NOMATCHCHAR	1119	0x45F	Unmatched character
NSERR_NEEDREBOOT	1120	0x460	The configuration must be saved and the system rebooted for these settings to take effect
NSQ_SAVECONFIG	1121	0x461	Do you want to save the new configuration?
NSERR_NOTSAVED	1122	0x462	The configuration changes were not saved - they will be lost on the next reboot
NSERR_CMDEXEC	1123	0x463	Error(s) occurred - some or all changes may not be applied
NSQ_REBOOT	1124	0x464	Do you want to reboot the system now?
NSERR_NOREBOOT	1125	0x465	The configuration changes will not take effect until the system is rebooted
NSERR_PGMFAILED	1126	0x466	External program failed
NSERR_CTXMODE	1127	0x467	Contextual CLI mode
NSERR_CMDINCOMPLETE	1128	0x468	Incomplete command
NSERR_CMDOUTOFCTX	1129	0x469	Command not valid here

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CONFIGSAVED	1130	0x46A	The running configuration has been saved
NSERR_CONFIGCLEARED	1131	0x46B	The running configuration has been cleared
NSERR_REBOOTING	1132	0x46C	NetScaler is rebooting now
NSERR_NOCONFIGSAVE	1133	0x46D	The running configuration has not changed
NSERR_REGEXNOANCHOR	1135	0x46F	First character of regular expression must be '^'
NSERR_REGEXNOMATCH	1136	0x470	Regular expression does not match
NSERR_REGEXINVALID	1137	0x471	Invalid regular expression
NSERR_REGEXNOTALLOWED	1138	0x472	Regular expression not allowed
NSERR_REGEXNOCMD	1139	0x473	No matches
NSERR_INVALIDIPV6_FORMAT	1140	0x474	Invalid IPv6 address
NSERR_INVALIDIPV6_TWO_DOUBECOLON	1141	0x475	Double-colon can appear only once in an IPv6 address
NSERR_INVALIDIPV6_NOPREFIX_LENGTH	1142	0x476	Prefix length must be given with an IPv6 address
NSERR_INVALIDIPV6_PREFIX_VALUE	1143	0x477	Prefix length must be in the range of 0-128
NSERR_TERMNAMEINVALID	1144	0x478	Invalid term name
NSERR_TERMINVALID	1145	0x479	Invalid syntax in term
NSERR_FORCEFAILOVER	1146	0x47a	Please confirm whether you want force-failover (Y/N)? [N]:
NSERR_FORCEFAIL_HEALTH_WARN	1147	0x47b	Force Failover may cause configuration loss, peer health not optimum. Reason(s):
NSERR_HELLOTIME_MULTIPLE	1148	0x47c	Invalid value. Hellotime Interval must be a multiple of 200

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ERROUTFILENAME	1150	0x47e	Character '/' not allowed in outfilename
NSERR_RNATIPDEL	1151	0x47f	address is bound to rnat config, can not be removed
NSERR_INVALIDALIAS	1152	0x480	Invalid alias
NSERR_NOSUCHALIAS	1153	0x481	No such alias
NSERR_ALIASEXISTS	1154	0x482	Alias already exists
NSERR_NOSUCHFILE	1155	0x483	No such file
NSERR_NOTREGFILE	1156	0x484	Not a file
NSERR_DEPRCMD	1157	0x485	Command deprecated
NSERR_DEPRARG	1158	0x486	Argument deprecated
NSERR_NOTLOGFILE	1159	0x487	Not a NetScaler log file
NSERR_NOPLENFORIPV6RANGE	1160	0x488	Prefix length cannot be specified for an IPv6 range
NSERR_INVALIDINT	1161	0x489	Integer not in range
NSERR_CMDAMBIGUOUS_USECOMPLETIONSOPTIONS	1162	0x48A	Ambiguous (use cmd completion for options)
NSERR_METADATA_INVAL_ENTITYTYPE	1163	0x48B	Invalid Entity Type
NSERR_SETNOTEXIST	1164	0x48C	set operation does not exist
NSERR_SETARGNOTEXIST	1165	0x48d	set operation for one of the modified argument does not exist
NSERR_INVALID_DS	1166	0x48e	Invalid datasource
NSERR_NOSUCHCOUNTER	1167	0x48F	No such counter

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INVALIDIPMASK	1168	0x490	Invalid IP address mask
NSERR_INVALIDIPPAT	1169	0x491	Invalid IP address pattern
NSERR_BADRC	1170	0x492	Unexpected return code
NSERR_INVALIDRANGETYPE	1171	0x493	Inappropriate range values for value type
NSERR_INVALIDRANGEVAL	1172	0x494	Invalid range value
NSERR_INVALIDIPV6_PREFIX_LENGTH	1173	0x495	Prefix length must not be given
NSERR_ENTITYDELETE_FAIL	1174	0x496	Can not delete active entity
NSERR_DONE	1175	0X497	This is the last result
NSERR_RNATIPDEL2	1176	0x498	An RNAT address requires an IP address to exist
NSERR_INCOMPATIBLEIP	1181	0x49d	Private IP and public IP must be either IPv4 or IPv6
NSERR_TRANSCR_IP	1182	0x49e	For transparent CR vserver, you cannot specify IP address
NSERR_PASSWORD_MISMATCH	1183	0x49f	Passwords entered do not match
NSERR_LEADING_TRAILING_SPACES	1053	0x41d	Leading or Trailing spaces in string
NSERR_SAVENOTDONE	1054	0x41e	The running configuration is not saved
NSERR_INVALIDIP_VSVRTYPE	1178	0x49a	Invalid IP address or Vserver Type
NSERR_HELLOTIME_DEADINTERVAL_COMBINATION	1177	0x499	The Hello Interval must be set to 200 msec when dead interval is less than 3 sec
NSERR_MAXSESSION	1179	0x49b	Maxsession cannot be set for this user

© 1999-2017 Citrix Systems, Inc. All rights reserved.

CLUSTER ERRORS

May 20, 2016

CLUSTER ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CLNOTEXIST	2448	0x990	Cluster does not exist
NSERR_TOOMANYCLS	2449	0x991	This node is already added to another cluster
NSERR_RMLOCALNODE	2450	0x992	Local Node cannot be deleted
NSERR_BKPLANEENABLED	2451	0x993	Cluster Backplane is enabled on this interface
NSERR_LOCALNODE_NOTCONFIGURED	2452	0x994	Clustering cannot be enabled as the local node is not a member of the cluster.
NSERR_HACL_CANNOT_COEXIST	2453	0x995	Clustering and HA cannot coexist.
NSERR_CLACPKEYNOTCRT	2454	0x996	LACP key for Node/cluster is not correct
NSERR_CLNODE_EXIST	2455	0x997	already in cluster
NSERR_BKPLANEDISABLED	2456	0x998	Disabling Cluster backplane interface
NSERR_CLAGTHROUGHPUT	2457	0x999	Currently throughput settings are not allowed on CLAG
NSERR_LICMISMATCH	2458	0x99A	License mismatch between Configuration Coordinator and local node
NSERR_CLIP_INVALIDNETMASK	2459	0x99B	CLIP netmask should be 255.255.255.255.
NSERR_BKPLANELACPDELETE	2460	0x99C	This is the last interface in LACP channel configured as backplane
NSERR_LINKSETMEMBER	2461	0x99D	Interface/Channel is Linkset member.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CONFIG_PARAM	2462	0x99E	this command in HA/Cluster mode. Use 'set psparam' command for this purpose.
NSERR_CLIP_INVALIDPREFIXLEN	2463	0x99F	CLIP prefix length should be 128.
NSERR_CLUSTER_L2NOTSUPP	2464	0x9A0	L2 Mode not supported in Cluster mode.
NSERR_LINKSET_ONEPERNODE	2465	0x9A1	Linkset can have only one Interface/Channel per Node
NSERR_LINKSET_NOTBOUND	2466	0x9A2	Interface/Channel not bound to Linkset
NSERR_LINKSET_IDINVALID	2467	0x9A3	Linkset name should be of form LS/
NSERR_CLUSTER_NBNOTSUPP	2468	0x9A4	CloudBridge is not supported in Cluster mode.
NSERR_CLUSTER_VPNNOTSUPP	2469	0x9A5	SSLVPN is not supported in Cluster mode.
NSERR_CLUSTER_AAANOTSUPP	2470	0x9A6	AAA is not supported in Cluster mode.
NSERR_CLUSTER_IPSECNOTSUPP	2471	0x9A7	IPSEC is not supported in Cluster mode.
NSERR_LINKSET_LAMEMBER	2472	0x9A8	Interface part of a Channel
NSERR_CLIP_LOCALNODE_NOTCONFIGURED	2473	0x9A9	CLIP cannot be added as the local node is not a member of the cluster.
NSERR_CLVERSIONMISMATCH	2474	0x9AA	Cluster version mismatch between Configuration Coordinator and local node
NSERR_CLUSTER_CMNOTSUPP	2475	0x9AB	Connection Mirroring is not supported in Cluster mode
NSERR_ARP_CLAG_OWNERNODE_MANDATORY	2476	0x9AC	Owner Node should be specified for CLAG.
NSERR_ARP_OWNERNODE_MANDATORY	2477	0x9AD	Owner Node should be specified.
NSERR_NOOPCCO	2478	0x9AE	Operation not permitted on the Configuration Coordinator.
NSERR_SPOTTED_CONFIG_PRESENT	2479	0x9AF	Operation not permitted as the node is part of spotted entity configuration.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CLUSTER_NNRES	2481	0x9B1	Error in communication with other nodes in the cluster
NSERR_LAST_CLUSTER_IP_REMOVAL	2482	0x9B2	A NetScaler cluster must have a minimum of one cluster IP address. This is the last available cluster IP address and hence it cannot be deleted.
NSERR_CONFIG_PARAM_INVALID	2483	0x9B3	Configured Param(s) too small for Cluster. Parameter value(s) will be bumped to OVS node count.
NSERR_CLUSTER_INVALIDNODE	2484	0x9B4	Owner node value should be between 0 and 31, both inclusive.
NSERR_LRONCLACPONLY	2485	0x9B5	Link Redundancy can be set only in Cluster LACP Channels
NSERR_MULTIBIND_NOT_SUPPINCLUSTER	2486	0x9B6	Multiple Bindings in Single Bind Command not Supported in Cluster.
NSERR_CLUSTER_FISNAMES_DUPNOTALLOWED	2487	0x9B7	In a cluster setup, the FIS name on each node must be unique.
NSERR_CLUSTER_CLAG_NOLOCALINTF	2488	0x9B8	CLA cannot be bound to the FIS as the CLA does not contain a local interface.
NSERR_FIS_MAX_LIMIT	2489	0x9B9	An appliance can have a maximum of 64 failover interface sets (FIS). This appliance or ownernode (cluster) already has maximum number of allowable FIS.
NSERR_BOUND_TO_FIS	2490	0x9BA	Interface bound to FIS.
NSERR_CHANNEL_OPEN_FAILED	2491	0x9BB	Unable to open channel
NSERR_NOT_BOUND_TO_FIS	2492	0x9BC	Interface not bound to this FIS
NSERR_CLUSTER_INC_NOTSUPP	2493	0x9BD	Feature not supported when cluster is configured in INC mode
NSERR_CONFIG_NG_CHANGED	4095	0xfff	Config NodeGroup changed, force cluster sync should be fired on newly added node to be in sync
NSERR_STRIPED_IP_EXIST	2494	0x9BE	Striped SNIPs/MIPs must be removed when transitioning to cluster-inc
NSERR_WRONG_NG	2495	0x9BF	All cluster nodes should be bound to DEFAULT_NG before transitioning to cluster-inc

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NG_REQUIRED	2496	0x9C0	The nodegroup parameter is mandatory when the -inc parameter is enabled
NSERR_NG_CHANGE	2497	0x9C1	NodeGroup can be changed only for transition from cluster to cluster-inc
NSERR_NG_INVALID	2498	0x9C2	Invalid nodegroup
NSERR_NOT_LAST_NODE	2499	0x9C3	The -flushall parameter can be enabled only when there is one node in the nodegroup
NSERR_MULTI_NG_BINDING_EXIST	2500	0x9C4	More than one Nodegroup has nodes bound to it
NSERR_INVALID_NC_CONFIG	2501	0x9C5	The owner group parameter is applicable only when the INC parameter is enabled on the cluster instance
NSERR_FIPSCL_CANNOT_COEXIST	2502	0x9C6	Clustering and FIPs configuration cannot coexist.
NSERR_NODE_BINDING_INC	2503	0x9C7	Node cannot be bound to Default nodeGroup on cluster-inc setup

© 1999-2017 Citrix Systems, Inc. All rights reserved.

CR ERRORS

May 20, 2016

CR ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PXY_CACHE_HMG	512	0x200	All caches in a content group should be of the same type
NSERR_PXY_RM_LAST_MEMT	513	0x201	Cannot remove last map entry from the table; remove the map table instead
NSERR_CSW_INS_INVAL_PFX	514	0x202	Invalid prefix specified in the URL
NSERR_PENG_EXPR_IVAL_NAME	515	0x203	Invalid expression name
NSERR_CSW_BIG_URL	516	0x204	URL specified is too long - maximum is 208, including . and *
NSERR_CSW_BIG_PFX	517	0x205	Prefix specified is too long - maximum is 199, excluding *
NSERR_CSW_BIG_SFX	518	0x206	Extension specified is too long, maximum is 8
NSERR_CSW_INVAL_SFX	519	0x207	Extension is invalid
NSERR_EXPR_NOMETHOD	520	0x208	Invalid method name
NSERR_EXPR_NOURLTOKENS	521	0x209	Invalid HTTP URL tokens
NSERR_EXPR_NOVERSION	522	0x20A	Invalid HTTP version
NSERR_EXPR_NOHDR	523	0x20B	Invalid HTTP header
NSERR_EXPR_NOCACNTL	524	0x20C	Invalid cache-control value
NSERR_EXPR_NOPRAG	525	0x20D	Invalid pragma value
NSERR_EXPR_NOQUERY	526	0x20E	Invalid query string

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ACTION_INUSE	528	0x210	Action name is already in use
NSERR_ACTION_HDR_INVAL	529	0x211	Invalid header
NSERR_URLQ_INVAL	530	0x212	?
NSERR_UNDEF_ACTION	531	0x213	Action directive or qualifier is not valid
NSERR_CPE_INUSE	532	0x214	Policy name is already in use
NSERR_CPE_REQACT_INVAL	533	0x215	Request action is not valid
NSERR_CPE_RSPACT_INVAL	534	0x216	Response action is not valid
NSERR_CPE_REQRULE_INVAL	535	0x217	Request rule is not valid
NSERR_CPE_RSPRULE_INVAL	536	0x218	Response rule is not valid
NSERR_ACTION_DEFINVAL	537	0x219	Default actions cannot be removed
NSERR_ACTION_NOTPRESENT	538	0x21A	Action does not exist
NSERR_PXY_INVAL_SERVICETYPE	539	0x21B	Invalid service type for virtual server
NSERR_CACHEPOLICY_INUSE	540	0x21C	Caching policy name is already in use
NSERR_CACHEGROUP_INTERNAL	543	0x21F	Built-in content groups cannot be removed
NSERR_CPE_INVAL	544	0x220	Policy name is invalid
NSERR_EXPR_NOLEN	545	0x221	URL length is not valid
NSERR_DNSWAIT	546	0x222	?
NSERR_GW_TIMEOUT	547	0x223	?
NSERR_CSWDMN_INUSE	548	0x224	Domain is already hosted by another server

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ACTION_NOMOD_HDR	550	0x226	Action has missing argument
NSERR_EXPR_INVAL_OPERATOR	551	0x227	Invalid operator for the qualifier
NSERR_EXPR_DEF_REM_INVAL	552	0x228	Default expression cannot be removed
NSERR_EXPR_TOOMANY	553	0x229	Expression limit reached
NSERR_ACTION_TOOMANY	554	0x22A	Action limit reached
NSERR_CSWPOLICY_TOOMANY	555	0x22B	CS policy limit reached
NSERR_CRDPOLICY_TOOMANY	556	0x22C	CR policy limit reached
NSERR_MAPPOLICY_TOOMANY	557	0x22D	Map policy limit reached
NSERR_FILTPOLICY_TOOMANY	558	0x22E	Filter policy limit reached
NSERR_CACHEPOLICY_TOOMANY	559	0x22F	Integrated cache policy limit reached
NSERR_CACHEGROUP_TOOMANY	560	0x230	Integrated cache content group limit reached
NSERR_CACHEPARAM_MEMALLOC_FAILED	561	0x231	Unable to allocate specified amount of memory
NSERR_CACHEGROUP_INUSE	562	0x232	Content group name is already in use
NSERR_CACHEGROUP_EXPCONFLICT	563	0x233	Expiry times conflict
NSERR_CACHEPARAM_INVAL	564	0x234	Cache parameter is invalid
NSERR_CACHEGROUP_PARAM_INVAL	565	0x235	Invalid invalidation parameter
NSERR_CACHEGROUP_QUERY_INVAL	566	0x236	Invalid query string
NSERR_ACTION_INVAL	567	0x237	Action name is invalid
NSERR_EXPR_DEF_SET_INVAL	568	0x238	Default expression cannot be set

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CACHEGROUP_RESSZ_MIN_GT_MAX	569	0x239	maximum response size
NSERR_FILTACION_INVALRESPCODE	570	0x23A	Invalid HTTP response code
NSERR_SVPN_CERT_CHALLENGE	571	0x23B	Destination Server challenged for a client certificate
NSERR_CACHEGROUP_HOST_REQ	576	0x240	Host required for a group with invalidation restricted to host
NSERR_CACHEGROUP_HOST_NREQ	577	0x241	Host not required
NSERR_CACHEGROUP_DYNGRP_NEXP	578	0x242	Cannot expire a parameterized content group
NSERR_CACHEGROUP_ONE_GRP_REQ	579	0x243	Only one content group required
NSERR_CACHEGROUP_ONE_ALL_REQ	580	0x244	ALL should be the only content group specified
NSERR_CACHEGROUP_MATCH_PARAM_INVAL	581	0x245	Invalid request hit parameter
NSERR_CACHEGROUP_DYN_RES_CACHE	582	0x246	Cannot specify a parameterized group with a response time CACHE action
NSERR_EXPR_INVAL_VALUE	583	0x247	Invalid expression value
NSERR_CACHEGROUP_PREFETCH_CONFLICT	584	0x248	Prefetch periods conflict
NSERR_CACHEGROUP_PREFETCH_ENABLE	585	0x249	Enable prefetch to set prefetch period
NSERR_CACHEGROUP_CCH_UNKNOWN	586	0x24A	Unknown cache-control header
NSERR_CACHEGROUP_RELEXP_X10MS	587	0x24B	Relative expiry milliseconds must be multiples of 10
NSERR_CACHEGROUP_PREFETCH_X10MS	588	0x24C	Prefetch period milliseconds must be multiples of 10
NSERR_CACHEGROUP_STATIC_TO_DYNAMIC	589	0x24D	This content group cannot be made parameterized
NSERR_PLCY_DEF_REM_INVAL	591	0x24F	Default policy cannot be removed
NSERR_CACHEGROUP_PREFETCH_REL_NREQ	592	0x250	Relative expiry and prefetch period should both be seconds or milliseconds
NSERR_EXPR_SET_INVAL_FLOWTYPE	593	0x251	Flow type cannot be changed

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_EXPR_TOO_BIG	594	0x252	Expression too long - maximum length is 1500, in fully-qualified form
NSERR_CPE_INVALID_IDRANGE	595	0x253	Priority should be between 0 and 64000 (inclusive)
NSERR_EXPR_TOO_BIG_EXT	596	0x254	Expression too long - maximum length is 800, in fully-qualified form
NSERR_CACHEGROUP_MATCHCOOKIE_DYN_REQ	597	0x255	Match cookies argument allowed only with parameterized groups
NSERR_CACHEGROUP_MATCHCOOKIE_NREQ	598	0x256	Cannot modify match cookies
NSERR_CACHEFWPXY_PRESENT	599	0x257	Forward proxy is already present
NSERR_CACHE_PREFETCH_REEVAL_NREQ	600	0x258	Cannot configure a content group to both prefetch and evaluate every miss
NSERR_CACHEFWPXY_TOOMANY	601	0x259	Forward proxy limit reached
NSERR_CACHE_PET_FC_NREQ	602	0x25A	Cannot enable both 'Poll Every Time' and 'Flash Cache'
NSERR_SAVECONFIG_IN_PROGRESS	603	0x25B	Configuration is being saved, please try again later
NSERR_GWSUBNET_NOT_EXIST	604	0x25C	The gateway is not directly reachable
NSERR_GW_REQ_SUBNET	605	0x25D	An existing route relies on the presence of this subnet
NSERR_URLPOL_NO_PRI	606	0x25E	Priority cannot be specified for URL-based content switching policy
NSERR_BAD_CR_ATTRIBS	607	0x25F	Bad cache redirection VIP attributes given
NSERR_MAC_NOT_SUPPORTED	608	0x260	MAC redirection not supported
NSERR_POLICY_NOT_SUPPORTED	609	0x261	Policy redirection not supported with NNTP
NSERR_CACHEABILITY_NOT_SUPPORTED	610	0x262	Cannot set cacheability on cache
NSERR_HOST_RT_NOT_ALLOWED	611	0x263	Host route advertisement not permitted for this IP
NSERR_ROUTING_NOT_ALLOWED	612	0x264	Routing protocols can run only on SNIP's or NSIP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_DIS_ROUTING_NSIP	614	0x266	Cannot disable dynamic routing on NSIP
NSERR_EXPRMISMATCH	624	0x270	?
NSERR_NO_HOST	627	0x273	No host header and default not set
NSERR_NO_DFLT	628	0x274	Host header present, no match, and default not set
NSERR_DFLTDMN_FIRST	629	0x275	No default domain translation entry
NSERR_PXY_CONF_LOOP	630	0x276	Source and target domains cannot be the same
NSERR_PXY_ME_DUP	632	0x278	Duplicate map entry
NSERR_PXY_IVAL_TGT	633	0x279	Target is invalid during map entry addition
NSERR_CONNECTED	634	0x27A	Proxy connection established
NSERR_AUTHENTICATE	635	0x27B	Authentication required
NSERR_LARGE_DOMAIN	636	0x27C	Domain length overflow
NSERR_PXY_IVAL_URL	640	0x280	Source/destination URL or the combination is not valid
NSERR_PXY_ME_USE	641	0x281	Map entry is already in use
NSERR_PXY_MT_TYPE	642	0x282	Map entries in a table should be homogeneous
NSERR_PXY_MT_USE	643	0x283	Map table is in use
NSERR_PXY_MB_INVAL	644	0x284	Invalid map bind operation
NSERR_PXY_MB_USE	645	0x285	You can bind only one table to a vserver
NSERR_PXY_SETDCDN	646	0x286	Invalid DCDN configuration
NSERR_PXY_SETDFLT	647	0x287	Invalid default domain specification

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PXY_FWD_IVAL	649	0x289	Invalid FWD PXY map entry
NSERR_PXY_MTX_PROT	656	0x290	PXY and cache protocol should be the same
NSERR_PXY_MTX_TRA	657	0x291	Invalid cache type on service
NSERR_PXY_MTX_FWD	658	0x292	Forward Pxy matrix
NSERR_PXY_MTX_REV	659	0x293	Reverse Pxy Matrix
NSERR_PXY_ADD_TRA_NONHTTP	660	0x294	transparent non-HTTP policy is invalid
NSERR_PXY_ADD_TRA_OTHER	661	0x295	transparent non-HTTP map/via should not be specified
NSERR_PXY_OPT_INVAL	662	0x296	ARP/ghost options invalid for REV/FWD proxy
NSERR_PXY_ADD_SVR_OTHER	663	0x297	Simple content based no proxy attribs
NSERR_PXY_MB_DUP	664	0x298	Bind entry already exists
NSERR_PXY_DFLT_DUP	665	0x299	Default already exists
NSERR_SEL_INUSE	672	0x2a0	Selector already exists
NSERR_SEL_PARSE_FAILED	673	0x2a1	Error in selector expression syntax
NSERR_NOSELECTOR	674	0x2a2	No such selector
NSERR_SEL_TOOMANY	675	0x2a3	Selector limit reached
NSERR_CACHEGROUP_NOSELPARAM	677	0x2a5	Cannot change param-based content group to selector-based (or vice-versa)
NSERR_NOCACHEGROUP	678	0x2a6	Content group does not exist
NSERR_NONREQ_SEL	679	0x2a7	Selector expression is not request based
NSERR_INVALARG	680	0x2a8	The action qualifier and the specified argument(s) do not match

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NOWILD_ALLOWED	681	0x2a9	Wildcard not allowed with BETWEEN operator
NSERR_DATE_INCOMPAT	682	0x2aa	Time arguments incompatible
NSERR_TIMEDATE_INVALID	683	0x2ab	Date/time value invalid
NSERR_CONTENTGROUP_COOKIE_REQ_PARAM	684	0x2ac	Hit parameters or inval parameters needed to enable cookie matching
NSERR_CONTENTGROUP_IGNORECASE_REQ_HITPARAM	685	0x2ad	Hit parameters needed to enable case-insensitive param value matching
NSERR_CONTENTGROUP_INVALPARAM_REQ	686	0x2ae	Inval parameters needed to enable host-based invalidation
NSERR_CLISEC_EXP_TOO_LONG	687	0x2af	Client security expression too long
NSERR_NONHTTP_CSW_BIND_HTTP_SSL_POLICY	688	0x2b0	HTTP/SSL protocol based policies can be bound only to HTTP/SSL CS vserver
NSERR_NONHTTP_CSW_BIND_DOMAIN_POLICY	689	0x2b1	Domain based policies can be bound only to HTTP/SSL CS vserver
NSERR_NONHTTP_CSW_BIND_URL_POLICY	690	0x2b2	URL based policies can be bound only to HTTP/SSL CS vserver
NSERR_POLBOUNDTO_TOO_MANY_VSVRS	691	0x2b3	Domain cannot be set - policy bound to more than one vserver.
NSERR_PI_CSW_URL_DOMAIN	692	0x2b4	Domain/URL options disallowed along with advanced expressions
NSERR_PI_TO_PE_CSW	693	0x2b5	Advanced expressions based policies cannot be bound to a classic CS vserver.
NSERR_PE_TO_PI_CSW	694	0x2b6	Classic expression policies cannot be bound to a advanced CS vserver.
NSERR_PRIORITY_COMP_PI_CSW	695	0x2b7	Priority is mandatory for advanced expressions.
NSERR_PI_TCPCSW	696	0x2b8	Advanced expressions can not be bound to non-HTTP CS Vserver.
NSERR_PE_GOTO	697	0x2b9	Priority Jump expression cannot be bound with a classic expression based policy.
NSERR_PI_TO_PE_POL_CSW	698	0x2ba	Advanced expression based policy cannot be converted to a classic expression based policy and vice versa.
NSERR_CACHEOBJECT_EVICT	699	0x2bb	Cached object removed on expiry.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_DNSFAIL	700	0x2bc	DNS request failed
NSERR_HC_RET_TYPE_CHANGE	701	0x2bd	Cannot change http callout return type.
NSERR_HC_NOT_HTTP_VS	702	0x2be	Not a HTTP or SSL vserver.
NSERR_HC_REQ_CONFIG_XOR	703	0x2bf	Full request expression and other request attributes cannot be set at the same time.
NSERR_HC_SERVICE_CONFIG_XOR	704	0x2c0	
NSERR_HC_RET_TYPE_REQD	705	0x2c1	Return type needs to be configured first.
NSERR_RTSP_CSW_BIND_IP_POLICY	706	0x2c2	Only IP based policies can be bound to RTSP CS vserver
NSERR_CSW_BIND_INCOMPAT_TGT	707	0x2c3	The target vserver is not compatible with the CS verver.
NSERR_CSW_BIND_INCOMPAT_CSPL	725	0x2d5	The target vserver is not compatible with the CS policylabel.
NSERR_AS_BAD_XMLNAMESPACE_PREFIX	708	0x2c4	Invalid XML-Namespace prefix.
NSERR_INVALID_SIP_EXPR	709	0x2c5	The SourceIP Expression specified is invalid.
NSERR_NO_BACKENDVSERVER	710	0x2c6	No backend LB vserver found.
NSERR_CONTENT_GROUP_TOOMANY	711	0x2c7	ContentGroup limit reached
NSERR_CACHE_MEM_SIZE_CHANGED	712	0x2c8	To use new Integrated Cache memory limit, save the configuration and restart the NetScaler.
NSERR_CACHE_MEM_SIZE_ZERO	713	0x2c9	No memory limit is configured for Integrated Cache. Use set cache parameter command to set the memory limit.
NSERR_L2CONN_NOT_ALLOWED	714	0x2ca	l2conn feature is supported only for ncore.
NSERR_SQL_NOT_ALLOWED	715	0x2cb	Database feature is supported only on nCore
NSERR_IPSEC_NOT_ALLOWED	716	0x2cc	IPSec feature is not supported
NSERR_HTMLPAGE_NOT_ALLOWED	717	0x2cd	Respondwith/sqlresponse_ok/sqlresponse_error actions cannot be set to respondwithhtmlpage action, and vice versa.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PE_NOT_ALLOWED_DB	718	0x2ce	PE expressions are not allowed for DataStream.
NSERR_CSW_NO_POLICY_NAME	719	0x2cf	Policy name is required with priority.
NSERR_SQL_MSG_EXCEED	720	0x2d0	Message length should be smaller than 512
NSERR_CSW_CLASSIC_APPFW_POL	721	0x2d1	Binding classic appfw policy to CS vserver is not supported.
NSERR_ROUTING_STRIPED_SNIP	722	0x2d2	Routing protocols can run only on a spotted SNIP in a cluster
NSERR_USERNAME_MISSING	723	0x2d3	UserName argument Missing
NSERR_ACTION_EXIST	724	0x2d4	Default Action for this type already exist.
NSERR_WRONG_SERVICENAME	726	0x2d6	Use corresponding 'gslb service' command.
NSERR_PE_NOT_ALLOWED_RULE	727	0x2d7	PE expressions are not allowed for RULE based persistence
NSERR_NODE_NOT_IN_CLUSTER	728	0x2d8	Node is not in the cluster config
NSERR_MCMX_CNRES	729	0x2d9	Error in communication with other cores
NSERR_HC_POST_REQ_CONFIG_XOR	730	0x2da	Parameters and body cannot be set at the same time for the POST method
NSERR_HC_CACGRP	731	0X2db	Operation on 'calloutContentGroup' content group not permitted
NSERR_CSW_BIND_INCOMPAT_BKUP	732	0x2dc	The backup vserver of the target vserver is not compatible with the CS vserver.
NSERR_CACHE_MEM_RESIZE_IN_PROGRESS	733	0x2dd	Integrated Cache memory resize is in progress. Please wait a few seconds.
NSERR_CACHE_MEM_RESIZE_FAILED	734	0x2de	Integrated Cache memory resize failed.
NSERR_CSW_BIND_INCOMPAT_TARGET	735	0x2df	The target vserver has to be an LB vserver.
NSERR_ATTRIB_MEM_FREED	736	0x2e0	ATTRIB memory already freed
NSERR_RULE_MISSING	737	0x2e1	EvalRule argument Missing
NSERR_CACHEGROUP_CCH_MAXAGE_MAX	738	0x2e2	max-age value cannot be greater than 2147483647

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CACHE_MEM_REDUCTION_NOT_ALLOWED	739	0x2e3	Integrated Cache memory limit cannot be reduced.
NSERR_CSW_POL_INCOMPAT_LB	740	0x2e4	The target vserver is not compatible with the CS policy expression.
NSERR_CS_INVALBACKUP	741	0x2e5	Content-Switching or Cache-Redirection vservers cannot be used as a backupvserver.
NSERR_CACHE_MEM_LIMIT_EXCEEDING	742	0x2e6	The IC memory configured for all the admin partitions that are defined on the appliance, must be limited to the IC memory that is configured on the default partition.
NSERR_PI_TO_PE_CRD	743	0x2e7	Advanced expressions based policies cannot be bound to a classic CR vserver.
NSERR_PE_TO_PI_CRD	744	0x2e8	Classic expression policies cannot be bound to a advanced CR vserver.
NSERR_INVAL_ARG_CRPL_LBVS	745	0x2e9	LB vserver cannot be bound to the CR vserver along with a CR policy.
NSERR_CRPEPOLICY_SETNOACTION	746	0x2ea	Cache-Redirection classic policy does not support action
NSERR_CACHE_MEM_LIMIT_NOT_CONFIGURED	747	0x2eb	Set the IC memory on the default partition before setting the IC memory for an admin partition.
NSERR_INVALID_BIND_PIPOL_CS	748	0x2ec	The Advanced Policy cannot be bound to a CS vserver
NSERR_INVALID_BIND_PIPOL_CR	749	0x2ed	The Advanced Policy cannot be bound to a CR vserver
NSERR_MEDIAC_MEMORY_ALLOC_FAILED	750	0x2ee	Memory allocation for media classification failed
NSERR_CRD_CLASSIC_APPFW_POL	751	0x2ef	Binding classic appfw policy to CR vserver is not supported.
NSERR_CSPOL_VPN_BIND	756	0x2f4	Invalid bind, VPN vserver is already set as a target Vserver on a CS action.
NSERR_CSACT_VPN	757	0x2f5	VPN vserver cannot be set as a target Vserver since CS policies are bound to it.
NSERR_CSPOL_ACT_VPN	759	0x2f7	CS policy cannot be bound to the vserver.CS action must be a LB vserver.
NSERR_VPNSERVER_NOTEXIST	760	0x2f8	The VPN Vserver does not exist
NSERR_VPNREFERRED_IN_CSBINDING	761	0x2f9	Cannot remove VPN Vserver which is referred in CS policy bindings.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CSW_ACTION_INCOMPTIBLE	764	0x2fc	Incompatible Content Switching action provided. During set operation the old and the new action should have compatible fields. eg, target LB vserver to target LB vserver.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

CVPN ERRORS

Jul 07, 2017

CVPN ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NO_SUCH_PROFILE	3248	0xcb0	Profile does not exist
NSERR_PROFILE_IN_USE	3249	0xcb1	Profile in use
NSERR_NO_SAMLPROFILE	3383	0xd37	Please specify a SAML SSO profile.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

DSK CACHE ERRORS

Jul 07, 2017

DSK CACHE ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_IC_CACHE_NOT_ENB	752	0x2f0	FEO enabled but IC needs to be enabled for most FEO features to work.
NSERR_DISK_ENGG_NOT_RUNNING	753	0x2f1	Disk Engine is not running, Save the config and reboot.
NSERR_DISK_ENGG_DISABLED	754	0x2f2	To take Disk Engine down, Save the config and reboot.
NSERR_IC_DISK_CACHE_PARAM_NOT_SET	755	0x2f3	Disk Cache not enabled. Set -enableDiskCache parameter to YES.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

EV SUBSCRIBER ERRORS

Jul 07, 2017

EV SUBSCRIBER ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_EV_SUBSCRIBER_EXISTS	3489	0xda1	There exists a subscriber with the same name
NSERR_EV_SUBSCRIBER_NOT_FOUND	3490	0xda2	Subscriber does not exist
NSERR_EV_SUBSCRIBER_MAX_CONFIG	3491	0xda3	Maximum number of subscribers configured already
NSERR_EV_SUBSCRIBER_ALLOC_FAIL	3492	0xda4	Memory allocation failure while adding a subscriber
NSERR_EV_SUBSCRIBER_INVALID_TYPE	3493	0xda5	Invalid subscriber type
NSERR_EV_SUBSCRIBER_INVALID_URI	3494	0xda6	Invalid URI
NSERR_EV_SUBSCRIBER_SINFO_FAIL	3495	0xda7	Failed to create server info for the subscriber
NSERR_EV_SUBSCRIBER_INVALID_EVENT	3496	0xda8	Invalid event type
NSERR_EV_SUBSCRIBER_INVALID_ENTITY	3497	0xda9	Invalid entity type
NSERR_EV_SUBSCRIBER_ALL_TO_SPECIFIC	3498	0xdaa	Trying to (un)bind a specific enitty to an event; ALL entities were bound previously
NSERR_EV_SUBSCRIBER_SPECIFIC_TO_ALL	3499	0xdab	Trying to (un)bind ALL entities to an event; specific entities were bound previously

© 1999-2017 Citrix Systems, Inc. All rights reserved.

FWDSESSION ERRORS

May 20, 2016

FWDSESSION ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_FWDSESSION_NAME_EXISTS	928	0x3a0	A forwarding session with same name exists
NSERR_FWDSESSION_NW_EXISTS	929	0X3a1	A forwarding session with same network exists
NSERR_FWDSESSION_ACL_EXISTS	930	0X3a2	This acl is already configured
NSERR_FWDSESSION_ACL_NOT_EXISTS	931	0x3a3	Specified ACL is not configured
NSERR_FWDSESSION_CONN_FAILOVER	932	0x3a4	Connection fail-over not supported for v6 forwarding sessions

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GENTOOL ERRORS

Jul 07, 2017

GENTOOL ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SSL_CONFFILE	1601	0x641	Error in SSL conf file
NSERR_SSL_NOCONFFILE	1602	0x642	Unable to load SSL configuration info
NSERR_SSL_SIGFAIL	1603	0x643	Signing operation failed
NSERR_SSL_INVALFORMAT	1604	0x644	Invalid format
NSERR_SSL_OUTFILE	1605	0x645	Problem in writing output file
NSERR_SSL_VERIFY_FAIL	1606	0x646	Signature verification failed
NSERR_SSL_FILECREATE	1607	0x647	Unable to create output file
NSERR_SSL_MINKEYSIZE	1608	0x648	Key size less than 1024 is not supported on FIPS Firmware Version 1.1.
NSERR_SSL_INVALID_REQ	1609	0x649	Invalid certificate request
NSERR_SSL_GENTOOL	1610	0x64A	Internal failure in SSL cert/key generation tool
NSERR_SSL_PEM_ONLY	1611	0x64B	Only PEM format private key can be encrypted
NSERR_SSL_PASSMISMATCH	1612	0x64C	Password verification failed
NSERR_SSL_PASSREQ	1613	0x64D	Password required for private key
NSERR_SSL_INVALPASS	1614	0x64E	Invalid password

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code (Decimal)	Error Code (Hex)	Error Message
NSERR_SSL_GNTOOLARG_MISSING	1616	0x650	Required argument missing
NSERR_SSL_INVAL_PUBEXP	1617	0x651	Invalid public exponent value
NSERR_SSL_INVALID_VALUE	1618	0x652	Invalid value
NSERR_SSL_MISSING_KVAL	1619	0x653	Value missing for the argument
NSERR_SSL_INVAL_DHGEN	1620	0x654	Invalid DH generator value
NSERR_SSL_NOFILE_CERTREQ	1621	0x655	No such certificate request file exists
NSERR_SSL_REQ_NOMATCH	1622	0x656	Certificate Request (CSR) and private key do not match
NSERR_SSL_PKCS12	1623	0x657	Problem in input PKCS12 file
NSERR_SSL_NOFILE_CAPVTKEY	1624	0x658	No such CA key file present
NSERR_SSL_NOFILE_CACERT	1625	0x659	No such CA certificate file exists
NSERR_SSL_NOFILE_SERIAL	1626	0x65A	No such serial file exists
NSERR_SSL_NOFILE_PKCS12	1627	0x65B	No such PKCS12 file exists
NSERR_SSL_READ_CACERT	1628	0x65C	In reading input CA certficate file
NSERR_READ_CERT	1629	0x65D	Error in reading input certficate file
NSERR_IMP_EXP_NOT_TOGETHER	1630	0x65E	Import and export can not be specified together
NSERR_NOFILE_CERT	1631	0x65F	No such certificate file exists
NSERR_NOFILE_KEY	1632	0x660	No such key file exists
NSERR_CACERTPKEY_MISMATCH	1633	0x661	CA certificate not matching with CA private-key

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_WRONG_REV_DATE	1635	0x663	Invalid revocation data in CA's database
NSERR_WRONG_SERIAL_NO	1636	0x664	Invalid serial number in CA's database
NSERR_CREATE_DATABASE_ENTRY	1637	0x665	Revocation entry creation failed
NSERR_ALREADY_REVKD	1638	0x666	Certificate already revoked
NSERR_LOAD_PKEY	1639	0x667	Private key loading failed
NSERR_CERTPKEY_NEEDED	1640	0x668	Both certificate and private key are needed for export
NSERR_PKCS12_NEEDED	1641	0x669	PKCS12 file missing for import
NSERR_ERR_FILEEXISTS	1642	0x66A	Cannot create output file. File already exists
NSERR_FIPSKEY_KEYFILE_OPTION	1643	0x66B	Both -keyFileName and -fipsKeyName cannot be specified
NSERR_OUTFILELENBIG	1644	0x66C	Output file name too long
NSERR_SSL_DEFPATH	1645	0x66D	Output filepath should be under the partition's default SSL directory
NSERR_SSL_CRL_DEFPATH	1646	0x66E	Output CRL filepath should be under the partition's default CRL directory
NSERR_SSL_DEFLOCATION	1647	0x66F	Input file(s) not present or not accessible in current partition
NSERR_DHREQD	1791	0x6FF	DH required
NSERR_SSL_CRL_DEFLOCATION	1650	0x672	Input CRL file not present or not accessible in current partition
NSERR_SSL_NOFIPS_KEY	1651	0x673	No such FIPS key
NSERR_SSL_IMPROPER_SERIALFILE	1652	0x674	Invalid serial file
NSERR_SSL_SERIAL_NUM_REVOKED	1653	0x675	Certificate serial number match with another certificate

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SSL_MAXRSAKEYSIZE	1654	0x676	Maximum allowed key size is 4096
NSERR_SSL_MUSTBEPEM	1655	0x677	The input file must be in PEM or SIM format.
NSERR_SSL_CANTBEPEM	1656	0x678	The input file cannot be in PEM format.
NSERR_SSL_BAD_LABEL	1657	0x679	The keylabel cannot be 'public' or 'private'.
NSERR_SSL_HSMKEY_LOAD	1664	0x680	Unable to Load Key in external Hardware Security Module (HSM).
NSERR_SSL_HSMKEY_PPE	1665	0x681	Internal error while adding HSM key.
NSERR_ERR_SSL_HSMKEY_EXISTS	1666	0x682	HSM Key already exists.
NSERR_ERR_SSL_HSMKEY_DELETE	1667	0x683	Unable to Delete Key in external Hardware Security Module (HSM).
NSERR_ERR_SSL_HSMKEY_NOENT	1668	0x684	No such resource.
NSERR_ERR_SSL_HSMKEY_REFEXT	1669	0x685	The HSM key is referenced by a certificate
NSERR_ERR_SSL_HSMKEY_NOTEXISTS	1670	0x686	No such HSM key.
NSERR_ERR_SSL_HSMKEY	1671	0x687	HSM Key Unknown Error, Probably debugging.
NSERR_ERR_SSL_GW_NOTUP	1672	0x688	Gateway daemon is booting up please wait for 20 to 30 secs.
NSERR_ERR_SSL_HSMKEY_FIPS	1673	0x689	Operation not permitted on a MPX-FIPS System.
NSERR_ERR_SSL_SANCERT_DUP	1674	0x68A	Duplicate domain cert cannot be added.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GSLB ERRORS

Jul 07, 2017

GSLB ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_BAD_DNS_OPTION	1792	0x700	Invalid DNS option specified
NSERR_DNS_BAD_CACHETYPE	1793	0x701	Only forward cache type can bind to DNS vserver
NSERR_DNS_NOTAVAIL	1794	0x702	No such DNS vserver exists
NSERR_DNS_BAD_VSTYPE	1795	0x703	DNS vserver can only be address-based
NSERR_GSLB_SITELKUP_FAILED	1796	0x704	GSLB site not found
NSERR_GSLBBOUNDTO_CSACT	1797	0x705	The GSLB Vserver referred by Content Switching action cannot be removed
NSERR_GSLBREFFERED_IN_CSBINDING	1798	0x706	Cannot remove GSLB Vserver which is reffered in CS policy bindings
NSERR_NOAREC	1801	0x709	Address record for host name not available
NSERR_INVALTTL	1806	0x70E	Invalid TTL value
NSERR_CNAMEEXISTS	1807	0x70F	Alias name record exists for the host name
NSERR_INVAL_SVCOPTION	1808	0x710	Option not applicable for this service
NSERR_GSLBBIND_EXISTS	1809	0x711	GSLB binding exists for the given host
NSERR_NOT_LOCALREMOTE	1810	0x712	Site information must be specified for a GSLB service

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_COUNT_MISMATCH	1811	0x713	match the number of services bound to the GSLB vserver
NSERR_IP_MISMATCH	1812	0x714	IP addresses of bound GSLB services and the address records do not match
NSERR_SVCIP_REPEAT	1813	0x715	A GSLB service with the same IP address is already bound to the GSLB vserver
NSERR_NOT_REMOTE	1814	0x716	Service bound to GSLB VIP with -gslb remote is a local service
NSERR_NOT_LOCAL	1815	0x717	Service bound to GSLB VIP with -gslb local is not a local service
NSERR_PROXY_REC	1816	0x718	A proxy record cannot be assigned to the domain
NSERR_NOPROXY_AREC	1817	0x719	Proxy address records cannot be bound to a GSLB vserver
NSERR_SVC_GSLBBIND_EXISTS	1819	0x71B	GSLB binding exists - unbind the GSLB vserver first
NSERR_SVCTYPEMISMATCH	1820	0x71C	Service type mismatch with existing service/vserver
NSERR_INVALID_PUBIP_OPTION	1822	0x71E	The public IP and public port are valid only for GSLB local service
NSERR_INVAL_NAME_SYNTAX	1823	0x71F	Invalid domain name syntax
NSERR_GSLB_BACKUP	1824	0x720	Backup VIP is backup of other GSLB VIP. Please unset the existing association and try again
NSERR_IS_BACKUP	1825	0x721	Backup VIP is backup of other VIP. Please unset the existing association and try again
NSERR_GSLB_HAS_BACKUP	1826	0x722	GSLB VIP has a backup. Please unset the backup and try the operation again
NSERR_GSLB_REQUIRES_INDEPENDENT_BACKUP	1827	0x723	Backup VIP to GSLB VIP must not be backup of any other VIP. Please unset the backup and try the operation again
NSERR_GSLB_NOQUALIFIER	1828	0x724	At least one qualifier should be given for the location

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_GSLB_NONAME	1829	0x725	The qualifier does not exist in the repository
NSERR_GSLB_CUSTOM_NOSPACE	1830	0x726	Too many custom locations
NSERR_GSLB_STATIC_NOSPACE	1836	0x72C	Too many static locations
NSERR_GSLBSITENAME_EXISTS	1831	0x727	A GSLB site with the same name exists
NSERR_GSLBSITEIP_EXISTS	1832	0x728	A GSLB site with the same IP address exists
NSERR_NO_GSLBSITE	1833	0x729	The GSLB site does not exist
NSERR_LOCAL_EXISTS	1834	0x72A	The local site already exists
NSERR_NO_GSLBSVC	1835	0x72B	The GSLB service does not exist
NSERR_NO_GSLBVIP	1837	0x72D	The GSLB vserver does not exist
NSERR_NO_DMN	1839	0x72F	The domain does not exist
NSERR_DMN_NOTBOUND	1840	0x730	The domain is not bound to this VIP
NSERR_GSLBDOMAIN_BOUND	1842	0x732	The domain is already bound to a GSLB vserver
NSERR_NOT_GSLBENT	1844	0x734	Operation not permitted on a non-GSLB entity
NSERR_GSLBDOMAIN_PERM	1845	0x735	Operation not permitted on a GSLB-configured domain
NSERR_GSLB_MAXQUAL	1846	0x736	Maximum qualifiers allowed is six
NSERR_GSLB_MAXLOCLENGTH	1847	0x737	Maximum allowed location length is 256 characters
NSERR_GSLBVIP_PERM	1848	0x738	Operation not permitted on a GSLB vserver
NSERR_GSLBSVC_PERM	1849	0x739	Operation not permitted on a GSLB service

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_GSLB_QUALTOOLONG	1851	0x73B	The qualifier is too long
NSERR_GSLB_PERSISTIDEXISTS	1852	0x73C	The persistence ID is already being used by a GSLB vserver
NSERR_GSLB_INVLD_LDNSTO_INTVL	1854	0x73E	Minimum LDNS entry timeout interval is 30 secs
NSERR_VIP_BACKUP_ISGSLB	1856	0x740	GSLB vserver cannot be backup of a non-GSLB vserver
NSERR_GSLB_PUBIP_PUBPORT_EXISTS	1857	0x741	The public IP and public port are being used by another GSLB service
NSERR_GSLBLOCALSVC_ENADIS_NOTALLOWED	1858	0x742	GSLB local service status not changed - please enable or disable the corresponding LB vserver instead
NSERR_GSLBLOCALSVR_ENADIS_NOTALLOWED	1859	0x743	Server status not changed - the server has a GSLB local service
NSERR_INVALID_PERSISTID	1860	0x744	Invalid persistence ID
NSERR_PERM_PERSISTID	1861	0x745	The persistence ID is required for persistence or spillover-persistency to be set
NSERR_INVALID_LOCALSITEIP	1862	0x746	Invalid GSLB local site IP address
NSERR_INVALID_REMOTESITEIP	1863	0x747	Invalid GSLB remote site IP address
NSERR_INVALID_SITETYPE	1864	0x748	Invalid GSLB site type
NSERR_INVALID_BACKUPIP	1865	0x749	Invalid backup IP for the GSLB domain
NSERR_GSLBVIP_MAXSVC	1866	0x74A	Cannot bind more than 32 services to a GSLB vserver
NSERR_INVALID_JITTER	1867	0x74B	RTT tolerance value must be between 1 and 100
NSERR_GSLBLOCSVC_DIS_NOTALLOWD	1868	0x74C	Cannot add GSLB local service with state set to DISABLED

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			address and port must be of type GSLB
NSERR_GSLBSVC_BIND_NOTALLOWD	1870	0x74E	Cannot bind GSLB service to a vserver with valid IP and port
NSERR_NSMAP_VERSION	1871	0x74F	Unrecognized NetScaler file version
NSERR_NSMAP_FORMAT	1872	0x750	Unrecognized file format specified
NSERR_NSMAP_EOF	1873	0x751	End of file has been reached
NSERR_NSMAP_PARSE	1874	0x752	File parsing error
NSERR_NSMAP_READ	1875	0x753	Error reading file
NSERR_NSMAP_WRITE	1876	0x754	Error writing file
NSERR_NSMAP_IOCTL	1877	0x755	Error occurred while transferring information to NetScaler
NSERR_NSMAP_OUTPUTFILE	1878	0x756	Error opening output file for writing
NSERR_NSMAP_INPUTFILE	1879	0x757	Error opening input file for reading
NSERR_NSMAP_REQUIRED_FIELD	1880	0x758	Required field missing
NSERR_NSMAP_DBFILE	1881	0x759	Error opening database file
NSERR_NSMAP_DBINSERT	1882	0x75A	Error inserting entry into the database
NSERR_NSMAP_DBSEARCH	1883	0x75B	Error searching the database
NSERR_GSLBPROXM_LICENCE_ABSENT	1884	0x75C	Cannot set proximity methods without GSLB proximity license
NSERR_GSLB_SITEPERSISTENCE_HTTPONLY	1885	0x75D	Site persistence is valid only for HTTP or SSL GSLB services
NSERR_GSLB_NOSITEPFX	1886	0x75E	Site prefix compulsory if site persistence is HTTPRedirect

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_GSLB_SITEDOMAIN_SIZE	1889	0x761	Site domain cannot be created - name length exceeds maximum
NSERR_GSLB_SITEDOMAIN_SYNTAX	1890	0x762	Site domain cannot be created - name is invalid
NSERR_GSLB_SITEDOMAIN_REFERS	1891	0x763	Site domain cannot be deleted - it has external DNS references
NSERR_GSLBDOMAIN_REFERS	1892	0x764	GSLB domain cannot be deleted - it has external DNS references
NSERR_GSLB_ARECORD_EXISTS	1893	0x765	GSLB record cannot be created - another A record is configured with the same name
NSERR_GSLB_SITECOOKIE_TIMEOUT_RANGE	1894	0x766	Invalid GSLB site cookie timeout
NSERR_GSLBDOMAIN_SYNTAX	1895	0x767	GSLB domain cannot be created - name is invalid
NSERR_GSLB_SO_PERM	1896	0x768	Spill-over cannot be set on a GSLB vserver
NSERR_GSLB_DEPRCOOKIETOUT	1897	0x769	Ignored deprecated cookietimeout setting since it has been made a per-domain configuration
NSERR_DNS_NAMESVR_SYNTAX	1898	0x76a	Invalid name server syntax
NSERR_DNS_ORIGSVR_SYNTAX	1899	0x76b	Invalid origin server syntax
NSERR_DNS_CONTACT_SYNTAX	1900	0x76c	Invalid contact name syntax
NSERR_DNS_MX_SYNTAX	1901	0x76d	Invalid mail exchange syntax
NSERR_DNS_CNAME_SYNTAX	1902	0x76e	Invalid canonical name syntax
NSERR_DNS_ALIAS_SYNTAX	1904	0x770	Invalid alias name syntax
NSERR_GSLB_NOLOCALVIP	1905	0x771	Cannot create GSLB local service - add the local vserver first
NSERR_GSLB_LOCALSVCEXISTS	1906	0x772	Cannot delete the vserver - corresponding GSLB local service still exists

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_GSLBLOCALSVC_PERM	1907	0x773	local service
NSERR_DNS_INVAL_REVDOMNAME_SYNTAX	1908	0x774	Invalid reverse domain name syntax
NSERR_DNS_ALIASREC	1909	0x775	Alias name cannot have any record
NSERR_GSLBLOCSVC_DELAYEDCLEAN_NOTALLOWD	1910	0x776	Down state flush cannot be set on GSLB local service
NSERR_NO_BKP_VIP	1911	0x777	Set a backup Vserver, to enable it.
NSERR_GSLB_NOSUCH_LDNSENTRY	1912	0x778	LDNS entry not present
NSERR_MAX_DNS_VIEW	1913	0x779	Number of DNS views exceeds limit
NSERR_GSLB_SITE_IP_EXISTS	1914	0x77a	GSLB Site IP already exists
NSERR_GSLB_LAST_MIP	1915	0x77b	MIP is not removed. At least one MIP is required
NSERR_RM_GSLB_SITE	1916	0x77c	IP cannot be removed as it is being used by the GSLB local site
NSERR_GSLB_IPRM_LAST_MIP	1917	0x77d	GSLB Site IP removed. MIP is not removed. At least one MIP is required
NSERR_GSLB_SVC_PUB_PORT_ERR	1918	0x77e	Cannot modify public port for remote GSLB service
NSERR_REC_HAS_REF	1919	0x77f	DNS record has references
NSERR_DNS_VIEW_REF	1920	0x780	The dns view cannot be removed as it has dependencies
NSERR_GSLB_SITE_PERSISTENCE_MATCH	1921	0x781	Mismatch in gslb site persistence type with other bound services
NSERR_GSLB_SITE_PERSISTENCE_CONFLICTS	1922	0x782	Site persistence conflicts with other services bound to the gslb vip(s)
NSERR_DNS_POLICY_INVAL	1923	0x783	Invalid DNS policy
NSERR_GSLBVIP_CNAME_MISMATCH	1924	0x784	GSLB vserver cannot be bound with cname based and ip based

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_GSLBVIP_CNAME_UNSUPPORTED_LBMETHODS	1925	0x785	GSLB vserver having cname services cannot have dynamic load balancing methods
NSERR_GSLB_SERVICE_MON_NOIPPORT	1926	0x786	Cannot bind monitor with zero ip or port to a cname gslb service
NSERR_GSLB_SERVICE_MON_SET_NOIPPORT	1927	0x787	Cannot set ip or port to zero when monitor is bound to a cname gslb service.
NSERR_GSLB_CNAME_SERVICE_SET	1928	0x788	Setting of these parameters on cname gslb service is not allowed
NSERR_GSLB_CNAME_VSERVER_SET	1929	0x789	Setting of these parameters on gslb vserver having cname services is not allowed
NSERR_GSLBVIP_CNAME_UNSUPPORTED_BKUP_LBMETHODS	1930	0x78a	GSLB vserver having cname services cannot have dynamic backup load balancing methods
NSERR_GSLBVIP_CNAME_UNSUPPORTED_EDR	1931	0x78b	GSLB vserver having cname services cannot have empty down response disabled
NSERR_GSLBVIP_CNAME_UNSUPPORTED_MIR	1932	0x78c	GSLB vserver having cname services cannot have multiple ip response enabled
NSERR_GSLBSVC_VIEW_NEXIST	1933	0x78d	View is not bound to the GSLB service
NSERR_DNS_TTL_MORE_THAN_MAX_ALLOWED	1934	0x78e	DNS RECORD TTL value greater than max TTL allowed
NSERR_GSLB_LB_MAXSITES	1935	0x78f	Maximum number of LB sites is 1024
NSERR_GSLB_HASCHILDREN	1936	0x790	Cannot remove site that has child site(s)
NSERR_GSLB_PARENTISCHILD	1937	0x791	Parent site cannot be a child to another site
NSERR_MAX_STREAM_SELECTORS	1938	0x792	Number of selectors exceeds limit
NSERR_MAX_STREAM_IDENTIFERS	1939	0x793	Number of identifiers exceeds limit
NSERR_NO_SUCH_SELECTOR	1940	0x794	The given selector does not exist

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_RLT_TIMESLICE_INVALID_VAL	1942	0x796	Time slice should be a multiple of 10
NSERR_ILLEGAL_SUBNET_MASK	1943	0x797	Incorrect subnet mask value
NSERR_STREAM_SELECTOR_INUSE	1944	0x798	The selector is being referenced by one or more identifiers
NSERR_IDENTIFIER_INUSE	1945	0x799	The identifier is being referenced by one or more policies
NSERR_SELECTOR_CANNOT_CHANGE_ATTRIB_TYPE	1946	0x79A	Cannot change request type selector to response type
NSERR_GSLB_IGN_TRIGMON	1947	0x79B	Trigger monitor setting assumed ALWAYS for GSLB local site
NSERR_SELECTOR_NOT_MORE_THAN_2_IPV6_EXP	1948	0x79C	Cannot exceed more than two IPV6 expressions in a selector
NSERR_REPEATED_MONITORS	1949	0x79D	Monitors cannot be repeated
NSERR_ALL_MONITORS_DISABLED	1950	0x79E	All monitors in selected list are disabled.
NSERR_GSLB_AAAARECORD_EXISTS	1951	0x79F	GSLB record cannot be created - another AAAA record is configured with the same name
NSERR_GSLBVIP_HETEROGENEOUS_SERVICE_IPVERSION	1952	0x7A0	Heterogeneous service binding is not allowed for GSLB vserver
NSERR_VIEWIP_IPV6	1953	0x7A1	View IP setting is not allowed for IPv6 GSLB service.
NSERR_BACKUPIP_IPV6	1954	0x7A2	IPv4 backup IP setting is not allowed for IPv6 GSLB vserver.
NSERR_BACKUP_VIP_MISMATCH	1955	0x7A3	Both primary and secondary vservers should be of same DNSRecordType
NSERR_GSLB_DOMAIN_CONVERSION	1956	0x7A4	Cannot associate gslb domain to the service.
NSERR_GSLB_IPV4_BACKUPIP	1957	0x7A5	IPv6 service binding is not allowed along with IPv4 backup IP.
			Cannot set a new selector when

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			bound to a policy.
NSERR_SEL_CONFLICTING_ATTRIBUTES	1959	0x7A7	Conflicting attributes in the new and current selector sets.
NSERR_GSLBVIP_CNAME_BACKUPIP	1960	0x7A8	GSLB vserver having cname services cannot have backup IP
NSERR_GSLB_HETEROGENEOUS_SITEIP	1961	0x7A9	Heterogeneous IPv6 and IPv4 GSLB site IPs not allowed.
NSERR_DNS_MAX_AAAA_REC	1962	0x7AA	Cannot exceed more than 5 IPv6 addresses.
NSERR_DNS_MAX_A_REC	1963	0x7AB	Cannot exceed more than 25 IP addresses.
NSERR_GSLB_RECTYPE_SERVICE	1965	0x7AD	Cannot set the record type while services bound to the GSLB vserver.
NSERR_GSLB_RECTYPE_DOMAIN	1966	0x7AE	Cannot set the record type while domains bound to the GSLB vserver.
NSERR_GSLB_RECTYPE_BACKUP_VIP	1967	0x7AF	Cannot set the record type while GSLB vserver has backup vserver.
NSERR_GSLB_BKPVIP_RECTYPE	1968	0x7B0	Cannot set the record type while GSLB vserver is a backup vserver.
NSERR_GSLBVIP_NAPTR_UNSUPPORTED_SVCTYPE	1969	0x7B1	GSLB vserver of NAPTR DNS record type can have only ANY as service type.
NSERR_BACKUPIP_IPV4	1970	0x7B2	IPv6 backup IP setting is not allowed for IPv4 GSLB vserver.
NSERR_GSLB_IPV6_BACKUPIP	1971	0x7B3	IPv4 service binding is not allowed along with IPv6 backup IP.
NSERR_GSLB_DB_REQ_TOO_BIG	1972	0x7B4	Static proximity Database request is too big.
NSERR_GSLB_DB_QUEUE_MAXED	1973	0x7B5	Static proximity Database queue is full.
NSERR_GSLB_DB_SERVER	1974	0x7B6	Static proximity database server is not running.
NSERR_GSLB_DB_TIMEOUT	1975	0x7B7	Static proximity database server is not responding.
NSERR_GSLB_DB_CLOSED	1976	0x7B8	Connection to static proximity

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_GSLB_COORDINATES	1977	0x7B9	Both longitude and latitude should be specified or not.
NSERR_INVALID_IPV6_PREFIXLEN	1978	0x7BA	Invalid IPv6 prefix length.
NSERR_GSLBDOMAIN_CANME_RECORD_EXISTS	1979	0x7BB	Cname record can't be created. Another record exists for the configured domain.
NSERR_GSLBDOMAIN_AAAA_CNAME_EXISTS	1980	0x7BC	Cname record exists for the configured domain.
NSERR_GSLB_NON_PARENT_REMOTESITE	1981	0x7BD	Cannot have a gslb remote site, other than the parent site, configured on an LB node
NSERR_ZONE_EXISTS	1982	0x7BE	Zone already configured.
NSERR_INVAL_KEYFLAGS	1983	0x7BF	Invalid flags value in DNSKEY
NSERR_DNSKEY_NONEXISTS	1984	0x7C0	Zone has no DNSKEY's
NSERR_NORECS_IN_ZONE	1985	0x7C1	No records in zone
NSERR_SIGNFAILED	1986	0x7C2	RSA Sign operation returned error
NSERR_DIGESTINIT_FAILED	1987	0x7C3	Digest init before RSA Sign operation returned error
NSERR_DIGESTUPDATE_FAILED	1988	0x7C4	Digest update before RSA Sign operation returned error
NSERR_KEYEXISTS	1989	0x7C5	The DNSKEY is already added for this zone
NSERR_NOPASVKEYS	1990	0x7C6	No Passive DNSKEY's in zone
NSERR_DNS_MAXKEYSIZE	1991	0x7C7	Maximum allowed public key size is 1024 and private key size is 4096
NSERR_NODATA_TO_SIGN	1992	0x7c8	No data to sign the record
NSERR_CNAME_SITE_NOTEXISTS	1993	0x7c9	Site name must be specified for cname based GSLB service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_DNS_PROXY_ZONE	1995	0x7cB	Cannot sign a proxy zone
NSERR_NOACTVKEYS	1996	0x7CC	No active DNSKEY's in zone
NSERR_NOTIFYPERIOD	1997	0x7cd	Notification period must be less than the expiry period
NSERR_LOAD_PUBKEY	1998	0x7ce	Public key loading failed
NSERR_DNSKEYGEN_UNSUPPORTED_ALGO	1999	0x7cf	Unsupported algorithm
NSERR_DNSKEYGEN_ERR_PUBFILE_OPEN	2000	0x7d0	Error opening public key file
NSERR_DNSKEYGEN_ERR_PRIVFILE_OPEN	2001	0x7d1	Error opening private key file
NSERR_DNSKEYGEN_ERR_DSFILE_OPEN	2002	0x7d2	Error opening delegation signer (DS) file
NSERR_GSLB_OPT_NOTSUPPORTED	2003	0x7d3	gslb option is not supported, GSLB site is mandatory for GSLB services
NSERR_GSLB_GFS_NOT_SUPPORTED	2004	0x7d4	Graceful shutdown is not supported for GSLB services.
NSERR_NSMAP_IMPORTFILE	2005	0x7d5	Error reading the imported file. See /var/log/ns.log file for more details.
NSERR_DNS_CNAMELOOP	2006	0x7d6	Addition of this CNAME record leads to a CNAME loop
NSERR_NEWSVC_NOT_4GSLB	2007	0x7d7	startup settings not applicable to GSLB vservers
NSERR_NON_CLUSTER_SYSTEM	2008	0x7d8	The option is only supported for clustering systems.
NSERR_PM_MON_NOT_ENABLED	2009	0x7d9	Path monitoring must be enabled to enable this option.
NSERR_GSLB_BACKUP_PERS_DEPRICATED	2010	0x7dA	GSLB Backup Persistence is deprecated. Please use Spillover Persistence instead.
NSERR_DNS_SIGNFAIL	2011	0x7dB	DNSSEC Sign operation failed

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_DNSSEC_OFFLOAD_NOSUPPORT	2013	0x7dd	DNSSEC Offload is not supported in this platform.
NSERR_DNS64_ACT_INVAL	2014	0x7dE	Invalid DNS64 action.
NSERR_DNS64_NOT_SUPPORTED_VS	2015	0x7dF	Specified policy can be bound only to DNS/DNS_TCP vserver.
NSERR_DNS64_INVAL_PREFIX	2016	0x7e0	Specified Prefix is either not allowed or Prefix length is not 96.
NSERR_GSLB_DB_REPEAT_REQUEST	2017	0x7e1	Location database request need to be repeated
NSERR_GSLBSITE_CONN_NOTSUPPORTED	2018	0x7e2	This may cause GSLB site MEP connection issues in some earlier releases.
NSERR_DNS64_TIMEOUT_ROUNDOFF	2019	0x7e3	Timeout threshold is rounded off to nearest multiple of 10 less then the given value.
NSERR_USIP_EV_NO_MEM_WARNING	2020	0x7e4	Service marked unusable due to lack of memory to post APBR rules for USIP.
NSERR_USIP_EV_NO_MEM_WARNING_DEL	2021	0x7e5	APBR rule on switch not deleted due to lack of memory to post APBR rules for USIP.
NSERR_USIP_EV_NO_SVC_GRP_MEM_WARNING	2022	0x7e6	One or more APBR rules were not configured on the switch due to lack of memory.
NSERR_USIP_EV_NO_SVC_GRP_MEM_WARNING_DEL	2023	0x7e7	One or more APBR rules were not deleted on the switch due to lack of memory.
NSERR_DNSKEY_MISMATCH	2024	0x7e8	The Key specified is not valid for this zone
NSERR_INVALID_REMOTE_CLIP	2025	0x7e9	Invalid remote GSLB site CLIP. CLIP belong to the local gslb site
NSERR_IPV6_CLIP_NOTSUPPORTED	2026	0x7ea	IPV6 CLIP is not supported
NSERR_NON_DEFAULT_DBFILE_LOC	2027	0x7eb	In case of HA setup and cluster system, the command might fail on other nodes if an identical copy of the file is not present at the same

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_DNS64_FROM_TO_TYPE	2028	0x7ec	IP From and IP To in location specification are of different IP type.
NSERR_DNS_NOSOA_NONS_ZONE_NO_PROXY_MODE	2029	0x7ed	Cannot delete SOA or NS records for a zone that is not deployed in proxy mode.
NSERR_DNS_ACTION_DROP_DEPRECATED	2031	0x7ef	Drop is deprecated. Please use Drop action type under Responder feature.
NSERR_DNS_ACTION_REWRITE_RESPONSE_DEPRECATED	2039	0x7f7	Rewrite Response is deprecated. Please use replace_dns_answer_section action type under Rewrite feature.
NSERR_IDENTIFIER_INVAL_THRESHOLD	2040	0x7f8	Stream Identifier min and max threshold cannot both be zero and min threshold cannot be greater than max threshold
NSERR_IDENTIFIER_INVAL_SORT	2041	0x7f9	Stream Identifier sort option is invalid
NSERR_NO_SUCH_DNSPROF	2042	0x7fa	DNS profile does not exist
NSERR_DNS_PROFILE_SVC_TYPE	2043	0x7fb	DNS profile cannot be set to this service type
NSERR_CSGSLB_POLICY_SET_NOTALLOWED	2046	0x7fe	Policy bound to CS GSLB vserver must have a gslb targetvserver
NSERR_CS_POLICY_SET_NOTALLOWED	2045	0x7fd	Policy bound to non CS GSLB vserver must not have a gslb targetvserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

GSLB2 ERRORS

May 20, 2016

GSLB2 ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_DNS_VIEW_NOENT	3008	0xbc0	DNS View does not exist
NSERR_GSLB_LOC_NOMEM	3009	0xbc1	Memory allocation for location failed.
NSERR_DNS_PRIO_INVAL_VALUE	3010	0xbc2	Cannot bind DNS Policy with priority > 65535
NSERR_DNSPROF_EXISTS	3011	0xbc3	DNS Profile already exists.
NSERR_DNSPROF_DEF	3012	0xbc4	Cannot remove default DNS Profile.
NSERR_GSLB_LOC_EXISTS	3013	0xbc5	Custom Location Overlaps with an already existing location.
NSERR_GSLB_LOC_NOENT	3014	0xbc6	Custom Location does not exist.
NSERR_GSLB_NOMEM	3015	0xbc7	Not enough memory for GSLB operation.
NSERR_DNS_INT_POL	3016	0xbc8	Internal Policy is not allowed to bind if builtin parameter is not set with any value.
NSERR_GSLB_MEP	3017	0xbc9	Cannot enable MEP (Network Metric / Persistency Exchange) on a Child Site.
NSERR_IP_ROUTE_EXISTS	3018	0xbca	Host route enabled on this IP, cannot be used for GSLB Site IP.
NSERR_DNS_VIEW_EXISTS	3019	0xbcb	DNS View already exists
NSERR_NAME_VSVR_PERM	3020	0xbcc	Operation not permitted on a Vserver based Nameserver.
NSERR_DNS_POL_INTERNAL	3021	0xbcd	Cannot remove or unbind internal DNS policy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_GSLB_NAPTR_SERVICE_SET	3023	0xbcf	Setting of these parameters on naptr gslb service is not allowed
NSERR_GSLB_NAPTR_REPLACEMENT_UNSET	3025	0xbd1	Unsetting of naptr replacement is not allowed if the gslb service is bound to gslb vserver of NAPTR dnsrectype.
NSERR_GSLBVIP_NAPTR_UNSUPPORTED_LBMETHODS	3026	0xbd2	GSLB vserver having NAPTR services can have only CUSTOM LOAD as lb method.
NSERR_GSLB_NAPTR_VSERVER_SET	3027	0xbd3	Setting of these parameters on natpr gslb vserver is not allowed.
NSERR_GSLB_NAPTR_DOMAIN_SIZE	3028	0xbd4	NAPTR domain cannot be created - name length exceeds maximum
NSERR_GSLB_NAPTR_DOMAIN_SYNTAX	3029	0xbd5	NAPTR domain cannot be created - name is invalid
NSERR_GSLB_NAPTR_REPLACEMENT_SUFFIX_SYNTAX	3030	0xbd6	Invalid NATPR Replacement Suffix
NSERR_GSLB_NON_NAPTR_SERVICE	3031	0xbd7	Operation permitted only on NAPTR GSLB service

© 1999-2017 Citrix Systems, Inc. All rights reserved.

HTML INJECTION ERRORS

May 20, 2016

HTML INJECTION ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PREBODY_FILE_NOT_FOUND	3840	0xF00	Prebody file not found
NSERR_PREBODY_FILE_TOO_BIG	3841	0xF01	Prebody file too big
NSERR_PREBODY_FILE_READ_ERR	3842	0xF02	Prebody file read error
NSERR_POSTBODY_FILE_NOT_FOUND	3843	0xF03	Postbody file not found
NSERR_POSTBODY_FILE_TOO_BIG	3844	0xF04	Postbody file too big
NSERR_POSTBODY_FILE_READ_ERR	3845	0xF05	Postbody file read error
NSERR_HTML_INJECTION_VAR_EXISTS	3846	0xF06	HTML injection variable already exists
NSERR_HTML_INJECTION_VAR_NEXISTS	3847	0xF07	HTML injection variable does not exist
NSERR_HTML_INJECTION_VAR_NAME_TOO_LONG	3851	0xF0B	variable name too long
NSERR_HTML_INJECTION_INVALID_VAR_NAME	3852	0xF0C	Invalid variable name
NSERR_HTML_INJECTION_INCOMLETE_VAR_DEFINE	3853	0xF0D	Unpaired %% for variable name

© 1999-2017 Citrix Systems, Inc. All rights reserved.

ICA CLIENT VERSION ERRORS

May 20, 2016

ICA CLIENT VERSION ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ICA_CLIENT_VERSION_UNKNOWN_CLIENT_TYPE	3536	0xdd0	Unknown ICA client type.
NSERR_ICA_CLIENT_VERSION_DEFAULT_WHITELISTING	3537	0xdd1	Failed to initialize the ICA whitelist: Switched to default static ICA whitelisting
NSERR_ICA_CLIENT_VERSION_NOT_FOUND_IN_EXCEPTION_LIST	3538	0xdd2	The ICA client version you are requesting to remove from the exception list is not present in the exception list.
NSERR_ICA_CLIENT_VERSION_INCORRECT_BASE_VERSION_FORMAT	3539	0xdd3	Invalid format for the ICA client base version entered.
NSERR_ICA_CLIENT_VERSION_INCORRECT_EXCEPTION_LIST_FORMAT	3540	0xdd4	Invalid format for the ICA exception list entered.
NSERR_ICA_CLIENT_VERSION_DUPLICATES_RECEIVED_TO_REMOVE	3541	0xdd5	Two or more of the same client versions entered in the ICA exception list.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

INAT ERRORS

May 20, 2016

INAT ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PUBLICIP	3888	0xf30	Invalid Public IP address
NSERR_PRIVATEIP	3889	0xf31	Invalid Private IP address
NSERR_PROXYIP	3890	0xf32	Invalid Proxy IP address
NSERR_WILDCARDTCPVIP	3891	0xf33	TCPPROXY enabled when wildcard TCP Vserver already present
NSERR_WILDCARDANYVIP	3892	0xf34	INAT rule with wildcard ANY vserver, not allowed
NSERR_FTPVIPPRESENT	3893	0xf35	FTP enabled when FTP vserver already present
NSERR_V46_USIP	3894	0xf36	USIP is not allowed in mixed topology
NSERR_INCOMPATIBLE_IP	3895	0xf37	Private and Proxy IP should share the same IP Type
NSERR_ONLY_NAT46_SUPPORTED	3896	0xf38	Stateless mode only supported for IPv4->IPV6
NSERR_NAT46_PROXY_IP	3897	0xf39	Proxy IP is not supported in stateless mode
NSERR_NAT46_INVALID_PREFIXLEN	3898	0xf3A	Prefix length should be 32/40/48/56/64/96
NSERR_NAT46_INVALID_SETTING	3899	0xf3B	Invalid configuration for nat46
NSERR_NAT46_PRIVATEIP_EXISTS	3900	0xf3C	Private IP already configured in another nat46 rule
NSERR_TFTPVIPPRESENT	3901	0xf3d	TFTP enabled when TFTP vserver already present

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message

© 1999-2017 Citrix Systems, Inc. All rights reserved.

INTERNAL NAMESERVER ERRORS

May 20, 2016

INTERNAL NAMESERVER ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NAME_SVR_EXISTS	2561	0xa01	Name server already exists.
NSERR_NAME_SVR_IDNS_NEMPTY	2562	0xa02	Name servers already configured.
NSERR_NAME_SVR_IVALPOLICY	2563	0xa03	Invalid LB method for vserver-based name server.
NSERR_NAME_SVR_IVALPROTO	2564	0xa04	Invalid service type for vserver-based name server.
NSERR_NAME_SVR_ADD_NEXIST_VIP	2565	0xa05	vserver does not exist.
NSERR_NAME_SVR_NEXIST	2566	0xa06	Name server does not exist.
NSERR_NAME_SVR_IDNS_PERM	2567	0xa07	Operation on internal entity not permitted
NSERR_NAME_SVR_IP_EXISTS	2568	0xa08	Name server exists with this IP address
NSERR_NAME_SVR_TYPE_NEXIST_VIP	2569	0xa09	vserver of given type does not exist.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

INTF ERRORS

May 20, 2016

INTF ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_1GSFPSPEEDLIMIT	3968	0xf80	1G SFP's are restricted to speed 1000 or AUTO only
NSERR_1GSFPDUPLEXLIMIT	3969	0xf81	1G SFP's are restricted to duplex FULL or AUTO only
NSERR_1GSPEEDLIMIT	3970	0xf82	1G port can not be configured as 10G speed
NSERR_1GDUPLEXLIMIT	3971	0xf83	1G port can not be configured as HALF duplex
NSERR_10GSPEEDLIMIT	3972	0xf84	10G ports on this platform can only be configured as speed 1G, 10G or AUTO only
NSERR_10GDUPLEXLIMIT	3973	0xf85	10G ports on this platform can only be configured as duplex FULL or AUTO only
NSERR_10GAUTONEGLIMIT	3974	0xf86	10G ports on this platform do not support AUTONEG
NSERR_10GSPEEDONLY	3975	0xf87	10G ports on this platform are restricted to speed 10000 or AUTO only
NSERR_FULLDUPLEXONLY	3976	0xf88	These ports do not support HALF duplex mode
NSERR_AUTOINVALSPEED	3977	0xf89	Static CLAG does not support speed as AUTO
NSERR_CLAGSPEEDREQ	3978	0xf8a	Static CLAG explicitly requires speed to be configured other than AUTO
NSERR_THROUGHPUTLIMIT	3979	0xf8b	Throughput limit error. Possible values: 1G port - 1000Mbps, 10G port - 10000Mbps, 40G port - 40000Mbps, Link Aggregation - 80000Mbps
NSERR_BANDWIDTHLIMIT	3980	0xf8c	Bandwidth limit error. Possible values: 1G port - 1000Mbps, 10G port - 10000Mbps, 40G port - 40000Mbps, Link Aggregation - 80000Mbps

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_VIF_LACP_CAUTION	3982	0xf8e	AUTO Dynamic LA works in only pass-through interface mode.
NSERR_LA_SLAVE_LIMIT	3983	0xf8f	Trying to bind more interfaces than allowed limit
NSERR_SLAVE_MTU_MISMATCH	3987	0xf93	Interface MTU should be same as Channel MTU.
NSERR_BKPLANE_MTU_MISMATCH	3988	0xf94	Interface MTU should be default or same as Backplane MTU.
NSERR_MTU_NOTALLOWED	3989	0xf95	MTU change is not allowed on this interface.
NSERR_BACKPLANE_MTU_LIMIT	3990	0xf96	Backplane Interface, Minimum user configured value for MTU is 1578.
NSERR_INTERFACE_SLAVE_MTU_LIMIT	3991	0xf97	Interface is a part of channel, MTU change is not allowed.
NSERR_PTP_CLUS_L3	3992	0xf98	Cluster INC Mode Enabled, PTP state change is not allowed.
NSERR_40GDUPLEXLIMIT	3993	0xf99	40G ports on this platform can only be configured as duplex FULL or AUTO only
NSERR_40GSPEEDONLY	3994	0xf9a	40G ports on this platform are restricted to speed 40000 or AUTO only
NSERR_LRSET_FC_NOT_ALLOWED	3996	0xf9c	Flow Controle settings not allowed for link redundant set interface
NSERR_LRSET_LACP_SET_NOT_ALLOWED	3997	0xf9d	Dynamic LA(LACP) settings are not allowed for link redundant set interface
NSERR_LRSET_LLDPSET_NOT_ALLOWED	3998	0xf9e	LLDP settings are not allowed for link redundant set interface
NSERR_LRSET_SPEED_NOT_ALLOWED	3999	0xf9f	speed setting not allowed for link redundant set interface
NSERR_LRSET_PRIORITY_NOT_ALLOWED	4026	0xfba	lrset priority not allowed for link redundant set master interface
NSERR_LRSET_THROUGHPUT_NOT_ALLOWED	4027	0xfbb	throughput setting not allowed for link redundant set master interface
NSERR_LRSET_BW_SET_NOT_ALLOWED	4028	0xfbc	Bandwidth settings are not allowed for link redundant set interface
			Lagtype setting not allowed for link redundant set

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_LRSET_DUPLEX_NOT_ALLOWED	4030	0xfbe	Duplex setting not allowed for link redundant set interface
NSERR_LRSET_AUTO_NEG_NOT_ALLOWED	4031	0xfbf	Disabling autonegotiation not allowed for link redundant set interface
NSERR_JF_ESX_VERSION_SUPPORT	4070	0xfe6	Jumbo Frames are supported on ESX 5.5 version and onward

© 1999-2017 Citrix Systems, Inc. All rights reserved.

LB ERRORS

May 20, 2016

LB ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SLESSLB_LBMETHOD_NOTSUPPORTED	1280	0x500	Incompatible LB method for sessionless vserver
NSERR_SLESSLB_PERSIST_NOTSUPPORTED	1281	0x501	Incompatible persistence method for sessionless vserver
NSERR_SLESSLB_TYPE_NOTSUPPORTED	1282	0x502	Sessionless vserver must be of type ANY, DNS or UDP
NSERR_SLESSLB_MODE_NOTSUPPORTED	1283	0x503	Sessionless vserver must have MAC or IPTUNNEL mode set
NSERR_SLESSLB_SVC_USIPNOTSET	1284	0x504	Service bound to a sessionless vserver must have UseSourceIP mode enabled
NSERR_WILDCARDVIP_LBMETHOD_INVAL	1286	0x506	Invalid LB method for a wildcard vserver
NSERR_WILDCARDVIP_PERSIST_INVAL	1287	0x507	Invalid persistence policy for a wildcard vserver
NSERR_LB_SO_THRESHOLD	1288	0x508	Spill-over threshold should be a non-zero value
NSERR_CONNFAILOVER_USIP	1289	0x509	Service must have Use Source IP option set in order to be bound to a connection failover enabled virtual server
NSERR_CONNFAILOVER_SERVICE	1290	0x50a	Connection failover can only be enabled on a virtual server of service type ANY
NSERR_LB_SO_DYNAMICCON_THRESHOLD	1291	0x50B	Spill-over threshold cannot be set for dynamic connection spill-over method
NSERR_LB_SO_ADDRVIP	1292	0x50C	DYNAMICCONNECTION spill-over cannot be set on a non-LB vserver
NSERR_CONNFAILOVER_NOTFOR_SLESS	1293	0x50d	Connection Failover is not supported for sessionless lb vserver\n
NSERR_NEWSVC_PERCENT_CAPPED	1294	0x50e	Percentage value of requests capped at100

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SIP_NOCALLID	1296	0x510	Missing Call-ID header field
NSERR_SIP_NOVIA	1297	0x511	Missing Via header field
NSERR_SIP_NOCSEQ	1298	0x512	Missing CSeq header field
NSERR_SIP_NOTO	1299	0x513	Missing To header field
NSERR_SIP_NOFROM	1300	0x514	Missing From header field
NSERR_SIP_NOMAX_FORWARDS	1301	0x515	Missing Max-Forwards header field
NSERR_SIP_SERVICE_UNAVAILABLE	1302	0x516	Service unavailable
NSERR_SVC_NOT_BOUND	1303	0x517	Service not bound to Vserver
NSERR_SVCTYPE_MISMATCH	1304	0x518	Service type mismatch between the vserver and service
NSERR_CH_CON_NOT_ALLOWED	1305	0x519	CONNECTION header cannot be set
NSERR_CH_INCOMPLETE_HDR	1312	0x520	Custom Header string should end with CRLF
NSERR_SPILLOVERDISABLED	1313	0x521	No spillover on the vserver due to max client setting '0' on a bound service
NSERR_MAXTIMEEXCEEDED	1314	0x522	Maximum time value for interval or responsetimeout or downtime is exceeded, rounding to maximum
NSERR_SCINVALPROTO	1315	0x523	Cannot bind SC policy to non-HTTP service or vserver
NSERR_ROOTREC	1316	0x524	Atleast one nameserver record should exist for the root domain
NSERR_ROOT_GLUEREC	1317	0x525	Root domain nameserver address record should have atleast one ip address
NSERR_VIPINSERT_NOT_SUPPORTED	1356	0x54c	Vserver IP and port header insertion option not supported on this vserver.
NSERR_VSVR_ALRDY_BOUND	1318	0x526	This vserver is already bound to a WLM

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INVALID_WLM_BINDING	1319	0x527	to WLM
NSERR_VSVR_NOT_BOUND	1320	0x528	Vserver not bound to WLM
NSERR_WLM_EXISTS	1321	0x529	WLM already configured on this IP, port
NSERR_INVALID_TIMEOUT	1322	0x52a	Invalid persistence timeout: default timeout assigned
NSERR_CONNFAILOVER_INCMODE	1323	0x52b	Stateful connection failover cannot be enabled in INC mode
NSERR_CONNFAILOVER_TCPB	1324	0x52c	Stateful connection failover cannot be enabled on a vserver that binds services with TCP buffering enabled
NSERR_CONNFAILOVER_TCPB_SVC_BIND	1326	0x52e	Service with TCP buffering enabled cannot be bound to a vserver with stateful connection failover enabled
NSERR_CONNFAILOVER_TCPB_SVC_PARAM	1327	0x52f	TCP buffering cannot be enabled on a service bound to a vserver with stateful connection failover enabled
NSERR_CONNFAILOVER_SERVICE_STATEFULL	1328	0x530	Stateful connection failover cannot be enabled on a vserver of this service type
NSERR_CONNFAILOVER_SSL_SVC_BIND	1329	0x531	SSL TCP service cannot be bound to a TCP vserver with stateful connection failover enabled
NSERR_CONNFAILOVER_HA_INC_NODE	1330	0x532	Ha node with INC mode enabled cannot be added to the system that has vserver(s) with stateful connection failover enabled
NSERR_CONNFAILOVER_SSL_SVC	1331	0x533	Stateful connection failover cannot be enabled on a vserver that binds services of type SSL_TCP
NSERR_INVALID_PERSISTENCE	1332	0x534	Invalid Persistence type
NSERR_RTSPSESSION_INVALID	1333	0x535	Invalid RTSP session entry
NSERR_LB_METHOD_SAMEAS_BACKUP_METHOD	1334	0x536	Backup load balancing method cannot be the same as the primary load balancing method
NSERR_SERVER_EXIST	1335	0x537	Server already exists

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_LB_SO_HEALTH	1336	0x538	HEALTH spill-over cannot be set on a non-LB vserver
NSERR_REQ_RULE_MISSING	1337	0x539	Request Rule is required, when Response rule is specified.
NSERR_CONNFAILOVER_BINDLO	1338	0x53a	Loopback service cannot be bound to vserver with connection failover enabled \n
NSERR_CONNFAILOVER_SETWITHLO	1339	0x53b	Connection Failover cannot be set to STATEFUL/STATELESS on vserver with bound loopback service\n
NSERR_REDIRECT_URL_NOT_APPLICABLE	1340	0x53c	Redirect URL cannot be configured on this vserver.
NSERR_STATEFUL_CONNFAILOVER_IPV6	1341	0x53d	Stateful connection failover cannot be enabled on an IPv6 vserver
NSERR_CONNFAILOVER_NOT_SUPPORTED	1342	0x53e	Connection Failover is not supported for pattern based vservers
NSERR_NOBIND_ONPUSH	1343	0x53f	Bindings not allowed on push vserver
NSERR_PUSH_VSVR	1344	0x540	Option cannot be set on push vserver
NSERR_ONLY_HTTPSSL	1345	0x541	Option can only be set on http/ ssl vserver
NSERR_PUSH_BIND_EXISTS	1346	0x542	A push vserver is already bound. Unbind before binding new vserver
NSERR_SET_PUSHVS_ONLY	1347	0x543	Push vserver should be of type push or ssl_push only
NSERR_SLESS_NO_SUPPORT	1348	0x544	Not supported for sessionless vservers
NSERR_CONNFAILOVER_V6SERVICE	1349	0x545	Connection Failover cannot be set to STATEFUL on vserver with IPv6 services bound
NSERR_CONNFAILOVER_VSERVER	1350	0x546	IPV6 service cannot be bound to vserver with connection failover set to STATEFUL
NSERR_LISTEN_POLICY_BIND	1351	0X547	Error in binding listen policy to vserver(normal vserver already present)
NSERR_NO_LISTEN_POLICY_FOR_DUMMYVS	1352	0x548	Cannot bind listen policy to dummy vservers
			Error in binding listen policy to

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			already present)
NSERR_NO_LISTEN_PRI	1354	0X54a	Error in binding listen priority to normal vserver
NSERR_NORMAL_VS_NONE_LISTENPOL	1355	0x54b	The vserver already has None Listen Policy\n
NSERR_NO_LISPRI_FOR_NONE_LISPOL	1357	0x54d	None listenpolicy doesnot have listen priority\n
NSERR_INVALID_POLICY_STRING	1358	0x54e	Invalid listenpolicy entered\n
NSERR_NODATALENOFFFORVS	1359	0x54f	Dataoffset and datalength are not required for this vserver\n
NSERR_LBGRPPERSISWITHRDP	1285	0x505	Persistency is not applicable for RDP vserver or LB group which has RDP vserver part of it
NSERR_CS_SOTHRESHOLD	1370	0x55a	Spillover threshold must be given with spillover method in Content Switching vserver
NSERR_INVALID_INSERTVSERVERIPPORT	1392	0x570	V6TOV4MAPPING is not applicable to ipv4 vserver
NSERR_LB_SERVER_NOTEXIST	1388	0x56c	The Target LB Vserver does not exist
NSERR_BOUNDTO_CSACT	1371	0x55b	The LB Vserver referred by Content Switching action cannot be removed
NSERR_BOUNDTO_CSPOLICY	1372	0x55c	The Content Switching Action bound to Content Switching Policy cannot be removed
NSERR_INVALID_MULTIPLEBIND	1373	0x55d	Multiple Bind not supported for Content Switching Advanced policies without action
NSERR_CSPOLICY_HASACTION	1374	0x55e	LB vserver/gotoPriorityExpression/invoke not allowed during bind when content swtiching policy already has an action.
NSERR_CS_NOTSUPPORT_LB	1375	0x55f	The Protocol of given LB vserver is not supported by Content Switching action
NSERR_ACTION_NOTNOOP_CS	1386	0x56a	Do not provide gotoPriorityExpression/invoke because policy has action
			Either LB vserver or GotoPriorityExpression or Invoke must

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CSPEPOLICY_SETNOACTION	1389	0x56d	Switching policy without action. Content Switching classic policy does not support action
NSERR_POLICY_USED_FOR_INVOKE	1390	0x56e	Do not give target LB beacuse the policy has been bound to multiple entities
NSERR_SPECIFY_BACKUP_NODE	1393	0x571	Please specify the backup node.
NSERR_SAME_NODES	1394	0x572	Owner and backup nodes need to different.
NSERR_INCOMPATIBLE_POLICY_SERVER_TYPE	1391	0x56f	Content switching vserver service type is not compatible with given policy expression
NSERR_PEPOLICY_BIND_NOTALLOWED	1395	0x573	Binding of classic content switching policy to content switching policylabel is not allowed
NSERR_TCPKA_IDLETIME_TOO_LOW	1396	0x574	Connection idle time cannot be lower than keep-alive probe interval
NSERR_OWNER_NODE_NOT_IN_CLUSTER	1397	0x575	Owner node is not in the cluster config or is NOT ACTIVE
NSERR_BACKUP_NODE_NOT_IN_CLUSTER	1398	0x576	Backup node is not in the cluster config or is NOT ACTIVE
NSERR_SPOTTED_VSERVER_NOT_SUPPORTED_IN_STANDALONE	1399	0x577	Spotted vserver configuration not supported in standalone or HA deplyoment
NSERR_SPOTTED_CONFIG_NO_MATCH	1400	0x578	Spotted config of the backup verser does not match the vserver
NSERR_OPTION_NOT_SUPPORTED	1401	0x579	Skip persistency is supported only for ANY and UDP service types
NSERR_OPTION_NOT_SUPPORTED_IP	1402	0x57a	BYPASS is not supported for IP based vservers
NSERR_OWNER_NODE_NOT_SET	1403	0x57b	Owner node should be specified for cluster
NSERR_INVALID_RENAME	1404	0x57c	Rename is supported only for Advanced policies
NSERR_INVAL_LOCAL_NAMSVR_TYPE	1405	0x57d	Only UDP based local nameservers are allowed
NSERR_INVALID_TFTPORT	1406	0x57e	Invalid port for tftp vserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NO_RENAME	1407	0x57f	Rename not supported
NSERR_USE_BINDCR_CMD	88	0x58	Use Bind CR command to bind CR Vserver instead of Bind CS
NSERR_LBREFFERED_IN_CSBINDING	1409	0x581	Cannot remove LB Vserver which is reffered in CS policy bindings
NSERR_DEFAULTLB_NOT_BOUND	1410	0x582	default LB mentioned is not bound
NSERR_NG_BIND_ENTITY	1411	0x583	Error binding the node group to an entity
NSERR_ENTITY_BOUND_TO_ANOTHER_NG	1412	0x584	The specified entity is already bound to a different node group
NSERR_ENTITY_ALREADY_BOUND	1413	0x585	The specified entity is already bound to this node group
NSERR_ENTITY_NOT_BOUND_TO_NG	1414	0x586	The specified entity is not bound to any node group
NSERR_NG_NOT_FOUND	1415	0x587	Nodegroup is not found
NSERR_NG_HAS_BINDINGS	1416	0x588	Nodegroup cannot be deleted, it has entities bound to it
NSERR_VSERVER_CANNOT_BIND	1417	0x589	Vserver cannot bind to the nodegroup
NSERR_VSERVER_CANNOT_UNBIND	1418	0x58a	Vserver cannot unbind to the nodegroup
NSERR_NG_CANNOT_BE_STRICT	1419	0x58b	Nodegroup cannot be made strict as there are no nodes bound to it, but entities are bound
NSERR_NG_STRICT_AND_ZERODEPTH	1420	0x58c	Cannot bind an entity to a strict nodegroup with no nodes bound to it
NSERR_UNBINDING_LAST_NODE_FROM_STRICT_NG	1421	0x58d	Cannot unbind the last node from a strict nodegroup with entities bound to it
NSERR_NG_ALREADY_ADDED	1422	0x58e	Nodegroup is already added
NSERR_NG_ENTITY_NOMEM	1423	0x58f	No memory to establish entity to nodegroup association
NSERR_NG_MAX_CONFIGURED	1424	0x590	Cannot add nodegroup; maximum number of nodegroups already added

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INVAL_CS_ACTION	1425	0x591	Action given is not a CS action
NSERR_SET_CSACTION_DISALLOWED	1426	0x592	Set disallowed for CS action from Target LB/targetVserver to Target Expression and vice versa
NSERR_RETAIN_VLAN	1427	0x593	macmoderetainvlan applicable only for wildcard(IP and port) macmode vserver
NSERR_BIND_BDGGROUP_SYNCVLAN	1428	0x594	Sync vlan cannot be bound to bridgegroup
NSERR_DEL_SYNCVLAN	1429	0x595	Sync vlan cannot be deleted
NSERR_CSPEPOL_PARTITION_NOT_ALLOWED	1430	0x596	Current NON-default partition only supports Advanced Polices.
NSERR_LESSTHAN_PPE	1438	0x59e	Max/Min Reusepool connections should be greater than or equal to running Packet Engines.
NSERR_MCLIMITS	1295	0x50f	Dummy error to avoid issues when upgrade done from 9.3.
NSERR_SIP_BAD_REQUEST	1431	0x597	Invalid SIP request.
NSERR_SIP_NO_CONTENT_LEN	1432	0x598	No content length present.
NSERR_SIP_RET_MAX_INCOM_LEN	1433	0x599	SIP InComp max limit reached.
NSERR_SIP_RET_RETRY	1434	0x59a	SIP packet retry.
NSERR_SIP_RET_NO_MEM	1435	0x59b	SIP no memory available.
NSERR_SIP_RET_HEADER_NOT_COMPLETE	1436	0x59c	Incomplete SIP headers.
NSERR_SIP_RET_FAILURE	1437	0x59d	SIP operation failed.
NSERR_CS_GSLB_DEFAULT	1439	0x59f	Default vserver type mismatch
NSERR_CS_GSLB_DNSRT_MISMATCH	1306	0x51a	DNSrecordType of CS GSLB vserver and Target Vserver should be same.
NSERR_CS_GSLB_INVALID_TARGET	1307	0x51b	CS Vserver of type GSLB can only have GSLB vserver as Target.
			CS Policy has no action or CS action

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_GSLB_SVC_ALREADY_BOUND_TO_CS	1309	0x51d	GSLB service bound to target Vserver is already bound to CS vserver
NSERR_CS_POLICY_NO_RULE	1310	0x51e	Binding CS policy to CS GSLB vserver requires rule
NSERR_VIP_BACKUP_ISCSGSLB	1311	0x51f	CS GSLB vserver cannot be backup of a non- CS GSLB vserver

© 1999-2017 Citrix Systems, Inc. All rights reserved.

LB2 ERRORS

Jul 07, 2017

LB2 ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ADDRPRESENT	1376	0x560	Address(s) is invalid or already present in this action, will be discarded.
NSERR_PERSISTENCE_PERSISTAVP_NOTDEFINED	1377	0x561	Persist Avp Number Must be defined For diameter Persistency
NSERR_MAX_DIAMETER_AVP_EXCEEDED	1378	0x562	Maximum number of diameter Persist Avp Codes exceeded
NSERR_AUTOSCALE_MIN_GT_MAX	1379	0x563	Minimum Autoscale member count exceeds the maximum value
NSERR_MEMBERPORT_REQD	1380	0x564	Member Port option has to be specified when autoScale option POLICY is chosen
NSERR_COOKIEINSERT_NOTSET	1381	0x565	Persistence type is not COOKIEINSERT.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

LSN ERRORS

May 20, 2016

LSN ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_LSN_NOPOOL	3960	0xf78	No such pool
NSERR_LSN_NOTRANSPRFL	3961	0xf79	No such transportprofile
NSERR_LSN_NOAPPSPRFL	3866	0xf1a	No such applicationprofile
NSERR_LSN_POOLNOTBOUND	4052	0xfd4	Pool not bound to group
NSERR_LSN_PRFLNOTBOUND	4053	0xfd5	Profile not bound to group
NSERR_LSN_NOTREADY	4054	0xfd6	LSN not initialized. Please check memory allocated to LSN.
NSERR_LSN_MAXBINDINGS	4055	0xfd7	Maximum bindings allowed exceeded
NSERR_LSN_ACL_DETERMINISTICNAT	4056	0xfd8	Client with ACLs not allowed for deterministic nat
NSERR_LSN_ACLTYPE	4057	0xfd9	ACL with action other than ALLOW or establishedflag set
NSERR_LSN_ACLINUSE	4058	0xfda	ACL already bound to a LSN/RNAT rule
NSERR_64LIB_MEM_NOTSUFFICIENT	4059	0xfdb	Active memory less than minimum required for LSN and Subscriber Session Store
NSERR_LSN_NOBINDING	4060	0xfdc	No such binding
NSERR_LSN_NONETWORK	4061	0xfdd	No such network
NSERR_LSN_CLIENTGROUP	4062	0xfde	Please remove client's group

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_LSN_NOCLIENT	4063	0xfdf	No client
NSERR_LSN_NOGLBLLOG	4072	0xfe8	Logging is disabled at Global level
NSERR_LSN_CLIENTINUSE	4080	0xff0	Client already present in a group
NSERR_LSN_PORTBLOCKSIZE	4081	0xff1	Portblocksize should be multiple of 256
NSERR_LSN_WRONG_PORTBLOCKSIZE	855	0x357	Portblocksize should not exceed 64512
NSERR_LSN_BINDINGEXIST	4082	0xff2	Binding already exists
NSERR_LSN_POOLGROUPTYPE	4083	0xff3	Both pool and group should be of same nat type
NSERR_LSN_POOLREUSE_DETERMINISTICNAT	4084	0xff4	Pool can be bound to only one group for determinstic nat
NSERR_LSN_PORT_APPSPROFEXIST	4085	0xff5	Profile already exist for one or more ports in this/bound group(s)
NSERR_LSN_TCPONLY	4086	0xff6	Allowed only for TCP profile
NSERR_LSN_MAPPINGFILTERINGTYPE	4087	0xff7	Filtering should be more or equally strict as mapping
NSERR_LSN_ICMPAPPSPROF_MAPPINGFILTERINGPORT	4088	0xff8	Port dependent Filtering or Mapping not allowed for ICMP
NSERR_LSN_ICMPAPPSPROF_PORT	4089	0xff9	Port bindings not allowed for ICMP applicationprofile
NSERR_LSN_STATIC_NOGROUP	4090	0xffa	Client ip not part of any active lsn group
NSERR_LSN_STATIC_INVPOOL	4091	0xffb	No natip available or natip given is invalid
NSERR_LSN_FINTIMGTSESSTIM	4092	0xffc	Finrsttimeout should not be more than session timeout
NSERR_LSN_SETPORTBLOCK_DETERMINISTICNAT	4093	0xffd	Port block size cannot be modified for deterministic nat
			Maximum allowed subscribers

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_64LIB_MEM_INCOMPLETE	3943	0xf67	Active memory could be different from configured memory limit, Please check 'show extendedmemory parameter'
NSERR_64LIB_MEM_EXCEEDSMAX	3944	0xf68	Configured memory cannot exceed maximum limit, Please check 'show extendedmemory parameter'
NSERR_64LIB_MEM_SIZE2M	3945	0xf69	Configured memory should be in multiples of 2MB
NSERR_LSN_IPINSERT	3957	0xf75	LSN IP addition failed
NSERR_LSN_ICMP_TIMEOUT_EXCEEDED	3958	0xf76	Maximum value of ICMP session timeout is 240 seconds
NSERR_LSN_TCP_UDP_TIMEOUT_EXCEEDED	3867	0xf1b	Maximum value of TCP/UDP session timeout is 600 seconds
NSERR_LSN_APPSPROF_PORTZERO	4073	0xfe9	Cannot bind/unbind port 0 to Application Profile
NSERR_LSN_MEM_DECR	3995	0xf9b	Active memory cannot be decreased without reboot
NSERR_LSN_NOSIPALGPRFL	3547	0xddb	No such sipalgprofile
NSERR_LSN_REM_APPPROFILE_ALGPROFILE_BOUND	3548	0xddc	Can not remove Appfprofile, SIP ALG profile and RTSP ALG Profile should be unbound fron the groups bound to the Appprofile
NSERR_LSN_UNBIND_APPPROFILE_ALGPROFILE_BOUND	3549	0xddd	Can not unbind to the Appprofile, SIP ALG profile and RTSP ALG Profile should be unbound fron the groups bound to the Appprofile
NSERR_LSN_NORTSPALGPRFL	4074	0xfea	No such rtspalgprofile
NSERR_LSN_SIPALGPROF_PORTZERO	4075	0xfeb	Cannot add port 0 to SIP ALG Profile
NSERR_LSN_RTSPALGPROF_PORTZERO	4076	0xfec	Cannot add port 0 to RTSP ALG Profile

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_LSN_IP6PROF_NAT64	4078	0xfee	With NAT64 add only IP6Prefix
NSERR_LSN_NETWORK6_EXISTS	3546	0xdda	network 6 based LSN client already exists for this network 6 address(Duplicate config)
NSERR_LSN_CLIENTNETWORK_INCORRECT_BINDING	3551	0xddf	Both Client and Network type MUST be of the same type, IPv4, IPv6 or DS-Lite Deterministic
NSERR_LSN_CLIENTGROUPTYPE	3553	0xde1	Both client and group should be of same type
NSERR_LSN_DSLITEPREFIXLEN	3554	0xde2	For a DS-Lite DETERMINISTIC type client, network6 MUST have a prefix length 128
NSERR_LSN_NOIP6PROFILE	3555	0xde3	No such IP6 profile
NSERR_LSN_IP6GROUP	4079	0xfef	Please remove associated group first
NSERR_LSN_DSLITE_INVALIDIP6	3556	0xde4	With DS-Lite add only SNIP6.
NSERR_LSN_GROUP_IP6INUSE	3557	0xde5	IP6Profile is already been used for some other LSN Group
NSERR_LSN_DUP_IP6PROFILE	3552	0xde0	Either IPv6 address or NAT Prefix already been used
NSERR_LSN_SIPALG_CANNOT_BIND	3882	0xf2a	Cannot bind SIPALG profile to the group, either the group is not in full cone mode or IP Pool pair is not enabled
NSERR_LSN_RTSPALG_CANNOT_BIND	3883	0xf2b	Cannot bind RTSP profile to the group, either the group is not in full cone mode or IP Pool pair is not enabled
NSERR_LSN_SIPALG_TRANSPORT_PROTOCOL_BINDINGEXIST	3884	0xf2c	Can not bind more than one SIPALG profile of same transport protocol to the lsn group
NSERR_LSN_RTSPALG_TRANSPORT_PROTOCOL_BINDINGEXIST	3885	0xf2d	Can not bind more than one RTSPALG profile of same transport protocol to the lsn group

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_LSN_CANNOT_UNBIND_APPPROFILE	3886	0xf2e	unless all SIP ALG Profiles and all RTSP ALG Profiles are unbound from all the groups to which appsprofile is bound
NSERR_LSN_RTSPALGPROF_TRANSPORT_SUPPORTED	3887	0xf2f	RTSP ALG profile for only TCP is supported
NSERR_LSN_SIPALGPROF_PORT_NOT_ADDED	3353	0xd19	Atleast sipsrcportrange or sipdstportrange should be added with positive portrange
NSERR_LSN_GRP_SESS_LIMIT	852	0x354	Group session limit must be greater than the number of Packet Engines
NSERR_LSN_DETER_POOL	3903	0xf3f	Cannot enable portblock allocation on deterministic pool
NSERR_LSN_SIP_RTSP_ALG_MAPPING_FILTERING_IPPOOLING	853	0x355	Can not set appsprofile unless all SIP ALG profiles and all RTSP ALG profiles are unbound from all the groups to which appsprofile is bound
NSERR_LSN_PORTBLOCKSIZE_NOTALLOWED	856	0x358	Portblock size should not be changed to zero from nonzero and to nonzero from zero
NSERR_LSN_POOLGROUP_PORTBLOCK_MISMATCH	857	0x359	Pool's portblock allocation and group's portblocksize fields are not compatable
NSERR_LSN_GRPID_PERM	3818	0xeea	GroupID should not be explicitly specified
NSERR_LSN_INVALID_DSLITE_PREFIX_LEN	3819	0xeeb	Invalid Ipv6 Prefix length for dslite client ipv6 network(only /128 prefix is allowed)
NSERR_LSN_NOLOGPRFL	3384	0xd38	No such log profile
NSERR_LSN_DOHASYNC	3396	0xd44	Force HA Sync should be given to sync Primary's pre-flush operation LSN sessions

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NAT64 ERRORS

Jul 07, 2017

NAT64 ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NAT64_ACL_EXISTS	3959	0xf77	ACL is bound to another NAT64 rule

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NET ERRORS

May 20, 2016

NET ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ACL_TIMED_OUT	977	0x3d1	ACL timed out
NSERR_MSR_ADVT_NULL_RT	978	0x3d2	Cannot monitor or advertise NULL route
NSERR_DEF_RT_UNSET	979	0x3d3	Unset operation on default route not allowed
NSERR_L2L3_VPATH	980	0x3d4	Unset operation on L2 Vpath or L3 Vpath routes not allowed
NSERR_ACTIVE_PERM_DIR_NULL_RT	981	0x3d5	Unset operation on active or permanent or direct or null routes not allowed
NSERR_DHCP_RT	982	0x3d6	Unset operation on dhcp route not allowed
NSERR_LB_RT	983	0x3d7	Unset operation on lb route not allowed
NSERR_MSR_STATIC_RT	984	0x3d8	Cannot set MSR on a non-static route
NSERR_ADV_DEF_RT	985	0x3d9	Advertise on default route not allowed
NSERR_VLAN_RT	986	0x3da	vlan route cannot be deleted through rm route command
NSERR_PERM_DIR_NULL_RT	987	0x3db	Unset operation on permanent or direct or null routes not allowed
NSERR_NOENTRY_RT	988	0x3dc	Entry not found in the routing table
NSERR_DIR_RT	989	0x3dd	Cannot add a direct route
NSERR_LOOP_RT	990	0x3de	Cannot add a route in loopback network

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_MON_ADV_NULL_RT	992	0x3e0	Cannot set msr or advertise on null gateway
NSERR_CLEAR_STATIC_DIR_L3VPATH	993	0x3e1	Clearing of static or direct or l3vpath not allowed
NSERR_HA_PEER_IP	994	0x3e2	Cannot add ip address. This is peer's NSIP
NSERR_NON_LOCAL_IP	995	0x3e3	IP address is already configured on a node in the cluster
NSERR_ADD_NSIP	996	0x3e4	Adding NSIP not allowed
NSERR_VRID_VIP_ONLY	997	0x3e5	VRID can be set only on VIP
NSERR_ARP_ICMP_SNIP_NSIP	998	0x3e6	ARP and ICMP cannot be disabled on SNIP | NSIP
NSERR_IPADDR_DISABLE	999	0x3e7	IP address state of SNIP | NSIP cannot be disabled
NSERR_NODE_NOEXIST	1000	0x3e8	Cluster node does not exist
NSERR_CLIP_SPOTTED_IP	1001	0x3e9	CLIP cannot be added to a specific node
NSERR_IP_MGMTACCESS	1002	0x3ea	Management Access cannot be enabled on this ip address
NSERR_IPTUN_MULTICAST	1003	0x3eb	Remote ipaddress cannot be multicast address
NSERR_IPTUNNEL_INVALIDNETMASK	1004	0x3ec	IP tunnel netmask should be 255.255.255.255.
NSERR_IPTUNNEL_VLAN_NETMASK	1005	0x3ed	Invalid netmask. For non-default vlan netmask should be multicast address.
NSERR_IPTUN_INV_LIP	1006	0x3ee	Invalid local ip. Either the local ip is not configured or local ipaddress has to be one among SNIP | VIP | MIP
NSERR_VXLAN_LIP_NOT_SNIP	1007	0x3ef	Invalid local ipaddress. For vxlan local ipaddress has to be SNIP
NSERR_INVALID_IPTUNNEL_NAME	1008	0x3f0	ERROR: Invalid IPTunnel name
NSERR_IPTUNNEL_PREFXLEN_GT_128	1009	0x3f1	IP tunnel prefix length should not be more than 128

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_TUN_PBR_DEP	1010	0x3f2	Cannot remove tunnel. A PBR is depending on this tunnel
NSERR_TUN_NOEXIST	1011	0x3f3	Tunnel does not exist
NSERR_INV_DROPFRAGCPUTHRESH	1012	0x3f4	Invalid dropFragCpuThreshold parameter. Should >=1 and <=100
NSERR_SRCIP_NOEXIST	1013	0x3f5	srcIP does not exist
NSERR_TUN_ON_DEF_ROUTE	1014	0x3f6	Cannot add tunnel on default route
NSERR_VLAN_NOT_SNIP_MIP_VIP	1015	0x3f7	IP address has to be either SNIP, MIP or VIP
NSERR_ACL_PROTO_TCP	1016	0x3f8	Protocol should be TCP for ESTABLISHED flag to be set
NSERR_HA_VERSION_MISMATCH	1017	0x3f9	HA version mismatch between primary and secondary
NSERR_HA_SYNC_VLAN	1018	0x3fa	syncvlan should be l2 vlan. No IP address should be bound to this VLAN
NSERR_LOOP_FIS	1019	0x3fb	Loopback interface cannot be bound to Failover Interface Set (FIS)
NSERR_VLAN_LOOP	1020	0x3fc	Loopback interface cannot be bound to vlan
NSERR_VLAN_TAGG_INTF_NOSUPP	1021	0x3fd	Vlan tagging is not supported on this interface
NSERR_DEFVLAN_IPBIND	1022	0x3fe	IP address to vlan binding is not allowed for default vlan
NSERR_PERMANENT_ARP_ND6_ENTRY	1023	0x3ff	Can't remove permanent ARP/ND6 entry
NSERR_HA_VER_MISMATCH_PROP_SYNC	1447	0x5a7	HA version mismatch. prop and sync not allowed
NSERR_VXLAN_TUNN_BOUND	1458	0x5b2	Cannot remove iptunnel. One or more Vxlan is bound to this iptunnel.
NSERR_HOSTRT_OSPFAREA_UNSET	1489	0x5d1	Unset of ospfArea and HostRtGw is allowed only for VIP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NETBRIDGE ERRORS

May 20, 2016

NETBRIDGE ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_TNL_NETBRIDGED	3920	0xf50	Tunnel bound to netbridge
NSERR_TNL_GRE	3921	0xf51	Tunnel protocol is not GRE
NSERR_TNL_RIPMASK	3922	0xf52	Remote IP mask of the tunnel should be 255.255.255.255
NSERR_VLAN_NETBRIDGED	3923	0xf53	Vlan already bound to some netbridge
NSERR_MAX_NETBRIDGE_TUNNEL	3924	0xf54	reached maximum netbridge tunnel bindings
NSERR_TUN_BOUND	3925	0xf55	Tunnel already bound to this netbridge
NSERR_SUBNET_BOUND	3926	0xf56	Subnet already bound to this netbridge
NSERR_TUN_CONF_NETBRIDGE	3927	0xf57	Tunnel is not configured
NSERR_VLAN_CONF_NETBRIDGE	3928	0xf58	Vlan is not configured
NSERR_IP_INVAL_NETBRIDGE	3929	0xf59	IP is not binded to netbridge
NSERR_IPV6_INVAL_NETBRIDGE	3930	0xf5a	IPV6 is not binded to netbridge
NSERR_INV_NETBRIDGE_NAME	3931	0xf5b	Netbridge name should be >=1 and <=32
NSERR_GRE_IPSEC_LIP_NOT_GIVEN	3932	0xf5c	Local ip address has to be given for GRE or IPSEC tunnels
NSERR_NETBRIDGE_TUNNELBOUNDTOPBR	3933	0xf5d	Tunnel is a nexthop for PBR

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			ETHERNET with DOT1Q
NSERR_NETBRIDGE_ONEVLAN_ETHERNET	3935	0xf5f	Only one VLAN can be bound when GRE carries plain ETHERNET
NSERR_NETBRIDGE_TUNNELPAYLOAD_DIFFERENT	851	0x353	All bound tunnels should have same GRE payload

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NETPROFILE ERRORS

May 20, 2016

NETPROFILE ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_IPSET_REF	3952	0xf70	IP set must be unbound before it can be removed
NSERR_NETPROFILE_REF	3953	0xf71	Netprofile must be unbound before it can be removed
NSERR_IPSET_BOUND	3954	0xf72	IP set is already bound to the network profile
NSERR_NO_SUCH_IPSET	3955	0xf73	IP set does not exist
NSERR_NO_SUCH_NETPROF	3956	0xf74	Netprofile does not exist
NSERR_OVERLAPPING_PORTRANGE	3558	0xde6	This range is overlapping the existing portrange.
NSERR_NO_SUCH_PORTRANGE	3559	0xde7	Port range does not exist
NSERR_NO_SUCH_NATRULE	3560	0xde8	natRule does not exist
NSERR_NATRULE_EXIST	3561	0xde9	natrule already exist

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NITRO ERRORS

May 20, 2016

NITRO ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NITRO_INVALID_OBJECTNAME	1232	0x4d0	Invalid object name
NSERR_NITRO_INVALID_JSON_INPUT	1233	0x4d1	Invalid JSON input
NSERR_NITRO_INVALID_JSON_DATATYPE	1234	0x4d2	Invalid JSON data type
NSERR_NITRO_INVALID_XML_INPUT	1235	0x4d3	Invalid XML input
NSERR_NITRO_INVALID_DATATYPE	1236	0x4d4	Invalid NITRO data type
NSERR_NITRO_INVALID_METHOD	1237	0x4d5	Invalid method name. It should be either post, put, get, stat or delete
NSERR_NITRO_PARSE_ERROR	1238	0x4d6	NITRO parse error
NSERR_NITRO_CMDEXEC_FAILED	1239	0x4d7	NITRO command Execution failed
NSERR_NITRO_INVALID_ACTION	1240	0x4d8	Invalid nitro action or operation
NSERR_NITRO_ROLLBACK_FAILED	1241	0x4d9	Rollback failed. Manually correct the configuration
NSERR_NITRO_ROLLBACK_NOT_SUPPORTED	1242	0x4da	Rollback not supported for this action
NSERR_NITRO_BULK_FAILED	1243	0x4db	Bulk operation failed
NSERR_NSAPP_TEMPLATE_EXISTS	1248	0x4e0	Template already exists
NSERR_NSAPP_INVALID_TEMPLATE	1249	0x4e1	Invalid Template

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NSAPP_FILE_ERROR	1251	0x4e3	Can not open the file
NSERR_NSAPP_INVALID_APP_INPUT	1252	0x4e4	Invalid Application input
NSERR_NSAPP_EXCEEDED_FILELENGTH	1253	0x4e5	File length exceeded 256 characters
NSERR_NSAPP_ENDPOINT_INUSE	1254	0x4e6	Public endpoint in use
NSERR_NSAPP_PROTOCOL_MISMATCH	1255	0x4e7	Protocol mismatch with existing configuration
NSERR_NSAPP_INVALID_VARNAME	1256	0x4e8	Invalid variable name
NSERR_NSAPP_ZIP_FILE	1257	0x4e9	Error in zip format
NSERR_NSAPP_MAXEP_REACHED	1258	0x4ea	Only one endpoint can be configured
NSERR_NSAPP_NOT_EXIST	1259	0x4eb	Application does not exist
NSERR_NSAPP_INVALID_SERVICETYPE	1260	0x4ec	Invalid servicetype - HTTP/HTTPS/SSL are only allowed
NSERR_NSAPP_TEMPLATE_FORMAT_ERROR	1261	0x4ed	Template format error
NSERR_NSAPP_APP_WITHOUT_APPUNITS	1262	0x4ee	Application without appunits can not be exported
NSERR_NSAPP_SERVICRGROUP_EXISTS	1263	0x4ef	Service group with this name already exists
NSERR_NSAPP_VARIABLE_NOT_EXISTS	1264	0x4f0	Variables do not exist in this template file

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NNM ERRORS

May 20, 2016

NNM ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NNM_FAILED	2528	0x9e0	General NNM failure
NSERR_NNM_CONN_CLEANED	2529	0x9e1	NNM connection was cleaned
NSERR_NNM_CONN_FAILED	2530	0x9e2	NNM connection failed
NSERR_NNM_NO_CONN	2531	0x9e3	NNM connection was not found
NSERR_NNM_CONN_DROPPED	2532	0x9e4	NNM connection dropped
NSERR_NNM_EINJECT	2533	0x9e5	NNM error injection
NSERR_NNM_VERSION_UNSUPPORTED	2534	0x9e6	NNM group payload version is unsupported
NSERR_NNM_TIMEOUT_MSG	2535	0x9e7	NNM timeout unable to send message
NSERR_NNM_TIMEOUT_ACK	2536	0x9e8	NNM timeout waiting for ACK
NSERR_NNM_MCAST	2537	0x9e9	NNM multicast/broadcast failed
NSERR_NNM_PENDING	2538	0x9ea	NNM operation is in a pending state (e.g., ACK held)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

PARTITION ERRORS

May 20, 2016

PARTITION ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_MAX_PARTITION	2752	0xac0	Maximum allowed partition limit reached
NSERR_PARTITION_NAME_EXIST	2753	0xac1	Partition name already exist
NSERR_NO_PARTITION	2755	0xac3	Partition does not exist
NSERR_DEL_PARTITION_BUSY	2756	0xac4	Partition is busy, can't delete
NSERR_DEL_PARTITION_INUSE	2757	0xac5	Partition is in use, can't delete
NSERR_NO_DEL_DFLT_PARTITION	2758	0xac6	Default Partition can't be delete
NSERR_RM_PARTITION_CONFIRM	2759	0xac7	Are you sure you want to remove partition (Y/N)? [N]:
NSERR_NOT_SUPPORTFORPARTITION	2760	0xac8	This feature is not supported on partition
NSERR_AP_CONFIG_NOT_ALLOWED	2761	0xac9	This configuration option is not supported on non-default partition
NSERR_VLAN_BDGGRP_PARTITION_EXIST	2762	0xaca	VLAN or BridgeGroup bound to a partition can not be deleted.
NSERR_PARTITION_NOT_SUPPORTED	2763	0xacb	This feature does not support partitioning
NSERR_VLAN_BDGGRP_PARTITION	2764	0xacc	The specified VLAN or BridgeGroup is already bound to other partition.
NSERR_PQ_ARG_UNSUPPORTED_PARTITION	2765	0xacd	PQ feature is not supported in non-default partition.
NSERR_SC_ARG_UNSUPPORTED_PARTITION	2766	0xace	SC feature is not supported in non-default partition.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			reached
NSERR_VXLAN_PARTITION_NOT_SUPPORTED	2768	0xad0	This bridgegroup/vlan has a vxlan extension - cannot bind it to a partition
NSERR_PARTITION_ID_EXIST	2790	0xae6	Partition-id is already in use by another partition.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

PBR ERRORS

Jul 07, 2017

PBR ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PBR_NEXTHOP_NOTDIRECT	912	0x390	PBR Nexthop should be direct
NSERR_PBR_NOLOOPBACK	913	0x391	PBR cannot be configured on the loopback interface
NSERR_PBRDELERROR	914	0x392	This PBR has already been removed
NSERR_PBR_IPPIP_EXISTS	915	0x393	PBR with identical parameter specification already exists
NSERR_PBR_PRIORITY_EXISTS	916	0x394	PBR with this priority already exists
NSERR_PBRMODCFGINFO	918	0x396	PBR modified, use 'apply pbrs' to commit this operation
NSERR_PBR_NEXTHOP_REQD	919	0x397	PBR Nexthop is required
NSERR_PBR_L2_CONFIG_INFO	920	0x398	L2 based PBRs work only for routed traffic
NSERR_PBRMODCFG_L2_INFO	921	0x399	L2 based PBRs work only for routed traffic: PBR modified, use 'apply pbrs' to commit this operation
NSERR_PBR_NO_MONITOR_GATEWAY	922	0x39a	Monitor can't be configured when nexthop is configured as gateway name
NSERR_PBR_INVALID_IPORGATEWAY	923	0x39b	Invalid Nexthop IPaddress/Gateway name
NSERR_PBR_MAX_RULE_EXCEEDED	924	0x39c	Number of PBRs on the system exceeds Maximum
NSERR_PBR_NEXTHOP_TUNNEL_REQD	925	0x39d	PBR Nexthop or IpTunnel is required
NSERR_PBR_DENY_MSR	926	0x39e	MSR cannot be enabled on PBR with DENY config

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			matching and 1s for not matching
Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PBR_IPSEC_ALREADY_BOUND	1445	0x5a5	For IPSEC tunnels, Only one PBR is allowed to be bound to a tunnel
NSERR_PBR_REMOVED	1446	0x5a6	PBR is removed
NSERR_PBR_L2PARAM_GRE	1456	0x5b0	PBR is having L2PARAM. Cannot have GRE as nexthop.
NSERR_PBR_VXLAN	1457	0x5b1	VXLAN cannot be a nexthop.
NSERR_PBR_TUNNELBOUNDTONETBRIDGE	933	0x3a5	Tunnel is already bound to Netbridge

© 1999-2017 Citrix Systems, Inc. All rights reserved.

PBR6 ERRORS

Jul 07, 2017

PBR6 ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PBR6_IPPIP_EXISTS	880	0x370	PBR6 with identical parameter specification already exists
NSERR_PBR6_DELERROR	881	0x371	PBR6 has already been removed
NSERR_PBR6_ADDERROR	882	0x372	Port can be specified only if protocol is TCP (6) or UDP (17)
NSERR_PBR6_PRIORITY_EXISTS	883	0x373	PBR6 with this priority already exists
NSERR_PBR6_ICMP_REQD	884	0x374	ICMPv6 type/code can be specified only if protocol is ICMPv6 (1)
NSERR_PBR6_UNSPECADDR	885	0x375	unspecified address (::) can not be configured in PBR6
NSERR_PBR6_MODCFGINFO	886	0x376	PBR6 modified, apply pbrs6 to activate change
NSERR_PBR6_PREFIXLEN	887	0x377	Prefix length should not be configured in PBR6. (Use range instead)
NSERR_PBR6_SPECIFY_NEXTHOPVLAN	888	0x378	Nexthop VLAN should be specified with Link local gateway address
NSERR_PBR6_INVALID_GATEWAY	889	0x379	Invalid PBR6 Gateway
NSERR_PBR6_INVALID_NEXTHOP_VLAN	890	0x37a	Invalid Nexthop VLAN ID
NSERR_PBR6_MAX_RULE_EXCEEDED	891	0x37b	Number of PBR6s on the system exceeds Maximum

© 1999-2017 Citrix Systems, Inc. All rights reserved.

May 20, 2016

PI ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PI_LONGVSVRNAME	3040	0xbe0	The name of the label must be less than 32 characters
NSERR_PI_PRIORITY_EXISTS	3041	0xbe1	A policy is already bound to the specified priority
NSERR_PICON_GOTO_NOT_END	3042	0xbe2	GotoPriorityExpression for this policy has to be END
NSERR_INVAL_THRESHOLD	3043	0xbe3	Invalid threshold size value. Threshold size must be lesser than the sample size
NSERR_DEF_INVALID_UNBIND	3044	0xbe4	Default binding cannot be unbinded
NSERR_BAD_INIT_VALUE	3045	0xbe5	Invalid initial value
NSERR_INIT_VALUE_TOO_BIG	3046	0xbe6	Initial value is larger than the maximum allowed text length and the ifValueTooBig action is undef
NSERR_INIT_VALUE_TRUNCATED	3047	0xbe7	Initial value has been truncated to the maximum allowed text length because the ifValueTooBig action is truncate
NSERR_BAD_ASSIGN_LHS	3048	0xbe8	Invalid -var parameter: expect $variable-name for a ulong or text variable or $variable-name[key-expression] for a map variable.
NSERR_BAD_OP_FOR_TYPE	3049	0xbe9	Op is not allowed for the variable type.
NSERR_BAD_VAR_TYPE	3050	0xbea	Invalid type: expected one of ulong, text, or map.
NSERR_ASSIGNMENT_LOG_UNSUPPORTED	3051	0xbeb	Log action is not supported with assignment action
NSERR_UNDEF_EXCEPTION	3052	0xbec	Undef exception raised.
NSERR_NO_VARIABLE_VALUE	3053	0xbed	No value exists for the variable and key.
NSERR_UNSUPPORTED_VARIABLE_USE	3055	0xbef	Variables are not supported for use with DNS.
NSERR_BAD_KEY_TYPE	3056	0xbf0	Invalid key type: expected type text
NSERR_BAD_SINGLETON_VAR	3057	0xbf1	Not a singleton variable
NSERR_BAD_MAP_VAR	3058	0xbf2	Not a map variable
NSERR_UNKNOWN_VARIABLE	3059	0xbf3	Variable is not configured.
NSERR_MAP_WITHOUT_KEY	3060	0xbf4	Map variable used without a key.
NSERR_SINGLETON_WITH_KEY	3061	0xbf5	Singleton variable used with a key.
NSERR_BAD_TEXT_MAX_LENGTH	3062	0xbf6	Invalid text type: expected text(max-length), where max-length is an unsigned integer >= 1 and <= 64000.
NSERR_BAD_MAP_TYPE_SYNTAX	3063	0xbf7	Invalid map type: expected map(key-type, value-type, max-entries), where max-entries is an unsigned long value <= 18446744073709551615 and greater than 0.
NSERR_BAD_MAP_KEY_TYPE	3064	0xbf8	Map key type must be text.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_BAD_KEY_VALUE_LENGTH	3066	0xbfa	Sum of the maximum map key and value lengths must be <= 64000.
NSERR_BAD_MAP_SET_TYPE	3067	0xbfb	Invalid type parameter for set.
NSERR_TRANSACTION_MAPS_NOT_SUPPORTED	3068	0xbfc	Maps with transaction scope are not supported.
NSERR_TRANSACTION_EXPIRE_NOT_SUPPORTED	3070	0xbfe	Expiration with transaction scope variables is not supported.
NSERR_INVOKE_NOT_SUPPORTED	3071	0xbff	Policy does not support invoke.
NSERR_INVALPIEXPR	3073	0xc01	Invalid expression
NSERR_INVALINTOP	3074	0xc02	Expecting numeric arguments in this context for '-', '+', '*' or '/'
NSERR_INVALCOMPARE	3075	0xc03	Invalid arguments to compare operation
NSERR_INVALBOOLOP	3076	0xc04	The '!', ' | | ' and '&&' operators can have only boolean argument(s)
NSERR_ARGNEG	3077	0xc05	Argument cannot be negative
NSERR_ARGEXCEED	3078	0xc06	Argument exceeds maximum allowed value
NSERR_ARGZERO	3079	0xc07	Operand or function argument should be non-zero
NSERR_MAXHEADER	3080	0xc08	Number of custom header exceeds limit
NSERR_INVALPICEXPR	3081	0xc09	Expression syntax error
NSERR_CEXPRLIMIT	3082	0xc0a	Expression too long
NSERR_EXPRLIMIT	3083	0xc0b	Expression too long
NSERR_SEXPRLIMIT	3084	0xc0c	String expression too long
NSERR_INVALPISEXPR	3085	0xc0d	String expression syntax error
NSERR_NOBIDI	3086	0xc0e	Bi-directional expression not allowed
NSERR_NOLAB	3087	0xc0f	Policy Label does not exist
NSERR_INTERNAL_LABEL_RM	3088	0xc10	Default Policy labels cannot be removed
NSERR_POLICY_FLOWTYPE_NONE	3089	0xc11	Policy should either be a request or response policy
NSERR_INVAL_PRIORITY	3090	0xc12	Priority should be in the range from 1 to 2147483647
NSERR_NUMERICEXPR	3091	0xc13	Expression should evaluate to numeric value
NSERR_BOOLEXPR	3092	0xc14	Expression should evaluate to true or false
NSERR_NOTBOUND	3093	0xc15	Cannot unbind a policy that is not bound
NSERR_POLICY_INUSE	3094	0xc16	Policy name already in use
NSERR_POLICY_SET_NOTALLOWED	3095	0xc17	Invalid rule/action for bound policy

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INVALID_BIND	3096	0xc18	Policy cannot be bound to specified policy label
NSERR_USE_INVOKE_RESULT	3097	0xc19	USE_INVOCATION_RESULT can be used only with invoke
NSERR_INVAL_EXPR_DATA_TYPE	3098	0xc1a	Invalid expression data type
NSERR_EXPR_CONST_ONLY	3099	0xc1b	Only constant parameters are allowed for function
NSERR_ALT_EXPR_STRING_ONLY	3100	0xc1c	Only string expressions are allowed for ALT
NSERR_ACTION_DEF_SET_INVAL	3101	0xc1d	Default action cannot be modified
NSERR_POLICY_DEF_SET_INVAL	3102	0xc1e	Default policy cannot be modified
NSERR_POLICY_LABEL_DEF_SET_INVAL	3103	0xc1f	Default policylabel cannot be modified
NSERR_INVAL_UNBIND	3104	0xc20	Policy not bound to specified policy label
NSERR_REGEX_INVALID	3105	0xc21	Invalid regular expression
NSERR_REGEX_BACKREF	3106	0xc22	Backreference in regular expression
NSERR_REGEX_RECURSIVE	3107	0xc23	Recursive regular expression present
NSERR_PIXL_EXPR_UNSAFE	3108	0xc24	Input expression is unsafe
NSERR_PI_ACTION_MAX_REF_REACHED	3109	0xc25	Cannot create policy. Binding limit for action reached (65536)
NSERR_PI_INVALID_UNSET	3110	0xc26	Cannot unset rule or action. No changes done
NSERR_EMPTY_STRING	3111	0xc27	Zero length string not allowed
NSERR_INVAL_BTW_ARGS	3112	0xc28	Arguments to BETWEEN are incorrect
NSERR_INVAL_ARG_SPECIFIED	3113	0xc29	GotoPriorityExpression, flowtype and invoke apply only to rewrite, responder and cache policies
NSERR_INVAL_ARG_CSPL	3854	0xf0e	Target vserver cannot be specified along with GotoPriorityExpression and invoke.
NSERR_FLOWTYPE_NOT_APPLICABLE	3114	0xc2a	Flowtype applies only to rewrite, responder and cache policies
NSERR_INVAL_ARG_SPECIFIED_CS	3115	0xc2b	Flowtype and invoke apply only to Advanced policies.
NSERR_INVAL_GOTOEXPR_CS	3116	0xc2c	GotoPriorityExpression applies only to Advanced policies
NSERR_REBIND_FAILED	3117	0xc2d	Rebinding of policy with new bind parameters failed. The policy is unbound.
NSERR_SECOND_BIND	3118	0xc2e	A policy can be bound only once
NSERR_NONHTTP_VS	3119	0xc2f	Specified policy can be bound only to HTTP/SSL vserver
NSERR_PI_LEX_FAILURE	3344	0xd10	Error in parsing
NSERR_PI_INVALIDNAME	3856	0xF10	Invalid PI entity name; names must begin with an ASCII alphabetic character or underscore and must contain only ASCII alphanumerics or underscores; reserved words may not be used
NSERR_PI_RST_CONN	3855	0xF0F	PI reset connection

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INVAL_XPATH_EXPR	3858	0xf12	XPath Expression Compilation Failed
NSERR_XPATH_EXPR_NOT_SUPPORTED	3859	0xf13	Unsupported XPath
NSERR_TARGET_INVAL	3860	0xf14	Target Vserver cannot be specified for this policy
NSERR_NON_RESP_HTTP_VS_LIST	3861	0xf15	Specified policy can be bound only to HTTP/SSL/TCP/SSL_BRIDGE/FTP/DNS_TCP/RTSP/SSL_TCP/NNTP/MSSQL/MySQL/ORACLE/DIAMETER/RADIUS/ANY vserver
NSERR_INVAL_VSERVER_BIND	3862	0xf16	Policy can be bound only to LB/CS vserver types
NSERR_APP_FREED_NSB	3863	0xf17	Application reset/drop connection and freed nsb
NSERR_MULT_BIND_INVAL	3864	0xf18	CVPN Policies cannot be bound to multiple entities
NSERR_CLASSIC_POLICY_ALREADY_BOUND	3865	0xF19	Cannot bind an advanced policy to a virtual server to which a classic policy is already bound
NSERR_ADVANCED_POLICY_ALREADY_BOUND	3872	0xF20	Cannot bind a classic policy to a virtual server to which an advanced policy is already bound
NSERR_MULT_UNBIND	3873	0xF21	Policy is bound to multiple global bindpoints. Please specify the appropriate bindpoint to unbind it.
NSERR_UNKNOWN_CERTKEY	3874	0xF22	The specified certkey object has not been configured
NSERR_NO_RSA_PUBLIC_KEY	3875	0xF23	The specified certkey object does not include a certificate with an RSA public key
NSERR_NO_RSA_PRIVATE_KEY	3876	0xF24	The specified certkey object does not include an RSA private key
NSERR_CERTKEY_NAME_TOO_LONG	3877	0xF25	The certkey name exceeds the maximum 31 characters
NSERR_INVALID_UTF_8	3878	0xF26	Invalid UTF-8 string
NSERR_UNRECOG_CHARSET	3879	0xF27	Unrecognized character set
NSERR_INCOMPATIBLE_NAME_EXPR_CHANGE	3880	0xF28	Incompatible named expression change for in-use expression.
NSERR_HEADER_EXPR_ONLY	3881	0xF29	Only classic expressions for headers allowed in EVAL_CLASSIC_EXPR
NSERR_RW_NOT_SUPPORTED_VS	3904	0xF40	Specified policy can be bound only to HTTP/SSL/TCP/SSL_TCP/DIAMETER/SSL_DIAMTER/SIP_UDP/RADIUS vserver
NSERR_INVALID_KEYVALUE	3905	0xf41	The keyvalue argument is incorrect.
NSERR_INVALID_RESOURCE_NAME	3906	0xF42	Resource of the given name does not exist
NSERR_INVALID_ACTION	3907	0xF43	Invalid action
NSERR_ASCII_ONLY_FUNCTION	3908	0xF44	Advanced expression function does not accept non-ASCII arguments
NSERR_INVALID_ENUM_VALUE	3909	0xF45	Invalid enumeration value
NSERR_INVALID_BINDPOINT	3910	0xF46	Invalid bind point.
NSERR_NO_FIPS_FOR_XML_DECRYPT	3911	0xF47	A FIPS key cannot be used for XML decryption.
NSERR_NO_CONFIGDB_DATA_SELECTED	3912	0xF48	No data selected from the configuration database.
NSERR_HEADER_NAME_LENGTH_LIMIT	3913	0xF49	Custom header name is too long.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_UNSUPPORTED_CVPN_SEARCH	3914	0xF4A	Dataset/dataset and xpath not supported for clientless VPN search.
NSERR_INVALID_HEADER_NAME	3915	0xF4B	Header name contains invalid characters (one or more of ( ) < > @ , ; : \\ \" / [ ] ? = { } and space) as per the HTTP spec RFC 2616. This is allowed to for non-conformant HTTP messages, but may not be what you intended.
NSERR_HEADER_NAME_TOO_LONG	3916	0xF4c	Header name is too long (maximum of 32 characters allowed).
NSERR_CSACTION_NOT_SUPPORTED	3918	0xF4E	Content Switching actions are not suported for classic content switching policies.
NSERR_PI_EXPR_TOO_BIG	3919	0xF4F	Expression too long - maximum length is 8191
NSERR_PI_INVALID_FUNCTION_NAME	3946	0xF6A	Cannot use PI reserved word as a function name.
NSERR_PI_FUNCTION_NAME_EXISTS	3947	0xF6B	Function with the same name already exists.
NSERR_EXTENSION_INVALID_UPDATE_NAME	3948	0xF6C	Add the extension before updating.
NSERR_PI_INVALID_EXTENSION_NAME	3949	0xF6D	Extension is not added.
NSERR_EXTENSION_FUNCTION_PROTOTYPE_CHANGED	3950	0xF6E	Extension has different prototype of a function than that is in use.
NSERR_EXTENSION_FUNCTION_UPDATE_PROBLEM	3951	0xF6F	Problem in updating function that is in use.
NSERR_EXTENSION_COMPILE_FAILURE	3962	0xF7A	Extension compilation error.
NSERR_EXTENSION_LOAD_FAILURE	3963	0xF7B	Extension loading error.
NSERR_PI_USED_FUNCTION_REMOVED	3964	0xF7C	Update attempted to remove a function that is in use.
NSERR_EXTENSION_FUNCTION_MULTIPLE_DEFINITIONS	3965	0xF7D	Extension has multiple definitions of a function.
NSERR_EXTENSION_INVALID_FUNCTION_PROTOTYPE	3966	0xF7E	Extension function's prototype is invalid.
NSERR_EXTENSION_MAX_FUNCTION_ARGUMENT	3967	0xF7F	Maximum number of function arguments can be 3.
NSERR_BAD_EXTENSION_TRACE_NAME	4038	0xFC6	Function or variable name can contain only alphanumerics and underscores and cannot begin with a digit.
NSERR_BAD_EXTENSION_TRACE_ARRAYS	3024	0xBD0	traceFunctions or traceVariables cannot be specified with trace off.
NSERR_PI_INVALID_GLOBAL_BINDTYPE	3828	0xEF4	Invalid global bindtype.
NSERR_INVALID_POLICY_GLOBAL_BINDING	3829	0xEF5	Policy cannot be bound/unbound to/from the given global bindtype.
NSERR_INVALID_PI_ANNOTATE	1465	0x5b9	Invalid PI annotation value
NSERR_INVALID_PI_ANNOTATE_LOGINSCHEMA	1466	0x5ba	Invalid loginSchema corresponding to PI annotation value
NSERR_ANGLE_BRACKETS_IN_TAGNAME	3830	0xEF6	Invalid syntax, angle brackets in tag specified

© 1999-2017 Citrix Systems, Inc. All rights reserved.

POL ERRORS

May 20, 2016

POL ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INVALIDPOL	2053	0x805	Binding invalid policy
NSERR_NOPOL	2054	0x806	No such policy exists
NSERR_RULEURL	2055	0x807	Rule or URL required
NSERR_DELMC	2056	0x808	Delay or maxConn argument required
NSERR_ACP	2057	0x809	Alternate content path not required
NSERR_ACS	2058	0x80A	Alternate content service not required
NSERR_NOSET_CEXP	2059	0x80B	No support to set compound expression value
NSERR_CEXP_DEPTH	2060	0x80C	Maximum recursive depth reached
NSERR_NOVPNAPP	2061	0x80D	No such intranet application exists
NSERR_NOSET_CSE	2062	0x80E	Changing rule to/from client security expression is not allowed when policy is in use
NSERR_INVAL_POLNAME	2063	0x80F	Invalid policy Name
NSERR_UNBIND_INVALIDPOL	2097	0x831	Policy not bound
NSERR_EXCEED_MAX_POL_LIMIT	2098	0x832	32 authentication policies are already bound
NSERR_INTERNAL_PI_ERROR	2103	0x837	Internal policy error

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NEG_POL_VIOL1	2104	0x838	Only negotiate and certificate policies can be bound together
NSERR_NEG_POL_VIOL2	2105	0x839	Negotiate policy in primary can be bound only along with ldap policy (with authentication turned off) in secondary
NSERR_NEG_POL_VIOL3	2106	0x83A	Negotiate policy in secondary can be bound only along with certificate policy in primary
NSERR_SAML_POL_VIOL1	2122	0x84A	SAML policy can only be the last in the cascade or the only one in the cascade
NSERR_SAML_POL_VIOL2	2123	0x84B	SAML policy in primary can be bound only along with ldap policy (with authentication turned off) in secondary
NSERR_SAML_POL_VIOL3	2124	0x84C	SAML policy cannot be bound to secondary cascade
NSERR_SAML_DISABLED	2125	0x84D	SAML policy binding is disabled
NSERR_LDAP_POL_VIOL1	2088	0x828	Only LDAP policies may be bound for group extraction.
NSERR_LDAP_POL_VIOL2	2089	0x829	An LDAP policy bound for group extraction must set -authentication DISABLED.
NSERR_NON_SELECTOR_POLICY	2107	0x83B	Only selector based policies can be bound to sql vserver
NSERR_BAD_SELECTOR	2108	0x83C	Selector does not match content group type
NSERR_BAD_CONTENTGROUP	2109	0x83D	Contentgroup not compatible
NSERR_SQL_POL_GLOBAL_BINDPOINT	2110	0x83E	Datastream cache policies cannot have global/override bindpoint
NSERR_APPQOE_INVALID_ACP	2111	0x83F	Alternate content path is invalid
NSERR_APPQOE_CLTRATE	2086	0x826	Client detect rate is not required
NSERR_APPQOE_DOSTHRESH_NREQ	2087	0x827	DoS attack threshold is not required
NSERR_APPQOE_DOSTHRESH_REQ	2090	0x82a	DoS attack threshold is required
NSERR_APPQOE_ACS_REQ	2091	0x82b	Alternate content service is required

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_APPQOE_ACP_REQ	2092	0x82c	Alternate content path is required with alternate content service
NSERR_CMPTYPE_MISMATCH	2093	0x82d	Compression policy may not take effect due to global parameter type mismatch
NSERR_APPQOE_VSVR_CONFIG_CONFLICT	2094	0x82E	AppQoE policy cannot be bound to vserver configured for PQ/SC
NSERR_APPQOE_SVC_CONFIG_CONFLICT	2095	0x82F	AppQoE policy cannot be bound to vserver with services configured for SC/HDOSP
NSERR_APPQOE_UNSUPPORTED_PROTO	2142	0x85E	Specified policy can be bound only to HTTP/SSL vserver
NSERR_APPQOE_UNSUPPORTED_VSVR	2143	0x85F	Specified policy can be bound only to LB vserver types
NSERR_AUTHN_CLASSIC_INVALID_ARG	2145	0x861	Invalid arguments for classic authentication policy binding
NSERR_AUTHN_ADVANCE_INVALID_ARG	2146	0x862	Invalid arguments for advanced authentication policy binding
NSERR_RENAME_NOT_PERMITTED	2147	0x863	Rename operation not permitted
NSERR_PI_AUTHZPLCY_NOT_SUPPORTED	2148	0x864	advanced authorization policy cannot be bound to aaa user/group
NSERR_SAML_2FACTOR_CONFLICT	2149	0x865	SAML 2-Factor policy should always be at the beginning of cascade
NSERR_CLASSIC_TMSESSPOL_ALREADY_BOUND	2152	0x868	Advanced TM Session Policy cannot be bound if Classic TM Session Policy is already bound to any entity (i.e. aaa user, aaa group, authentication vserver, tm global)
NSERR_ADVANCED_TMSESSPOL_ALREADY_BOUND	2153	0x869	Classic TM Session Policy cannot be bound if Advanced TM Session Policy is already bound to any entity (i.e. aaa user, aaa group, authentication vserver, tm global)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

PQ ERRORS

May 20, 2016

PQ ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_PQ_BINDVIP	1360	0x550	Cannot bind same priority policy to VIP
NSERR_PQ_INVALPRIO	1361	0x551	Invalid priority value
NSERR_PQ_2BIGRULE	1362	0x552	Rule argument too big
NSERR_PQ_INVALWT	1363	0x553	Invalid weight value
NSERR_PQ_POLEXIST	1364	0x554	Policy name already in use
NSERR_PQ_NOPOL	1365	0x555	No such PQ policy exists
NSERR_PQ_2MANYREF	1366	0x556	Cannot remove a policy which is bound to VIP
NSERR_PQ_NOLBVIP	1367	0x557	No such LB vserver exists
NSERR_PQ_PHSCONFIG	1368	0x558	Cannot configure PQ on VIP if physical service has PQ configuration
NSERR_PQ_NOBIND	1369	0x559	Cannot unbind PQ Policy which is not bound

© 1999-2017 Citrix Systems, Inc. All rights reserved.

REWRITE ERRORS

May 20, 2016

REWRITE ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_RW_ACT_INVAL	2817	0xb01	Invalid rewrite action
NSERR_RW_UNDEFACT_INVAL	2818	0xb02	Invalid undef action or log action
NSERR_ACTFLOWMISMATCH	2819	0xb03	Flow types of target and string expression are incompatible
NSERR_RONLY_TAR_EXPR	2820	0xb04	Target must be a simple reference to an unmodified portion of the message
NSERR_DATASET_BINDFAIL	2821	0xb05	Unable to bind the pattern to dataset/patset
NSERR_DATASET_UNBINDFAIL	2822	0xb06	Pattern does not exist in dataset/patset
NSERR_DATASET_NOTPRESENT	2823	0xb07	Dataset/Patset does not exist
NSERR_RSP_ACT_INVAL	2824	0xb08	Invalid responder action
NSERR_RSP_POLICY_FLOWTYPE_REQ	2825	0xb09	Responder policy must be a request policy
NSERR_TAR_FLOWTYPE_NRES	2826	0xb0a	Flow type of target should not be response type
NSERR_RSP_CONFIG_LOCK	2827	0xb0b	Responder configuration is temporarily disabled
NSERR_RSP_ACT_MUST_BE_NOOP	2828	0xb0c	Non-terminating policy must have NOOP action
NSERR_PATSET_INVALID_REGEX	2829	0xb0d	Patset contains invalid regex.
NSERR_DATASET_BINDFAIL_DUP_INDEX	2830	0xb0e	Pattern index already in use, try using other index
NSERR_PATSET_BINDFAIL_PATLEN_LT_WU_MINLEN	2833	0xb11	Patset search on a body target not allowed if any of the patset patterns are < 3 or > 112 bytes in length
NSERR_TARGET_NOT_ALLOWED_IN_RULE	2834	0xb12	Expression involving Target not allowed in rule.
NSERR_RW_TARGET_NOT_ALLOWED_IN_STRBUILDER	2835	0xb13	Expression involving Target not allowed in StringBuilder expression.
NSERR_PIT_INVAL	2836	0xb14	Invalid Packet data
NSERR_INVAL_INVOKEPOINT	2837	0xb15	Invalid invoke point
NSERR_PIT_MAX_PACKET	2838	0xb16	Packet size exceeds maximum size
			Regular expression for patterns not allowed when refine search is

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_EXTEND_INVALID	2840	0xb18	Extend not allowed for non-body expressions
NSERR_NON_EXTEND_EXPR	2841	0xb19	Non extend expressions are not allowed
NSERR_EXTEND_INVAL_PIRL	2842	0xb1a	Extend not allowed in the string builder expression
NSERR_INVAL_TAR_EXPR	2843	0xb1b	Invalid target expression
NSERR_INVAL_PATTERN_SEARCH_SET	2844	0xb1c	Cannot set both pattern and search
NSERR_INVAL_SEARCH_ARGS	2845	0xb1d	Invalid argument: search supports text, xpath, xpath_json, xpath_html, regex and patset
NSERR_RSP_ACT_MUST_BE_RESET_DROP	2846	0xb1e	Policy action must be DROP | RESET
NSERR_INVAL_SEARCH_SYNTAX	2847	0xb1f	Invalid search syntax
NSERR_INVAL_SEARCH_XPATH_SYNTAX	2848	0xb20	Invalid xpath syntax
NSERR_INVAL_SEARCH_PATSET_SYNTAX	2849	0xb21	Invalid patset syntax
NSERR_INVAL_SEARCH_REGEX_SYNTAX	2850	0xb22	Invalid regex syntax
NSERR_RW_REQ_BODY_NOT_ALLOWED	2851	0xb23	Request body based expression is not allowed
NSERR_RENAME_NOTSUPPORTED	2852	0xb24	Renaming this entity is not supported as entity based expression is configured
NSERR_ENTITY_REMOVAL_NOTALLOWED	2853	0xb25	Removing this entity is not allowed as entity based expression is configured
NSERR_PI_ENTITY_EXISTS	2854	0xb26	Advanced expression entity with same name already exists.
NSERR_INCOMPATIBLE_CALLOUT_CHANGE	2855	0xb27	Incompatible callout change for in-use callout.
NSERR_STRINGMAP_NOTPRESENT	2856	0xb28	String map does not exist
NSERR_HTTP_PROFILE_ACTION_NOT_ALLOWED	2857	0xb29	Responder action using HTTP protocol expressions is not allowed in HTTP profile
NSERR_RSP_ACT_MUST_BE_DROP_NOOP_RESPONDWITH	2858	0xb2A	Policy action must be DROP | NOOP | RESPONDWITH
NSERR_RSP_ACT_MUST_BE_DROP_NOOP	2899	0xb53	Policy action must be DROP | NOOP
NSERR_RSP_UNDEF_ACT_MUST_BE_DROP_NOOP	2859	0xb2B	Policy undef action must be DROP | NOOP
NSERR_INCOMPATIBLE_UNDEF	2860	0xb2C	Incompatible global undef action for SIP policy
NSERR_PIT_ACTION_EVAL_INVAL	2861	0xb2d	Action cannot be evaluated due to wrong input type.
NSERR_DATASET_INVALID	2866	0xb32	Invalid dataset element

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_TIMER_ACT_INVAL	2868	0xb34	Invalid timer action
NSERR_TIMER_ENTITY_INUSE	2869	0xb35	Timer entity name already in use
NSERR_TIMER_ENTITY_GLOBAL_BINDPOINT_INVAL	2870	0xb36	Invalid global bindpoint
NSERR_PATSET_BUILTIN	2871	0xb37	Built-in Patsets cannot be modified or deleted
NSERR_CROSS_MAX_LIMIT	2872	0xb38	Maximum value can be 2147483647
NSERR_INVALID_IPV4_FORMAT	2873	0xb39	Value should be in IPv4 format
NSERR_INVALID_IPV4_MAX_FIELD_VALUE	2880	0xb40	Maximum value of an IPv4 field can be 255
NSERR_INVALID_IPV6_FORMAT	2881	0xb41	Invalid IPv6 address format
NSERR_INVALID_NUMERIC_FORMAT	2882	0xb42	Value should be in decimal or hexadecimal format
NSERR_DATASET_PATTERN_ALREADY_BOUND	2883	0xb43	Specified pattern is already bound to dataset/patset
NSERR_PI_CIRCULAR_REFERENCE_DETECTED	2884	0xb44	Circular reference detected
NSERR_INVALID_ULONG_FORMAT	2885	0xb45	Value should be in unsigned long format like 23445, 88888999
NSERR_INVALID_MAC_ADDRESS_FORMAT	2886	0xb46	Value should be in mac address format like ff:ff:ff:ff:ff:ff
NSERR_INVALID_DOUBLE_FORMAT	2887	0xb47	Value should be in double format like 223, 22.3, 234e2
NSERR_CROSS_ULONG_MAX_LIMIT	2888	0xb48	Maximum value can be 18446744073709551615
NSERR_CA_ACT_INVAL	2889	0xb49	Invalid ContentAccelerator action
NSERR_INVAL_AVP_INSERT_EXPR	2890	0xb4a	Rewrite policy has invalid action. For Diameter bind points use DIAMETER.NEW_AVP() in action and for Radius bind points use RADIUS.NEW_AVP() in action as stringBuilderExpr expression to insert/replace AVP
NSERR_INVAL_AVP_LOOKUP_EXPR	2891	0xb4b	Rewrite policy has invalid action. For Diameter/Radius bind points, use AVP lookup expression as target expression. e.g. RADIUS.REQ.AVP(127)/DIAMETER.REQ.AVP(9000)/AVP_START/AVP_END
NSERR_INVAL_DELETE_AVP_EXPR	2892	0xb4c	Rewrite policy has invalid action. Deletion of AVP should be done using AVP based expression like RADIUS.REQ.AVP(33)/DIAMETER.REQ.AVP(9999)
NSERR_INVAL_NEW_MESSAGE_EXPR	2893	0xb4d	Responder policy has invalid action. Use expressions like DIAMETER.NEW_ANSWER/DIAMETER_NEW_REDIRECT as target expression to create new diameter message and RADIUS.NEW_ANSWER/RADIUS.NEW_ACCESSREJECT for RADIUS message
NSERR_INVAL_DIAMETER_REDIRECT_ACTION	2894	0xb4e	Binding Invalid Diameter action. Use DIAMETER.NEW_REDIRECT as target expression to create new redirect message
NSERR_INVAL_DIAMETER_ACTION	2895	0xb4f	Invalid action for Diameter bind point.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CA_URL_INVAL	2896	0xb50	URL length more than 2000
NSERR_PIT_CONNECTION_ERROR	2897	0xb51	Connection error
NSERR_INVAL_SEARCH_AVP	2898	0xb52	AVP code should be a number
NSERR_DATASET_PATTERN_LIMIT_OVERFLOW	2900	0xb54	Cannot bind more than 5000 patterns to a patset/dataset
NSERR_INVAL_RADIUS_AVPCODE	2901	0xb55	RADIUS AVP code should be in 0-255 number range
NSERR_INVAL_RADIUS_RESPONSE	2902	0xb56	Invalid RADIUS response code
NSERR_RESP_ACT_REASON_PHRASE_REQUIRED	2903	0xb57	Please also specify a reason phrase with a user-defined HTTP status code.
NSERR_RESP_ACT_RESPONSE_CODE_REASON_PHRASE_NOT_ALLOWED	2904	0xb58	responseStatusCode and reasonPhrase options are allowed only for respondwithhtmlpage and redirect action types.
NSERR_RESP_ACT_INVALID_REDIRECT_CODE	2905	0xb59	For redirect action type the response status code must be in the range 300-399.
NSERR_RESP_ACT_INVALID_HTTP_STATUS_CODE	2906	0xb5A	Invalid HTTP status code.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

RISE ERRORS

Jul 07, 2017

RISE ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_RISE_NOT_ENABLED	3473	0xd91	Feature RISE is not enabled
NSERR_RISE_NS_MODE_APBR_DISABLED	3474	0xd92	Operation not permitted as ns mode RISE_APBR is disabled. Retry after ns mode RISE_ABPR is enabled
NSERR_RISE_NS_MODE_RHI_DISABLED	3475	0xd93	Operation not permitted as ns mode RISE_RHI is disabled. Retry after ns mode RISE_RHI is enabled
NSERR_RISE_RHI_NOT_IMPLEMENTED	3476	0xd94	RISE RHI mode is not implemented
NSERR_RISE_NOT_LICENSED	3477	0xd95	RISE cannot be enabled in Standard NS license
NSERR_RISE_DISABLED_ALREADY	3478	0xd96	RISE mode is already disabled
NSERR_RISE_WARN_DIRECT_DISABLE	3479	0xd97	Disabling direct mode with one or more active RISE profile(s) on DIRECT ATTACH mode may cause corresponding RISE profile(s) to not come UP on reboot
NSERR_RISE_WARN_INDIRECT_DISABLE	3480	0xd98	Disabling indirect mode will cause any active RISE profile(s) to go down

© 1999-2017 Citrix Systems, Inc. All rights reserved.

RNAT ERRORS

Jul 07, 2017

RNAT ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_RNAT_NAME_EXISTS	976	0x3d0	A RNAT with same name exists
NSERR_VLAN_IPV6_LL_N_ALLOW	962	0x3c2	IP address cannot be link-local address
NSERR_VLAN_INTF_BOUND_NSVLAN	963	0x3c3	Cannot unbind interface from nsvlan
NSERR_VLAN_INVALID	964	0x3c4	Incorrect vlan specified
NSERR_SYNC_VLAN_IPBOUND	965	0x3c5	Ipaddress binding is not allowed on sync vlan
NSERR_VLAN_SUBNET_ALREADY_BOUND	966	0x3c6	Subnet already bound to this vlan
NSERR_UNABLE_GET_VLAN_IP	967	0x3c7	Unable to get vlan of the IPaddress
NSERR_IP_VLAN_BOUND	968	0x3c8	Ipaddress already bound to a vlan
NSERR_VLAN_UNABLE_ADD	969	0x3c9	Unable to add vlan
NSERR_DEFVLAN	970	0x3ca	Default vlan cannot be modified
NSERR_IPV6_ROUTING_BRIDGEGRP	971	0x3cb	vlan is part of a bridge group. Cannot enable ipv6 routing
NSERR_VLAN_SDXMGMTVLAN_CLUSTER	972	0x3cc	Operation not permitted. vlan is SDX management vlan in a cluster
NSERR_ACL_INUSE_RNAT	973	0x3cd	This ACL is used for RNAT/LSN/FORWARDINGSESSION. Action should be ALLOW
NSERR_RNAT_CM_TCPPROXY	974	0x3ce	TCPPROXY should be disabled for connection failover to work properly.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SNMP ERRORS

May 20, 2016

SNMP ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INVALTHRESHOLD	3457	0xd81	Threshold value for this alarm is a percentage (1 - 100)
NSERR_WRONGTHRESHOLDS	3458	0xd82	Normal threshold must be lower than alarm threshold
NSERR_INVALID_SRCIP	3459	0xd83	Valid source IP: NSIP, SNIP, MIP. In cluster setup, valid source IP: NSIP, CLIP, Striped SNIP
NSERR_THRESHOLD_UNSETTABLE	3460	0xd84	Threshold value cannot be set/unset for this alarm
NSERR_TIME_UNSETTABLE	3461	0xd85	Time interval cannot be set/unset for this alarm
NSERR_WRONG_LOW_THRESHOLDS	3462	0xd86	Normal threshold must be higher than alarm threshold
NSERR_NOSNMPUSER	3463	0xd87	SNMP user doesn't exist
NSERR_WRONG_SNMPVERSION	3464	0xd88	Bind command allowed only for V3 traps
NSERR_V3TRAP_COMMUNITY	3465	0xd89	V3 traps cannot have community
NSERR_IPINUSE	3466	0xd8A	IP is used in SNMP trap configuration
NSERR_DEFAULTONLY	3467	0xd8B	Alarm not supported in non-default partition

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL ERRORS

Jul 07, 2017

SSL ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SSL_CERT	1536	0x600	Invalid certificate
NSERR_SSL_PKEY	1537	0x601	Invalid private key, or PEM pass phrase required for this private key
NSERR_SSL_NOMATCH	1538	0x602	Certificate and private key do not match
NSERR_SSL_CERTTYPE	1539	0x603	Invalid cetificate type
NSERR_SSL_NOCERT	1540	0x604	Certificate does not exist
NSERR_SSL_REFEXT	1541	0x605	Certificate is referenced by a CRL, OCSP responder, vserver, service, another certificate, or a policy expression using XML_ENCRYPT() or XML_DECRYPT()
NSERR_SSL_BIND	1542	0x606	Certificate binding does not exist
NSERR_SSL_LINK	1543	0x607	Certificate can't be linked to the same certificate
NSERR_SSL_NEED_SSLPROTO	1544	0x608	Object's protocol type is not SSL
NSERR_SSL_NOLINK	1545	0x609	Certificate does not have any CA link
NSERR_SSL_BINDOR	1546	0x60A	Current certificate replaces the previous binding
NSERR_SSL_NOSVRCERT	1547	0x60B	Certificate is not a server certificate
NSERR_SSL_ISSUBMIS	1548	0x60C	Issuer certificate mismatch
NSERR_SSL_CRL	1549	0x60D	Invalid CRL

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SSL_NOCRL	1550	0x60E	CRL does not exist
NSERR_SSL_DHCOUNT	1551	0x60F	DH Refresh count should be 0 or >=500
NSERR_SSL_SESSTO	1552	0x610	Session timeout should be > 0
NSERR_SSL_ERSACOUNT	1553	0x611	eRSA Refresh count should be 0 or >=500
NSERR_SSL_DH_SIZE	1554	0x612	DH params of size greater than 2048 bits not supported
NSERR_DHPATH	1555	0x613	DH file path mandatory if DH enabled
NSERR_CERTHEADER	1556	0x614	Cert header tag mandatory if cert enabled
NSERR_SESSHEADER	1557	0x615	Sess header tag mandatory if sess enabled
NSERR_CIPHER_PERM	1558	0x616	Default ciphers/aliases cannot be added or removed
NSERR_SSL_ERSADISABLED	1559	0x617	Setting eRSA count when eRSA is disabled
NSERR_SSL_DHDISABLED	1560	0x618	Setting DH count when DH is disabled
NSERR_SSL_SESSDISABLED	1561	0x619	Setting session timeout when session reuse is disabled
NSERR_SSL_PKEY_SIZE	1562	0x61A	Certificate with key size greater than RSA4096 or DSA2048 bits not supported
NSERR_SSL_NOT_APPLICABLE	1563	0x61B	Option is not applicable for this SSL protocol
NSERR_CERTDNHEADER	1564	0X61C	Cert DN tag mandatory if cert-DN enabled
NSERR_CERTISSUERHEADER	1565	0X61D	Cert issuer tag mandatory if cert-issuer enabled
NSERR_CIPHERHEADER	1566	0X61E	Cipher tag mandatory if cipher header enabled
NSERR_SSL_INTERNALERR	1567	0x61F	Internal Error
NSERR_SSL_NOCACERT	1568	0x620	cacert does not exists

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SSL_REFRESHDIS	1569	0x621	crl refresh disabled
NSERR_SSL_SVRPORTNEEDED	1570	0x622	Server/port information is needed for enabling auto refresh
NSERR_SSL_BASEOBJNEEDED	1571	0x623	baseDN is required for enabling auto refresh
NSERR_SSL_CIPHER_REDIRECT	1572	0x624	Send Internal Cipher mismatch error page
NSERR_SSL_NODSA	1573	0x625	Loading of certificate and key of type DSA(DSS) is not supported with FIPS
NSERR_SSL_FIPSREFEXT	1574	0x626	The FIPS key is referenced by a certificate
NSERR_SSL_NOFIPSKEY	1575	0x627	No such FIPS key
NSERR_NOFIPSCARD	1576	0x628	Operation not permitted - no FIPS card present in the system
NSERR_FIPSCARDNOTCONF	1577	0x629	Operation not permitted - FIPS card is not configured
NSERR_SSL_SSLV2_REDIRECT	1578	0x62A	Send Internal SSL protocol mismatch error page
NSERR_SSL_MODSIZE64	1579	0x62B	Modulus size in bytes should be multiple of 64
NSERR_SSL_NONFIPSKEY	1580	0x62C	Configuration of non-FIPS key on FIPS system not allowed
NSERR_NFIPS_FIPS_UPD	1581	0x62D	Cannot update a non-FIPS certificate with a FIPS certificate
NSERR_FIPS_NFIPS_UPD	1582	0x62E	Cannot update a FIPS certificate with a non-FIPS certificate
NSERR_SSL_ISSUER_NOTIN_GLBCERTLIST	1583	0x62F	Unable to find the CA certificate for the CRL
NSERR_SSL_CRLSIGCHECK_FAIL	1584	0x630	Signature check on the CRL failed
NSERR_SSL_PORTREWRITE	1585	0x631	SSL port rewrite can be enabled only when SSL redirect is enabled
NSERR_SSL_SSLV2_RENEG_CLIENT_CERT	1586	0x632	Send Internal error page for SSLv2 protocol and client authentication with session renegotiation

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			during the update operation
Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SSL_CERT_NOT_YET_VALID	1588	0x634	The specified certificate is not yet valid
NSERR_SSL_CERT_EXPIRED	1589	0x635	The certificate has expired
NSERR_SSL_EXPIRED_BRKLINK	1590	0x636	All incompatible CA links were broken during the update operation. (Note: the certificate has expired)
NSERR_SSL_NYVALID_BRKLINK	1591	0x637	All incompatible CA links were broken during the update operation. (Note: the certificate is not yet valid)
NSERR_SSL_OCSP_RESPCERT	1592	0x638	No such responder certificate.
NSERR_SSL_OCSP_SIGNCERT	1593	0x639	No such signing certificate.
NSERR_SSL_OCSP_AIA	1600	0x640	Invalid AIA value

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSL2 ERRORS

Jul 07, 2017

SSL2 ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SSL_FIPSCARDLOCKED	3584	0xe00	FIPS card locked due to three unsuccessful login attempts
NSERR_SSL_DOMINCOMPAT	3585	0xe01	Certificate is registered to a different domain; use the 'no domain check' option to force the operation
NSERR_SSL_NOMTHDCHANGE	3586	0xe02	Cannot change refresh method of CRL
NSERR_SSL_URLSRVRNEEDED	3587	0xe03	Either URL or server-IP required on CRL
NSERR_SSL_INVALID_URL	3588	0xe04	Invalid URL
NSERR_SSL_MIXPARAMS	3589	0xe05	LDAP and HTTP parameters cannot both be specified
NSERR_SSL_PKEY_MINSIZE	3590	0xe06	Certificate of size smaller than 512 bits not supported
NSERR_SSL_SYNCINPROGRESS	3591	0xe07	Another synchronization is already in process, please try again later
NSERR_SSL_SYNCFAILED	3592	0xe08	Synchronization failed, please try again
NSERR_SSL_CIPHGRP_REFCNT	3593	0xe09	Cipher group has bound entities or is referenced by an SSL vserver or SSL service
NSERR_SSL_CVM_NODSA	3594	0xe0A	Loading DSA(DSS) certificate and key not supported on this platform
NSERR_CERTHASHHEADER	3595	0xe0B	CertHash header tag mandatory if certHash is enabled
NSERR_SSL_CRLINREFRESH	3596	0xe0C	CRL refresh in progress, details cannot be displayed

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			any more CRL's
NSERR_SSL_CRLINDELETION	3598	0xe0E	CRL deletion in progress, details cannot be displayed
NSERR_SSL_INDELETE_NOREFRESH	3599	0xe0F	CRL deletion in progress, cannot be refreshed
NSERR_SSL_INREFRESH_NODELETE	3600	0xe10	CRL refresh in progress, cannot be removed
NSERR_NOMIX	3601	0xe11	Cannot specify both HTTP data insertion and SSL actions
NSERR_NOPOLICY_NONTRSVC	3602	0xe12	Cannot bind SSL policy to SSL backend service
NSERR_SSL_SSLPOL_BIND_CONST	3603	0xe13	Cannot bind non-SSL policy to SSL vserver/service
NSERR_SSL_NO_USABLE_CIPHERS	3604	0xe14	No usable ciphers configured on the SSL vserver/service
NSERR_SSL_CERT_NOT_CA	3605	0xe15	Not a CA certificate
NSERR_SSL_CACERT_NO_CRLSIGN	3606	0xe16	Specified certificate is either not a CA cert, or does not have privileges to issue CRLs
NSERR_SSL_CRL_EXPIRED	3607	0xe17	CRL has expired
NSERR_SSL_CRL_NOTYET_VALID	3608	0xe18	CRL is not yet valid
NSERR_SSL_PARSING_DELTA_CRL_EXTN	3609	0xe19	Parsing of Delta-CRL extension failed
NSERR_SSL_DELTA_CRL_MISSING_BASE_CRL	3610	0xe1A	Base-CRL for the specified Delta-CRL is missing
NSERR_NOFIPSCIPHER	3611	0xe1B	Specified cipher/cipher-alias is not FIPS-approved
NSERR_NOFIPSCIPHERGRP	3612	0xe1C	Cipher group does not contain all FIPS-approved ciphers
NSERR_NONFIPSCIPHERTOGRP	3613	0xe1D	Cannot add non FIPS approved cipher to cipher group
NSERR_NONFIPSALIASTOGRP	3614	0xe1E	Cannot add non FIPS approved cipher alias to cipher group

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			another cipher group
NSERR_SSL_IMPORT_FIPSKEY_NAME_MISMATCH	3616	0xe20	Specified FIPS key name does not match with the exported FIPS key name
NSERR_SSL_PKEY_SIZE_CA	3617	0xe21	CA certificate of size greater than 4096 bits not supported
NSERR_SSL_CRL_PORT_MISMATCH	3618	0xe22	Port specified in URL does not match - port parameter
NSERR_SSL_PKEY_SIZE_VPX	3619	0xe23	Certificate with key size greater than RSA512 or DSA512 bits not supported
NSERR_SSL_DH_SIZE_VPX	3620	0xe24	DH params of size greater than 512 bits not supported
NSERR_NOENT_CIPHER	3621	0xe25	No such cipher/cipherAlias/cipherGroup
NSERR_FIPSFWWRONGMAJOR	3622	0xe26	The current firmware's major version is not supported for update operation
NSERR_FIPSFWWRONGMINOR	3623	0xe27	The current firmware's minor version is not supported for update operation
NSERR_FIPSFWUPDATED	3624	0xe28	The current firmware is already updated to 4.6.1
NSERR_SSL_PENDING_CMDS	3625	0xe29	Other commands (card health monitoring/traffic) are pending to FIPS card. Please try the firmware update command after some time
NSERR_FIPSFWUPDATEDOREBOOT	3626	0xe2A	Operation not permitted - FIPS card firmware update done, please reboot the system
NSERR_SSL_SNI_NOTENABLE	3627	0xe2B	SNI feature not enabled on the vserver/service
NSERR_SSL_NO_CN	3628	0xe2C	CommonName not present in certificate, it is must for a certificate to use for SNI
NSERR_SSL_DUP_SNICERT	3629	0xe2D	Trying to bind SNI certificate with duplicate CommonName, operation failed
NSERR_SSL_SNI_NOTVALID_SERV	3630	0xe2E	Operation not permitted - SNI feature not supported on SSL backend/internal service

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			vserver/certkey pair. Use unbind ssl certkey.
NSERR_OCSP_SIGNER_NOKEY	3633	0xe31	Signing certificate must also have a private key.
NSERR_SSL_NOT_SUPPORTED	3634	0xe32	OCSP responder must be an HTTP server; SSL is not supported.
NSERR_OCSP_TOO_MANY_RESPONDERS	3635	0xe33	Too many OCSP responders configured to add another. Please delete some and try again.
NSERR_OCSP_NO_DNS_SERVER_CONFIGURED	3636	0xe34	Unable to resolve DNS name.
NSERR_SSL_DUP_SNICERT_BRKLINK	3637	0xe35	Some of the existing SNI cert bindings are broken due to presence of certificate with duplicate Common Name.
NSERR_SSL_NO_CN_BRLLINK	3638	0xe36	All exitsing SNI Cert bindings are broken during update operation due to missing Common Name.
NSERR_NGFIPSRESETREBOOT	3639	0xe37	Operation not permitted - FIPS card was reset, please reboot the system
NSERR_NGFIPSINITREBOOT	3640	0xe38	Operation not permitted - FIPS card was initialized, please reboot the system
NSERR_FIPSCMDTIMEOUT	3641	0xe39	Operation timedout on the FIPS card, please try again
NSERR_SSL_SIMTIMEOUT	3642	0xe3a	Operation timed out or repeated, please wait for 10 mins and redo the SIM/HA configuration steps.
NSERR_SSL_NGFIPS_QFULL	3643	0xe3b	FIPS card command queue full. Please try again later
NSERR_SSL_NOMEM_VSVRSRVLISTNODE	3644	0xe3c	Failed to allocate memory for CertkeyVserverServList Node.
NSERR_SSL_CERTKEY_SIZE64	3645	0xe3d	Certificate with key size (modulus) that is not multiple of 512 bits is not supported
NSERR_SNI_ATK	3646	0xe3e	Host header field in the HTTP request does not match with the SNI domain name
NSERR_SNI_NOHOSTHDR	3647	0xe3f	Host header missing in the HTTP header for SNI enabled session

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_CRL_SHMEM_ALLOC_FAIL	3648	0xe40	failed
NSERR_SSL_PE_NO_SET_UNSET	3649	0xe41	Set/Unset operation not supported for classic SSL polices
NSERR_SSL_NOT_CTL_POL	3650	0xe42	Not a control policy
NSERR_SSL_NOT_DATA_POL	3651	0xe43	Not a data policy
NSERR_SSL_TYPE_REQD	3652	0xe44	Type is required
NSERR_SSL_NOT_PI_POL	3653	0xe45	Advanced policy attributes specified with classic policy
NSERR_SSL_PE_POL_POL_LBL	3654	0xe46	Cannot bind classic policy to policy label
NSERR_SSL_CERT_MISSING_PARAM	3655	0xe47	Required parameters missing in the certificate. Please check the certificate for completeness
NSERR_SSL_NOMEM_CERTKEY_OCSPRESP_LISTNODE	3656	0xe48	Failed to allocate memory for CertkeyOCSPRespList Node.
NSERR_SSL_OCSP_DUPLICATE	3657	0xe49	An ocspResponder is already bound with the specified priority.
NSERR_SSL_BIND_CP	3658	0xe4a	Binding of SSL classic policy is not supported, use equivalent advanced SSL policy
NSERR_SSL_ADD_CP	3659	0xe4b	Adding SSL classic policy is not supported, use equivalent advanced rule
NSERR_SSL_BUNDLE_IC_FILE_EXISTS	3660	0xe4c	Certificate file for intermediate certificate already exists.
NSERR_SSL_BUNDLE_SCERT_MISSING	3661	0xe4d	Server certificate must be placed first in certificate bundle file.
NSERR_SSL_BUNDLE_CERT_MISSING	3662	0xe4e	No certificates present in the certificate bundle file.
NSERR_SSL_BUNDLE_FAILED	3663	0xe4f	Processing of certificate bundle file failed.
NSERR_SSL_BUNDLE_PARSE_ERR	3664	0xe50	Unable to parse the certificate bundle file.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			certificates limit of 9.
NSERR_SSL_BUNDLE_MAX_KEY	3666	0xe52	Only one private-key is allowed in the certificate bundle file.
NSERR_SSL_BUNDLE_IC_FILE_CREATE_FAILED	3667	0xe53	Intermediate certificate file creation failed.
NSERR_SSL_SKIPCA_OPNOTPER	3668	0xe54	skipCA is not permitted for this entity.
NSERR_SSL_ISSUER_MISMATCH	3669	0xe55	Certificate Issuer mismatch
NSERR_SSL_SET_POLICY_ACTION_TYPE	3670	0xe56	Action type cannot be changed from the previous configured action type
NSERR_SSL_DTLS_NOTSUPP	3671	0xe57	Virtual server of type DTLS is not supported on this platform
NSERR_SSL_INVALID_CN_NAME	3672	0xe58	Invalid Common Name.
NSERR_SSL_ECC_NOT_SUPPORTED	3673	0xe59	ECDHE ciphers supported only on FE SSL entities on VPX, MPX and BE MPX
NSERR_SSL_NO_PROTOCOL_ENABLED	3674	0xe5a	SSL or TLS protocols not enabled on the service.
NSERR_DTLS_PROFILE_REFEXT	3676	0xe5c	DTLS profile is referenced by a vserver or front-end service.
NSERR_SSL_NO_ECC_CURVES	3677	0xe5d	No ECC curves bound for ECDHE ciphers.
NSERR_CRYPTODEV_MAX_LIMIT	3678	0xe5e	Crypto Device count exceeds maximum available.
NSERR_SSL_PROFILE_ATTACHED	3679	0xe5f	Operation not permitted. Use set ssl profile for setting these parameters.
NSERR_SSL_PROFILE_USED	3680	0xe60	Profile is being used by Virtual Server or Service.
NSERR_SSL_PROFILE_NOT_VAILD_PARAM	3681	0xe61	Specified parameters are not applicable for this type of SSL profile.
NSERR_IMPORT_SSL_INVALID_DHFILE_ERROR_OBJECT	3682	0xe62	Import failed:DH file being imported is invalid.
NSERR_IMPORT_SSL_INVALID_CRLFILE_ERROR_OBJECT	3683	0xe63	Import failed:CRL file being imported is invalid.
NSERR_IMPORT_SSL_INVALID_CERTFILE_ERROR_OBJECT	3684	0xe64	Import failed:Certificate file being

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_IMPORT_SSL_INVALID_KEYFILE_ERROR_OBJECT	3685	0xe65	Import failed:Key file being imported is invalid
NSERR_FIPSFW_FILEPATH	3686	0xe66	Invalid firmware file path
NSERR_FIPSFW_FILEOPEN	3687	0xe67	Unable to open firmware file
NSERR_FIPSFW_FILEIO	3688	0xe68	Read error occurred for firmware file
NSERR_FIPSFW_FSTAT	3689	0xe69	Failed to get status of firmware file
NSERR_FIPSFW_AUTH_FILEPATH	3690	0xe6a	Invalid firmware signature file path
NSERR_FIPSFW_AUTH_FILEOPEN	3691	0xe6b	Unable to open firmware signature file
NSERR_FIPSFW_AUTH_FILEIO	3692	0xe6c	Read error occurred for firmware signature file
NSERR_FIPSFW_AUTH_FSTAT	3693	0xe6d	Failed to get status of firmware signature file
NSERR_FIPSFW_BEGIN_ERROR	3694	0xe6e	Firmware update process failed BEGIN command
NSERR_FIPSFW_UPDATE_ERROR	3695	0xe6f	Firmware update process failed UPDATE command
NSERR_FIPSFW_END_ERROR	3696	0xe70	Firmware update process failed END command
NSERR_FIPSFW_STATE_ERROR	3697	0xe71	Firmware update process failed, invalid state
NSERR_FIPSFW_INVALID_CHUNK_TYPE	3698	0xe72	Firmware update process failed, invalid chunk type
NSERR_SSL_OCSP_WITH_CK_HSMKEY	3699	0xe73	Configuration of OCSP signing certificate with external HSM key not supported.
NSERR_SSL_HSMKEY_DEFLOCATION	3700	0xe74	Input HSM Key Simple file not present under the default directory /var/opt/nfast/kmdata/local/
NSERR_SSL_HSMKEY_IDENT_MISMATCH	3701	0xe75	HSM Key Name must be same as HSM Key Ident

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			supported.
NSERR_SSL_HSMKEY_BUNDLE	3703	0xe77	Addition of certificate-bundle with external HSM Key is not supported.
NSERR_CPE_EXPRESSION_PARTITION_INVALID	3704	0xe78	Classic expression is not supported in current partition.
NSERR_AP_INVALID_EXPRESSION	3705	0xe79	Expression value not supported.
NSERR_SSL_DH_BUSY	3706	0xe7a	DH Param in use. Please try later...
NSERR_FIPSFW22_MINKEYSIZE	3707	0xe7b	Key size less than 2048 is not supported on FIPS Firmware Version 2.2.
NSERR_SSL_MAXSAN	3708	0xe7c	Maximum number of SANs reached
NSERR_SSL_BADSNICERT	3709	0xe7d	Bad SNI certificate
NSERR_SSL_AESGCM_SHA2_NOT_SUPPORTED	3710	0xe7e	AES-GCM/SHA2 ciphers not supported on VPX and FIPS
NSERR_SSL_NOCIPHERGRP	3711	0xe7f	Cipher Group does not exist.
NSERR_SSL_SVCVSR_NOT_FOUND	3712	0xe80	No SSL vserver/service found with this name.
NSERR_SSL_DEF_DTLS_PROFILE	3713	0xe81	Cannot modify default DTLS profile.
NSERR_SSL_OP_ON_FIPS_NOT_PERM	3714	0xe82	The nonFipsCiphers option is not applicable any more
NSERR_SSL_OP_ON_NONFIPS_NOT_PERM	3715	0xe83	Operation permitted only on MPX-FIPS platform.
NSERR_SSL_LOAD_KEYS	3716	0xe84	Loading of internal keys failed.
NSERR_SSL_PASSWORD_DECODE	3717	0xe85	Decode of the encrypted passphrase failed.
NSERR_SSL_NOCERT_FILE	3718	0xe86	Certificate file name not specified.
NSERR_SSL_DEFAULT_CERT_DEL	3719	0xe87	Cannot delete internal default certificate.
			FIPS certificate cannot be bound as a CA

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			the FIPS key.
NSERR_SSL_CERTKEY_HSM	3721	0xe89	Cannot bind a certificate with HSM key to a service.
NSERR_SSL_DEFAULT_CERT_BIND	3722	0xe8a	Cannot bind internal default certificate.
NSERR_SSL_CMD_DEPRECATED	3723	0xe8b	Command deprecated, operation not applicable to this platform.
NSERR_SSL_NO_LIC	3724	0xe8c	Only export ciphers are allowed without proper SSL license.
NSERR_SSL_DTLS_NA	3725	0xe8d	Command arguments does not apply to vserver/service of type DTLS.
NSERR_SSL_ECC_SUPPORT	3726	0xe8e	ECC curve is not supported for this entity/platform.
NSERR_SSL_PROFILE_FE	3727	0xe8f	Cannot bind front-end profile to a back-end SSL service.
NSERR_SSL_PROFILE_BE	3728	0xe90	Cannot bind back-end profile to a front-end SSL vserver.
NSERR_SSL_OP_ON_FIPS_WOFW22_NOT_PERM	3729	0xe91	Operation not permitted on MPX-FIPS platform without FIPS firmware version 2.2 or higher.
NSERR_SSL_NONSSL_VSERVER	3730	0xe92	Vserver is not of type SSL.
NSERR_SSL_UDP_DTLS	3731	0xe93	Cannot bind policy to vserver of type DTLS.
NSERR_SSL_TLS11_12_SUPPORT	3732	0xe94	Enabling of TLSv1.1/1.2 is not supported on this entity/platform.
NSERR_SSL_HSM_KEY_SVC_SVCGRP	3733	0xe95	Binding of certificate with HSM key is not supported on back-end service or serviceGroup.
NSERR_DTLS_PROFILE	3734	0xe96	Cannot bind DTLS profile to a SSL vserver.
NSERR_SSL_PROFILE_NO_USABLE_CIPHERS	3735	0xe97	No usable ciphers configured for some of the SSL vserver/service binded to this profile
NSERR_SSL_CIPHGRP_NO_USABLE_CIPHERS	3736	0xe98	No usable ciphers configured for some of the SSL vserver/service binded to this cipher group

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			allowed
NSERR_SSL_DEFAULT_PROFILE_ENABLED	3738	0xe9a	Disabling ssl default profile is not allowed
NSERR_SSL_PROFILE_DEFAULT_DISABLED	3739	0xe9b	Operation not permitted. To do this Enable default ssl profile by setting 'set ssl parameter -defaultProfile E'
NSERR_SSL_PROFILE_DEFAULT_ENABLED	3740	0xe9c	Operation not permitted. To do this Disable default ssl profile by setting 'set ssl parameter -defaultProfile D'
NSERR_SSL_CIPHER_EXIST_HIGHER_PRI	3741	0xe9d	Specified cipher is already bound with higher priority
NSERR_SSL_DEFAULT_PROFILE_PERM	3742	0xe9e	Default ssl profile can not be removed
NSERR_SSL_DH_SIZE_16B	3743	0xe9f	DH params of size is not 16B aligned
NSERR_SSL_DTLSPROFNOTSUPP	3754	0xeaa	Cannot set DTLS Profile Name for a ssl service.
NSERR_SSL_PUSHENCNOTSUPP	3755	0xeab	Push encryption trigger is not supported for SSL Service.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SSLVPN AAA ERRORS

Jul 07, 2017

SSLVPN AAA ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_ERR_AAA_LICENSE	2624	0xa40	MaxAAAUsers value not allowed by license
NSERR_USR_NOINTRAIP	2625	0xa41	No intranet IP bound to user
NSERR_USR_NOTCONFIGURED	2626	0xa42	User does not exist
NSERR_INVAL_AAA_GRP	2627	0xa43	Group does not exist
NSERR_INVAL_COMBNATION	2628	0xa44	NONE and ALL cannot be used simultaneously
NSERR_INVAL_MIP_IIP	2629	0xa45	Both Mapped IP and Intranet IP must be specified
NSERR_INVAL_MIPOFF_IIPOFF	2630	0xa46	Mapped IP and Intranet IP cannot be OFF at the same time
NSERR_USEREXIST	2631	0xa47	User already exists
NSERR_GROUPEXIST	2632	0xa48	Group already exists
NSERR_USERALREADYBOUND	2633	0xa49	User is already bound to the group
NSERR_USERNOTBOUND	2640	0xa50	User is not bound to the group
NSERR_ENTITYNOTBOUND	2641	0xa51	Entity not bound
NSERR_GROUPNOTEXIST	2642	0xa52	Group does not exist
NSERR_INVALIDLOGLEVEL	2643	0xa53	NONE cannot be combined with other options

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			enabled on the same vserver
NSERR_DH_IPPORT	2645	0xa55	The SG second hop ip and port are required
NSERR_DHINUSE	2646	0xa56	Bound double hop server cannot be removed
NSERR_INVAL_AAAGLOBAL_POLTYPE	2647	0xa57	Only preauthentication policies can be bound to AAA global
NSERR_NO_IP	2648	0xa58	ServerIP not configured for the authentication type
NSERR_AAATM_LIC	2649	0xa59	Feature not licensed [AAA]
NSERR_AAATM_DISABLED	2650	0xa5a	Feature(s) not enabled [AAA]
NSERR_NO_AUTH_HOST	2651	0xa5b	No Authentication Host specified
NSERR_AUTH_ON	2652	0xa5c	Turn authentication off first
NSERR_KILLPENDING	2653	0xa5d	Another kill command in progress
NSERR_AAATM_NO_AUTH_VS	2654	0xa5e	No Authentication vserver name specified or the server does not exist
NSERR_AAATM_401AUTH_ON	2655	0xa5f	Turn off 401 based authentication first
NSERR_UNAUTHRZED	2656	0xa60	Unauthorized
NSERR_KILL_INPROGRESS	2657	0xa61	A kill session command is in progress. Try again later
NSERR_WI_FRM_NOTEXIST	2658	0xa62	Farm does not exist
NSERR_WI_FRM_LAST	2659	0xa63	At least one Farm should be configured: can not remove last farm
NSERR_WI_NOTINST	2660	0xa64	Web interface not installed
NSERR_WI_GENFAILED	2661	0xa65	Unable to generate website
NSERR_WI_SITE_EXIST	2662	0xa66	Site already exists

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_WI_SITE_NOTEXIST	2663	0xa67	Site does not exist
NSERR_WI_SITE_INVAL_AGURL	2664	0xa68	Invalid agURL
NSERR_WI_SITE_INVAL_STAURL	2665	0xa69	Invalid staURL
NSERR_WI_SITE_ONLY_MPX	2666	0xa6a	WI can only be installed on NetScaler nCore builds
NSERR_WI_INSTFAILED	2667	0xa6b	Installation failed. Please check the log file /var/log/wicmd.log
NSERR_WI_MAXSITE_EXCD	2668	0xa6c	Maximum number of WI Sites exceeded. Please check the log file /var/log/wicmd.log
NSERR_WI_STAWITHOUTAGURL	2669	0xa6d	STAUrl cannot be specified without AGUrl.
NSERR_WI_AGURLWITHOUTSTA	2670	0xa6e	AGUrl cannot be specified without STAUrl.
NSERR_WI_RELWITHOUTAGURL	2671	0xa6f	SessionReliability cannot be ON without AGUrl.
NSERR_WI_AUTHWITHOUTAGURL	2672	0xa70	AuthenticationPoint cannot be specified without AGUrl.
NSERR_WI_TWOTKTWITHOUTREL	2673	0xa71	UseTwoTickets cannot be ON without SessionReliability.
NSERR_WI_TWOTKTWITHOUTSECSTA	2674	0xa72	UseTwoTickets cannot be ON without SecondSTAUrl.
NSERR_WI_SECSTASAME	2675	0xa73	SecondSTAUrl should be different from StaURL.
NSERR_WI_LICENSE	2676	0xa74	Web Interface on NS Feature not licensed.
NSERR_WI_SECSTAWITHOUTSTA	2677	0xa75	SecondSTAUrl cannot be specified without STAUrl.
NSERR_AUTH_NEGOTIATE	2678	0xa76	Negotiate authentication required
NSERR_WI_INSTSITESREDUCED	2679	0xa77	Memory available is not sufficient for the passed maxSites value
NSERR_WI_INCOMPATIBLEAUTHPOINT	2680	0xa78	WI Authentication methods can not be specified unless authentication point is WI

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			another WI site
NSERR_WI_SITE_TRANSLATION_NOEXIST	2682	0xa7a	Translation Map does not exist
NSERR_WI_SITE_TRANSLATION_LAST	2683	0xa7b	Can't Unbind Last Translation Map
NSERR_WI_SITE_NOAGURL	2684	0xa7c	AGURL is not set
NSERR_AGSVC_AUTHFAIL	2685	0xa7d	AG-Service request with invalid AAAC cookie
NSERR_TM_INVALID_PERS_CONFIG	2686	0xa7e	Please specify both persistentCookie and persistentCookieValidity parameters
NSERR_WI_SMARTCARD_NOT_SUPPORTED	2687	0xa7f	SmartCard AGAuthentication method is not supported with sites created using wi package
NSERR_WI_SEARCH_NOT_SITEWEB	2688	0xa80	Show Search Attribute supported only for site type XenAppWeb
NSERR_KCD_ACCOUNT_NOT_EXISTS	2689	0xa81	The KCD account does not exist
NSERR_KCD_KEYTAB_NOT_EXISTS	2690	0xa82	The keytab file does not exist
NSERR_KCD_VS_PRINCIPLE_NOT_FOUND	2691	0xa83	The Vserver principle not found
NSERR_KCD_REM_INUSE	2692	0xa84	KCD account is in use and cannot be removed
NSERR_KCD_TOO_MANY_VS_PRINCIPLE	2693	0xa85	Too many host principles
NSERR_WI_REFRESH_NOT_SITEWEB	2694	0xa86	Show Refresh Attribute supported only for site type XenAppWeb
NSERR_WI_UIMODE_NOT_SITEWEB	2695	0xa87	User Interface Mode Attribute supported only for site type XenAppWeb
NSERR_WI_UILAYOUT_NOT_SITEWEB	2696	0xa88	User Interface Layout Attribute supported only for site type XenAppWeb
NSERR_WI_MESSAGESTRS_NOT_SITEWEB	2697	0xa89	App Welcome Message, Welcome Message, Footer Text, Login Sys Message, Pre Login Button, Pre Login Message, Pre Login Title Attributes supported only for site type XenAppWeb
			Please turn ON authentication or authn401

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_BIND_NEGPARAM_AAAGLOBAL	2705	0xa91	Negotiate params is already bound. Please unbind existing configuration
NSERR_NO_PRODUCTION_FARM	2706	0xa92	WI site will not be fuctional till a production farm is bound
NSERR_SAML_FORM_CONFLICT	2707	0xa93	Traffic cannot have both SAML SSO and Form SSO on same action
NSERR_MAXLOGIN_FAILLOGIN_CONFLICT	2708	0xa94	Please specify both max login attempts and fail login timeout
NSERR_KCD_ACCOUNT_EXIST	2709	0xa95	The KCD account does not exist
NSERR_KCD_ACCOUNT_NOTCONFIGURED	2710	0xa96	The KCD account is not configured
NSERR_NEGACTION_KEYTAB_CONFLICT	2711	0xa97	Action should have either keytab file or both username and password
NSERR_KEYTAB_INVALID_ENC	2712	0xa98	Unsupported encryption type in keytab file
NSERR_HTTP_SSL_PROXY_CONFLICT	2713	0xa99	Cannot configure both HTTP and SSL proxy
NSERR_INVALIDIP_PORT	2714	0xa9a	Invalid IP or Port
NSERR_ADVEPA_NOTSUPP	2715	0xa9b	VServer does not support advanced EPA scans
NSERR_ADVEPA_NOPROFILE	2716	0xa9c	EPA profile with this name does not exist
NSERR_ADVEPA_INUSE	2717	0xa9d	Bound EPA profile cannot be removed.
NSERR_ADVEPA_EXISTS	2718	0xa9e	EPA profile exists.
NSERR_ADVEPA_BOUND	2719	0xa9f	EPA profile already bound to vserver.
NSERR_ADVEPA_NOTBOUND	2720	0xaa0	EPA profile not bound to vserver.
NSERR_ADVEPA_ON	2721	0xaa1	Advanced EPA is ON, can not bind classic preauthentication policy.
			There has been a design change in the support of OPSWAT specific EPA scans. EPA Profile Configuration is no longer needed.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_DEVICEPROFILE_INTERNAL_ERR	2723	0xaa3	An Internal Error occurred while operating on the Device Profile, please contact your Admin
NSERR_AAA_SESS_INVALID_INDEX	2724	0xaa4	Invalid attribute index. Attribute index should be between 1 and 16 both included
NSERR_AAATM_FORCETIMEOUT	2725	0xaa5	Force timeout setting of START or RESET must accompany a desired time in minutes
NSERR_KCDACCOUNT_KEYTAB	2726	0xaa6	Keytab file cannot be specified along with other parameters in kcdaccount
NSERR_KCDACCOUNT_REALM	2727	0xaa7	Realm must be specified by itself or extracted from delegated username/keytab
NSERR_WEBAUTH_FULLEXP	2728	0xaa8	Please specify full request expression
NSERR_WI_INVALID_SETTING_FOR_AUTHPOINT	2729	0xaa9	Setting is Invalid for given Authpoint
NSERR_WI_SETTING_VALIDONLYFOR_XENAPPWEB	2730	0xaaa	Setting is Invalid for SiteType other than XenAppWeb
NSERR_WI_PARAMETER_UPGARDE_ERROR	2731	0xaab	Migration of WebInterface.conf parameter to CLI didn't happen successfully
NSERR_SAMLIDP_ENC_CERT	2732	0xaac	Encryption option requires ServiceProvider (SP) certificate.
NSERR_WF_NOT_INST	2733	0xaad	Web Front is not installed
NSERR_WF_INSTFAILED	2734	0xaae	Installation failed. Please check the log file /var/log/wfcmd.log
NSERR_AAA_TOOMANY_EXP	2735	0xaaf	Number of expressions found exceeds allowed limit
NSERR_SAML_REDIR_CONFLICT	2736	0xab0	Signature and Digest algorithms are incompatible for Redirect Binding
NSERR_SET_MAXAAAUSER_NOT_ALLOWED	2737	0xab1	Set maxaaaUser not allowed per partition
NSERR_AAATM_INCOMP_NO_PROFILE	2744	0xab8	Profile does not exist or does not have Host configured

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SST ERRORS

May 20, 2016

SST ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_SST_SUBSCR_ALLOC_FAILED	3504	0xdb0	Cannot allocate memory for subscriber in extended memory space
NSERR_SST_SUBSCR_LOOKUP_FAILED	3505	0xdb1	Cannot find subscriber
NSERR_SST_SUBSCR_REQUEST_ERROR	3506	0xdb2	Invalid request for subscriber
NSERR_GX_SERVICE_CONFIG_XOR	3507	0xdb3	service and vserver cannot be set at the same time.
NSERR_GX_NOT_DIAMETER_VS	3508	0xdb4	Not a DIAMETER or SSL_DIAMETER vserver.
NSERR_GX_NOT_DIAMETER_SERVICE	3509	0xdb5	Not a DIAMETER or SSL_DIAMETER service.
NSERR_SST_SUBSCRIBER_ENTRY_EXCEEDED	3510	0xdb6	Size of subscriber entry exceeds total allowed size.
NSERR_GX_PCRF_NOT_CONFIGURED_VSERVER_SERVICE	3511	0xdb7	PCRF not configured. To configure PCRF, please use set gxInteface -vserver/service
NSERR_SST_STATIC_SUBSCRIBER_SESSION_DELETION	3512	0xdb8	Static subscriber profiles cannot be deleted using this command, please use 'rm subscriber profile'
NSERR_SST_SUBSCRIBER_SESSION_EXIST	3513	0xdb9	subscriber profile cannot be added as corresponding subscriber session already exists
NSERR_SST_NOT_RADIUS_SERVICE	3514	0xdba	Not RADIUS Listener service
NSERR_MAX_SERVICE_FUNCTIONS	3515	0xdbb	Number for service functions exceeding the limit
NSERR_NOSUCH_SVCFUNC	3516	0xdbc	Service Function is not present with specified name

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_NOSUCH_SVCPATH	3517	0xdbd	name
NSERR_INVALID_SVCINDEX	3518	0xdbe	The value of service index is invalid
NSERR_SVCPATH_SVCINX_EXISTS	3519	0xdbf	A service function is already bound with given service index

© 1999-2017 Citrix Systems, Inc. All rights reserved.

SVPN ERRORS

May 20, 2016

SVPN ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSCFG_INFO	2049	0x801	Sending the /cfg information
NSCS_INFO	2050	0x802	Backend server info exists
NSCSPROBE_INFO	2051	0x803	Probe to backend server pending
NSAPPPROBE_INFO	2052	0x804	Client side connection being closed
NSCFG_MP_INFO	2112	0x840	Sending the /mp_cfg Information
NSERR_NOINTRANETIP	2113	0x841	No Intranet IP available
NSERR_ALREADYLOGEDIN	2114	0x842	The user is already logged-in
NSERR_URLINUSE	2115	0x843	Bound URL/bookmark cannot be removed.
NSERR_VPNAPPINUSE	2116	0x844	Bound VPN application cannot be removed.
NSERR_NOTSUPP_TRANS_INTERCPT	2117	0x845	Transparent interception is not yet supported.
NSERR_CLNT_CERT_RENEG	2118	0x846	Client SSL certificate renegotiation failed
NSERR_DEFAULTCMDPLCY	2119	0x847	Default command policy cannot be removed
NSERR_CLNT_CERT_REQD	2120	0x848	Client SSL certificate is required
NSERR_INVAL_CERTFIELD	2121	0x849	Invalid certificate field

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
			ranges
NSERR_VPNAPP_PROXY_NETMASK	2065	0x811	Proxy interception does not support netmasks
NSERR_VPNAPP_PROXY_DSTPORT_RANGE	2066	0x812	Proxy interception does not support destination port ranges
NSERR_VPNAPP_PROXY_PROTOCOL	2067	0x813	Proxy interception supports only TCP
NSERR_VPNAPP_PROXY_HOSTNAME	2068	0x814	Proxy interception does not support hostname interception
NSERR_VPNAPP_TRANS_SRCIP	2069	0x815	Transparent interception does not support source IP
NSERR_VPNAPP_TRANS_SRCPORT	2070	0x816	Transparent interception does not support source port
NSERR_VPNAPP_NO_INTERCEPTION_TYPE	2071	0x817	Intranet application requires an interception type
NSERR_VPNAPP_CLIAPP_PORT	2072	0x818	Both client application name and destination port cannot be specified
NSERR_VPNAPP_CLIAPP_PROTO	2073	0x819	Protocol can not be specified when client application name is present
NSERR_VPNAPP_PROXY_CLIAPP	2074	0x81a	Proxy interception does not support client application based interception
NSERR_VPNAPP_MISSING_PROTO	2075	0x81b	Protocol must be specified
NSERR_VPNAPP_MISSING_ARG	2076	0x81c	One of destIP, IPRange or hostname has to be specified
NSERR_VPNAPP_TOO_MANY_ARG	2077	0x81d	At most one of destIP, IPRange or hostname may be specified
NSERR_FS_AUTHFAIL	2078	0x81e	?
NSERR_NSIPV6NOTPRESENT	2079	0x81f	No IPV6 Netscaler IP has been configured
NSERR_REMOVE_SESSION	2096	0x830	No IPV6 Netscaler IP has been configured
NSERR_STAWI_EXIST	3250	0xcb2	A STA or WIDBS configuration exists. Unset it first

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INVALID_TMTRAFFIC_ACTION	2100	0x834	Invalid traffic action
NSERR_INVALFSSO	2101	0x835	SSO should be turned on for setting formsso action
NSERR_INVALIDURL	2102	0x836	Action url should be root relative url
NSERR_INVALID_FLOWTYPE_FORAPPFLOW	2126	0x84E	Flowtype needs to be REQUEST or ICA_REQUEST or OTHERTCP_REQUEST for appflow policy when bound to VPN vservers
NSERR_TOO_MANY_SERVERS	2127	0x84F	Too many servers bound
NSERR_BLOCKING_ATTR_WARNING_FOR_VPN_APPFLOW	2961	0xb91	Policy will not apply to VPN traffic if it has blocking attributes
NSERR_SAMLIDP_ACTION	2975	0xb9f	Invalid SAML IDP Action
NSERR_SAML_ACTION_AUTHNCTX	2976	0xba0	None option is mutually exclusive with other Authentication Class References
NSERR_ABSOLUTE_URL	2977	0xba1	URL has to be an absolute URL
NSERR_INVALID_AG_BIND	2978	0xba2	SSLVPN vserver can be bound to only SSL type CS vserver
NSERR_NO_VSERVER	2979	0xba3	Please specify the associated vserver name
NSERR_SSSO_NOT_LICENSED	2980	0xba4	This feature needs enterprise or platinum license.
NSERR_MULT_AGBIND	2981	0xba5	Only one VPN vserver can be bound to a CS vserver.
NSERR_VPN_VSERV_PROTOCOL_NOT_ALLOWED	2982	0xba6	VPN Vserver of service type 'HTTP' is not allowed.
NSERR_SET_NOT_ALLOWED_ON_ICAONLY_BIT	2983	0xba7	Changing 'ICAonly' parameter is not allowed in Cluster Deployment.
NSERR_SAMLIDP_NO_ART_BINDING	2986	0xbaa	Artifact binding is not supported in SAML IDP

© 1999-2017 Citrix Systems, Inc. All rights reserved.

TD ERRORS

Jul 07, 2017

TD ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_TD_EXISTS	944	0x3b0	The specified traffic domain is already configured.
NSERR_TD_VLAN	945	0x3b1	The specified VLAN/VXLAN/bridge group is already bound to another traffic domain. These entities can belong to only one traffic domain.
NSERR_TD_NOTEXIST	946	0x3b2	The specified traffic domain is not configured.
NSERR_TD_MISMATCH_WITH_L3VLAN	947	0x3b3	L3 VLANs can not be bound to a traffic domain.
NSERR_TD_MISMATCH	948	0x3b4	The binding entities have incompatible traffic domain identifiers.
NSERR_TD_NOSUPP	949	0x3b5	This feature is not supported in a non-default traffic domain.
NSERR_TD_CONFIG_NOT_ALLOWED	950	0x3b6	This configuration option is not supported in a non-default traffic domain.
NSERR_TD_IP_NOTEXIST	951	0x3b7	The IP address is not configured for the specified traffic domain.
NSERR_TD_VLAN_NOTEXIST	952	0x3b8	The specified VLAN is not configured.
NSERR_TD_DEFVLAN	953	0x3b9	Default VLAN cannot be bound/unbound to/from traffic domain.
NSERR_TD_BDG_NOTEXIST	954	0x3ba	The specified bridge group is not configured.
NSERR_TD_VLAN_BDG_EXIST	955	0x3bb	Operation not permitted. The specified VLAN is already bound to a bridge group.
NSERR_TD_SERVER_MISMATCH	956	0x3bc	The specified virtual server already bound to a service in a different traffic domain.
			Not allowed for LLB: Services bound to the specified virtual

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_DUPIP	958	0x3be	IP address duplication with VMAC address detected. Please try a different IP address.
NSERR_VMAC_VLAN	959	0x3bf	Traffic domain can be either VLAN based or VMAC based.
NSERR_TD_SYNCVLAN	960	0x3c0	SYNC VLAN cannot be bound/unbound to/from traffic domain.
NSERR_TD_NOT_VLAN	961	0x3c1	The specified VLAN/VXLAN/bridge group is not bound to given traffic domain.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

URLTRANS ERRORS

May 20, 2016

URLTRANS ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_URLTRANS_MAX_ENTITIES	2960	0xb90	Number of URL Transformation entities exceeds limit
NSERR_URLTRANS_INVAL_PROFILE	2962	0xb92	Invalid URL Transformation profile
NSERR_URLTRANS_ACTION_INUSE	2963	0xb93	Action name is already in use
NSERR_URLTRANS_PRIORITY_EXISTS	2964	0xb94	An object with this priority already exists
NSERR_URLTRANS_REQ_PCRE_ERR	2965	0xb95	Invalid PCRE expression under 'reqUrlFrom'
NSERR_URLTRANS_RESP_PCRE_ERR	2966	0xb96	Invalid PCRE expression under 'resUrlFrom'
NSERR_URLTRANS_REQ_INTO_ERR	2967	0xb97	Invalid expression under 'reqUrlInto'
NSERR_URLTRANS_RESP_INTO_ERR	2968	0xb98	Invalid expression under 'resUrlInto'
NSERR_URLTRANS_COOKIE_INTO_ERR	2969	0xb99	Invalid expression under 'cookieDomainInto'
NSERR_URLTRANS_COOKIE_PCRE_ERR	2970	0xb9a	Invalid PCRE expression under 'cookieDomainFrom'
NSERR_URLTRANS_TOO_MANY_BACKREFS	2971	0xb9b	More than 5 back references in PCRE expression
NSERR_URLTRANS_MISSING_FROM	2972	0xb9c	Each 'Into' expression must have a corresponding 'From' pattern
NSERR_URLTRANS_NOT_SUPPORTED_VS	2973	0xb9d	Specified policy can be bound only to HTTP/SSL vserver
NSERR_URLTRANS_BAD_COMMENT	2974	0xb9e	Invalid comment

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VLAN ERRORS

May 20, 2016

VLAN ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INTERFACEBOUND	2080	0x820	Interface is already bound to this VLAN
NSERR_IFACE_NO_UNBIND	2081	0x821	Untagged interface cannot be removed from default VLAN. To remove, bind to other VLAN.
NSERR_IFACE_MAX_VLANS	2082	0x822	Maximum number of tagged VLANs bound to the interface exceeded or the binding of this VLAN is not allowed on the interface.
NSERR_VLAN_RTEXIST	2083	0x823	An existing route relies on the presence of this vlan
NSERR_VLAN_LLEXIST	2084	0x824	An existing link-local ip relies on the presence of this vlan
NSERR_BOUNDTONSVLAN	2085	0x825	Interface/Channel bound to NSVLAN

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VMAC ERRORS

May 20, 2016

VMAC ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_INTERFACE_BOUND	3329	0xd01	Interface already bound
NSERR_VRID_INTERFACE_NOT_BOUND	3330	0xd02	Interface not bound
NSERR_NO_SUCH_INTERFACE	3331	0xd03	No such interface
NSERR_DYNAMIC_ENTRY	3346	0xd12	Dynaimc Entry exists.
NSERR_PRIO_INTFVRID	3347	0xd13	Invalid Priority for interface bound VRID
NSERR_INTFVRID	3348	0xd14	Binding interface is not allowed for Dynamic or IP bound VRID
NSERR_NOTSTATIC_VRID	3349	0xd15	Not a Configured VRID
NSERR_IPVRID	3350	0xd16	IP binding is not allowed for Interface VRID
NSERR_IPVRID_BOUND	3351	0xd17	VRID already bound to this IP
NSERR_INTF_TRACKINTF_BOUND	3362	0xd22	Interfaces can either be tracked or bound to a vrID.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VPATH ERRORS

Jul 07, 2017

VPATH ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_VPATH_INCOMPATIBLE_IP	4032	0xfc0	Destination and Gateway ip should be of same type
NSERR_VPATH_GW_NA	4033	0xfc1	Gateway not applicable with this encap mode
NSERR_VPATH_LIB_NAVAIL	4048	0xfd0	vPath library not available in this appliance
NSERR_VPATH_LIB_INCOMPAT	4049	0xfd1	Incompatible vPath library
NSERR_VPATH_LIB_INVALID_ARG	4050	0xfd2	Error retrieving version information from vPath library
NSERR_VPATH_NOTSUPPORTED_ICMPERR	4051	0xfd3	Option supported only in vPath Environment

© 1999-2017 Citrix Systems, Inc. All rights reserved.

VXLAN ERRORS

Jul 07, 2017

VXLAN ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_VXLAN_INCOMP	2608	0xa30	Vlan extension and IP address - both cannot be configured on vxlan
NSERR_VXLAN_TUN_BOUND	2609	0xa31	Tunnel is already bound to this vxlan
NSERR_INV_MCAST	2610	0xa32	Another multicast group already configured for this vxlan
NSERR_TUN_NOT_BOUND	2611	0xa33	Tunnel not bound to this vxlan
NSERR_IP_NOT_BOUND	2612	0xa34	IP/IPv6 address not bound to this vxlan
NSERR_INV_TUN	2613	0xa35	Tunnel should be of VXLAN type
NSERR_NO_VXLAN	2614	0xa36	Vxlan is not configured
NSERR_VLAN_EXT	2615	0xa37	Vlan already extended by a different vxlan
NSERR_VXLAN_TD	2616	0xa38	Vxlan is bound to a traffic domain
NSERR_VXLAN_ROTUING	2617	0xa39	Vlan extension and routing - both cannot be configured on vxlan

© 1999-2017 Citrix Systems, Inc. All rights reserved.

WARNING ERRORS

Jul 07, 2017

WARNING ERRORS

Error Code	Error Code(Decimal)	Error Code(Hex)	Error Message
NSERR_AUTHORIZE		NS_WARNING | 0x001	?
NSERR_AUTHFAIL		NS_WARNING | 0x002	?
NSWAR_SSL_BINDOR		NS_WARNING | NSERR_SSL_BINDOR	Current certificate replaces the previous binding
NSWAR_SSL_BRKLINK		NS_WARNING | NSERR_SSL_BRKLINK	All incompatible CA links/Cert Bindings were broken during the update operation
NSWAR_SSL_CERT_NOT_YET_VALID		NS_WARNING | NSERR_SSL_CERT_NOT_YET_VALID	The specified certificate is not yet valid
NSWAR_SSL_CERT_EXPIRED		NS_WARNING | NSERR_SSL_CERT_EXPIRED	The certificate has expired
NSWAR_SSL_EXPIRED_BRKLINK		NS_WARNING | NSERR_SSL_EXPIRED_BRKLINK	All incompatible CA links were broken during the update operation. (Note: the certificate has expired)
NSWAR_SSL_NYVALID_BRKLINK		NS_WARNING | NSERR_SSL_NYVALID_BRKLINK	All incompatible CA links were broken during the update operation. (Note: the certificate is not yet valid)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Java, .NET, and Python API

Oct 14, 2015

This section provides basic information for using the Java, .NET, and Python SDKs that are provided for the NITRO API. The API are categorized on their scope and purpose.

Important

- All NITRO operations are logged in the /var/log/nitro.log file on the appliance.

- Executable samples are available in the *<NITRO_SDK_HOME>/sample* directory.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tutorial: Create Your First NITRO Application

Sep 12, 2012

After completing this tutorial, you will understand and be able to use NITRO to log in to the appliance, create a load balancing virtual server, retrieve details of an lbvserver, delete an lbvserver, save the configurations on the appliance, and log out of the appliance.

- Using Java API to Create your First NITRO Application
- Using .NET API to Create your First NITRO Application

> ## Note
>
> Before you begin, make sure that you have the latest NITRO SDK and that the client application satisfies the prerequisites for using the NITRO SDK.
>
> All NITRO exceptions are captured by the *com.citrix.netscaler.nitro.exception.nitro_exception* class. For a more detailed description, see Exception Handling.
>
> The executable code for the sample is available in the <NITRO_SDK_HOME>/sample/ directory.

Using Java API to Create your First NITRO Application

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.
2. Create a new class and name it **MyFirstNitroApplication**.
3. Create an instance of com.citrix.netscaler.nitro.service.nitro_service class. This instance is used to perform all operations on the appliance:
   nitro_service ns_session = new nitro_service("10.102.29.170","HTTP");
   This code establishes a connection with an appliance that has IP address 10.102.29.170 and uses the HTTP protocol. Replace 10.102.29.170 with the IP address of the NetScaler appliance that you have access to.
4. Use the nitro_service instance to log in to the appliance using your credentials:
   ns_session.login("admin","verysecret");
   This code logs into the appliance, with user name as admin and password as verysecret. Replace the credentials with your login credentials.
5. Enable the load balancing feature:
   String[] features_to_be_enabled = {"lb"};
   ns_session.enable_features(features_to_be_enabled);
   This code first sets the features to be enabled in an array and then enables the LB feature.
6. Create an instance of the com.citrix.netscaler.nitro.resource.config.lb.lbvserver class. You will use this instance to perform operations on the lbvserver.
   lbvserver new_lbvserver_obj = new lbvserver();
7. Use the lbvserver instance to create a new lbvserver:
   new_lbvserver_obj.set_name("MyFirstLbVServer");
   new_lbvserver_obj.set_ipv46("10.102.29.88");
   new_lbvserver_obj.set_servicetype("HTTP");
   new_lbvserver_obj.set_port(88);
   new_lbvserver_obj.set_lbmethod("ROUNDROBIN");

© 1999-2017 Citrix Systems, Inc. All rights reserved.

lbvserver.add(ns_session,new_lbvserver_obj);
This code first sets the attributes (name, IP address, service type, port, and load balancing method) of the lbvserver locally and then adds it to the appliance by using the corresponding add() method.

8. Retrieve the details of the lbvserver you have created:
   new_lbvserver_obj = lbvserver.get(ns_session,new_lbvserver_obj.get_name());
   System.out.println("Name : " +new_lbvserver_obj.get_name() +"\n" +"Protocol : " +new_lbvserver_obj.get_servicetype());
   This code first retrieves the details of the lbvserver as an object from the NetScaler, extracts the required attributes (name and service type) from the object, and displays the results.
9. Delete the lbvserver you created in the above steps:
   lbvserver.delete(ns_session, new_lbvserver_obj.get_name());
10. Save the configurations:
    ns_session.save_config();
11. Log out of the appliance:
    ns_session.logout();

Using .NET API to Create your First NITRO Application

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.
2. Create a new class and name it **MyFirstNitroApplication**.
3. Create an instance of com.citrix.netscaler.nitro.service.nitro_service class. This instance is used to perform all operations on the appliance:
   nitro_service ns_session = new nitro_service("10.102.29.170", "http");
   This code establishes a connection with an appliance that has IP address 10.102.29.170 and uses the HTTP protocol. Replace 10.102.29.170 with the IP address of the NetScaler appliance that you have access to.
4. Use the nitro_service instance to log in to the appliance using your credentials:
   ns_session.login("admin","verysecret");
   This code logs into the appliance, with user name as admin and password as verysecret. Replace the credentials with your login credentials.
5. Enable the load balancing feature:
   String[] features_to_be_enabled = {"lb"};
   ns_session.enable_features(features_to_be_enabled);
   This code enables load balancing on the appliance.
6. Create an instance of the com.citrix.netscaler.nitro.resource.config.lb.lbvserver class. You will use this instance to perform operations on the lbvserver.
   lbvserver new_lbvserver_obj = new lbvserver();
7. Use the lbvserver instance to create a new lbvserver:
   new_lbvserver_obj.name = "MyFirstLbVServer";
   new_lbvserver_obj.ipv46 = "10.102.29.88";
   new_lbvserver_obj.servicetype = "HTTP";
   new_lbvserver_obj.port = 80;
   new_lbvserver_obj.lbmethod = "ROUNDROBIN";
   lbvserver.add(ns_session,new_lbvserver_obj);
   This code first sets the attributes (name, IP address, service type, port, and load balancing method) of the lbvserver locally and then adds it to the appliance by using the corresponding add() method.
8. Retrieve the details of the lbvserver you have created:
   lbvserver new_lbvserver_obj1 = lbvserver.get(ns_session,new_lbvserver_obj.name);
   System.Console.Out.WriteLine("Name : " +new_lbvserver_obj1.name +"\n" +"Protocol : " +new_lbvserver_obj1.servicetype);
   This code first retrieves the details of the lbvserver as an object from the NetScaler, extracts the required attributes

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

(name and service type) from the object, and displays the results.

9. Delete the lbvserver you created in the above steps:
   lbvserver.delete(ns_session, new_lbvserver_obj.name);
10. Save the configurations:
    ns_session.save_config();
11. Log out of the appliance:
    ns_session.logout();

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tutorial: Create a NetScaler Cluster

Nov 05, 2013

This tutorial gives you the step-by-step process to create a NetScaler cluster. After completing this tutorial you will be able to create a two-node NetScaler cluster. To add more appliances to the cluster you must repeat the procedure that adds and joins the node to the cluster.

- Using Java API to Create a Cluster
- Using .NET API to Create a Cluster

> **Note**
>
> The executable code for the sample is available in the *<NITRO_SDK_HOME>/sample/* directory.

Using Java API to Create a Cluster

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.
2. Create a new class and name it CreateCluster.
3. Log on to one of the appliances that you want to add to the cluster and create a cluster:

```
//Connect to the first appliance that you want to add to the cluster
nitro_service nonClipSession0 = new nitro_service(nsipAddress0,protocol);
nonClipSession0.login(uName,password);

//Create a cluster instance
clusterinstance newClusterInstance = new clusterinstance();
newClusterInstance.set_clid(1);
clusterinstance.add(nonClipSession0,newClusterInstance);

//Add the appliance to the cluster
clusternode ClusterNode0 = new clusternode();
ClusterNode0.set_nodeid(0);
ClusterNode0.set_ipaddress(nsipAddress0);
ClusterNode0.set_state("ACTIVE");
ClusterNode0.set_backplane("0/1/1");
clusternode.add(nonClipSession0,ClusterNode0);

//Add the cluster IP address
nsip newNSIPAddress = new nsip();
newNSIPAddress.set_ipaddress(clipAddress);
newNSIPAddress.set_netmask("255.255.255.255");
newNSIPAddress.set_type("CLIP");
nsip.add(nonClipSession0,newNSIPAddress);

//Enable the cluster instance
clusterinstance.enable(nonClipSession0, newClusterInstance);

//Save the configurations
nonClipSession0.save_config();

//Warm reboot the appliance
nonClipSession0.reboot(true);
```

The cluster is created and the first node is added to the cluster. This node becomes the initial configuration coordinator of the cluster.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

4. Log on to the cluster IP address to add other appliances to the cluster:

```
//Connect to the cluster IP address
nitro_service clipSession = new nitro_service(clipAddress,protocol);
clipSession.login(uName,password);

//Add the node to the cluster
clusternode ClusterNode1 = new clusternode();
ClusterNode1.set_nodeid(1);
ClusterNode1.set_ipaddress(nsipAddress1);
ClusterNode1.set_state("ACTIVE");
ClusterNode1.set_backplane("1/1/1");
clusternode.add(clipSession,ClusterNode1);

//Save the configurations
clipSession.save_config();
```

5. Log on to the appliance that you added in the previous step and join it to the cluster:

```
//Connect to the node that you have just added to the cluster
nitro_service nonClipSession1 = new nitro_service(nsipAddress1,protocol);
nonClipSession1.login(uName,password);

//Join the node to the cluster
cluster newCluster = new cluster();
newCluster.set_clip(clipAddress);
newCluster.set_password(password);
cluster.join(nonClipSession1,newCluster);

//Save the configurations
nonClipSession1.save_config();

//Warm reboot the appliance
nonClipSession1.reboot(true);
```

The second node is now a part of the cluster.

6. Verify the details of the cluster by logging on to the cluster IP address

```
//Retrieving the cluster node details
Long id = new Long(1);
clusternode node= clusternode.get(clipSession, id);
System.out.println("Node ID: "+ node.get_nodeid() + " | Admin state: " + node.get_state() + " | Backplane interface: "+ node.get_backplane());

//Retrieving the cluster instance details
Long id1 = new Long(1);
clusterinstance instance= clusterinstance.get(clipSession, id1);
System.out.println("Cluster instance ID: "+ instance.get_clid() + " | Operational state: " +instance.get_operationalstate());
```

Using .NET API to Create a Cluster

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.
2. Create a new class and name it CreateCluster.
3. Log on to one of the appliances that you want to add to the cluster and create a cluster:

```
//Connect to the first appliance that you want to add to the cluster
nitro_service nonClipSession0 = new nitro_service(nsipAddress0,protocol);
nonClipSession0.login(uName,password);

//Create a cluster instance
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
clusterinstance newClusterInstance = new clusterinstance();
newClusterInstance.clid = 1;
clusterinstance.add(nonClipSession0,newClusterInstance);

//Add the appliance to the cluster
clusternode ClusterNode0 = new clusternode();
ClusterNode0.nodeid = 0;
ClusterNode0.ipaddress = nsipAddress0;
ClusterNode0.state = "ACTIVE";
ClusterNode0.backplane = "0/1/1";
clusternode.add(nonClipSession0,ClusterNode0);

//Add the cluster IP address
nsip newNSIPAddress = new nsip();
newNSIPAddress.ipaddress = clipAddress;
newNSIPAddress.netmask = "255.255.255.255";
newNSIPAddress.type = "CLIP";
nsip.add(nonClipSession0,newNSIPAddress);

//Enable the cluster instance
clusterinstance.enable(nonClipSession0, newClusterInstance);

//Save the configurations
nonClipSession0.save_config();

//Warm reboot the appliance
nonClipSession0.reboot(true);
```

The cluster is created and the first node is added to the cluster. This node becomes the initial configuration coordinator of the cluster.

4. Log on to the cluster IP address to add other appliances to the cluster:

```
//Connect to the cluster IP address
nitro_service clipSession = new nitro_service(clipAddress,protocol);
clipSession.login(uName,password);

//Add the node to the cluster
clusternode ClusterNode1 = new clusternode();
ClusterNode1.nodeid = 1;
ClusterNode1.ipaddress = nsipAddress1;
ClusterNode1.state = "ACTIVE";
ClusterNode1.backplane = "1/1/1";
clusternode.add(clipSession,ClusterNode1);

//Save the configurations
clipSession.save_config();
```

5. Log on to the appliance that you added in the previous step and join it to the cluster:

```
//Connect to the node that you have just added to the cluster
nitro_service nonClipSession1 = new nitro_service(nsipAddress1,protocol);
nonClipSession1.login(uName,password);

//Join the node to the cluster
cluster newCluster = new cluster();
newCluster.clip = clipAddress;
newCluster.password = password;
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
cluster.join(nonClipSession1,newCluster);

//Save the configurations
nonClipSession1.save_config();

//Warm reboot the appliance
nonClipSession1.reboot(true);
```

The second node is now a part of the cluster.

6. Verify the details of the cluster by logging on to the cluster IP address

```
//Retrieving the cluster node details
uint id = 1;
clusternode node= clusternode.get(clipSession, id);
System.Console.Out.WriteLine("Node ID: " + node.nodeid + " | Admin state: " + node.state + " | Backplane interface: " + node.backplane);

//Retrieving the cluster instance details
uint id1 = 1;
clusterinstance instance= clusterinstance.get(clipSession, id1);
System.Console.Out.WriteLine("Cluster instance ID: "+ instance.clid + " | Operational state: " +instance.operationalstate);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Connecting to the NetScaler Appliance

Oct 16, 2015

The first step towards using NITRO is to establish a session with the NetScaler appliance and then authenticate the session by using the NetScaler administrator's credentials.

You must create an object of the *com.citrix.netscaler.nitro.service.nitro_service* class by specifying the NetScaler IP (NSIP) address and the protocol to connect to the appliance (HTTP or HTTPS). You then use this object and log on to the appliance by specifying the user name and the password of the NetScaler administrator.

Note:

- For the python SDK, the package path is of the form *nssrc.com.citrix.netscaler...*
- You must have a user account on that appliance. The configuration operations that you perform are limited by the administrative roles assigned to your account.

The following sample code establishes a session with a NetScaler appliance with IP address 10.102.29.60 by using the HTTPS protocol and also sets a session timeout period (in seconds) of 60 minutes.

Java - Sample code to establish session COPY

```
//Specify the NetScaler appliance IP address and protocol

nitro_service ns_session = new nitro_service("10.102.29.60","https");



//Specify the login credentials

ns_session.login("admin","verysecret",3600);
```

.NET - Sample code to establish session COPY

```
//Specify the NetScaler appliance IP address and protocol

nitro_service ns_session = new nitro_service("10.102.29.60","https");



//Specify the login credentials

ns_session.login("admin","verysecret",3600);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Python - Sample code to establish session

COPY

```
#Specify the NetScaler appliance IP address and protocol

ns_session = nitro_service("10.102.29.60","https")



#Specify the login credentials

ns_session.login("admin","verysecret",3600)
```

Disable SSL Checks

When using HTTPS, you must make sure that the root CA is added to the truststore. By default, NITRO validates the SSL certificate and verifies the hostname. Disable these validations as shown in the following sample codes.

Java - Sample code for disabling SSL checks

COPY

```
ns_session.set_certvalidation(false);

ns_session.set_hostnameverification(false);
```

.NET - Sample code for disabling SSL checks

COPY

```
ns_session.certvalidation = false;

ns_session.hostnameverification = false;
```

Python - Sample code for disabling SSL checks

COPY

```
ns_session.certvalidation = false

ns_session.hostnameverification = false
```

Some points to note with regards to session timeout for NetScaler 10.5 and later versions:

© 1999-2017 Citrix Systems, Inc. All rights reserved.

- When restricted timeout param is enabled, NITRO, by default, uses the timeout value that is configured for the logged in user. You can customize this value but it must be limited to the value specified for the user. If no value is specified for the user, the default timeout value of 15 minutes is used.
- When restricted timeout param is not enabled, NITRO uses the default value of 30 minutes as session timeout.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

General API Usage

Sep 04, 2015

NetScaler resources are organized into a set of packages or namespaces. Each package or namespace corresponds to a NetScaler feature. For example, all load-balancing related resources, such as load balancing virtual server, load balancing group, and load balancing monitor are available in *com.citrix.netscaler.nitro.resource.config.lb*.

Note: For the python SDK, the package path is of the form *nssrc.com.citrix.netscaler.....*

Similarly, all application firewall related resources, such as application firewall policy and application firewall archive are available in *com.citrix.netscaler.nitro.resource.config.appfw*.

Each NetScaler resource is represented by a class. For example, the class that represents a load balancing virtual server is called **lbvserver** (in *com.citrix.netscaler.nitro.resource.config.lb*). The state of a resource is represented by properties of a class. You can get and set the properties of the class.

Note: The setter and getter properties are always executed locally on the client. They do not involve any network interaction with the NITRO web service. All properties have basic simple types: integer, long, boolean, and string.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Adding a NetScaler Resource

Oct 14, 2015

To create a new resource, instantiate the resource class, configure the resource by setting its properties locally, and then upload the new resource instance to the NetScaler appliance.

The following sample code creates a load balancing virtual server.

Java - Sample code to add a NetScaler resource COPY

```
//Create an instance of the lbvserver class

lbvserver new_lbvserver_obj = new lbvserver();



//Set the properties of the resource locally

new_lbvserver_obj.set_name("MyFirstLbVServer");

new_lbvserver_obj.set_ipv46("10.102.29.88");

new_lbvserver_obj.set_port(88);

new_lbvserver_obj.set_servicetype("HTTP");

new_lbvserver_obj.set_lbmethod("ROUNDROBIN");



//Upload the resource to NetScaler

lbvserver.add(ns_session,new_lbvserver_obj);
```

.NET - Sample code to add a NetScaler resource COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
//Create an instance of the lbvserver class

lbvserver new_lbvserver_obj = new lbvserver();



//Set the properties of the resource locally

new_lbvserver_obj.name = "MyFirstLbVServer";

new_lbvserver_obj.ipv46 = "10.102.29.88";

new_lbvserver_obj.port = 88;

new_lbvserver_obj.servicetype = "HTTP";

new_lbvserver_obj.lbmethod = "ROUNDROBIN";



//Upload the resource to NetScaler

lbvserver.add(ns_session,new_lbvserver_obj);
```

Python - Sample code to add a NetScaler resource

COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
#Create an instance of the lbvserver class

new_lbvserver_obj = lbvserver()


#Set the properties of the resource locally

new_lbvserver_obj.name = "MyFirstLbVServer"

new_lbvserver_obj.ipv46 = "10.102.29.88"

new_lbvserver_obj.port = 88

new_lbvserver_obj.servicetype = "HTTP"

new_lbvserver_obj.lbmethod = "ROUNDROBIN"


#Upload the resource to NetScaler

lbvserver.add(ns_session, new_lbvserver_obj)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Enabling a NetScaler Resource

Oct 14, 2015

To enable a resource, invoke the **enable()** method and to disable, invoke the **disable()** method.

The following sample code enables a load balancing virtual server named "lb_vip".

Java - Sample code to enable a NetScaler resource

COPY

```
lbvserver obj = new lbvserver();

obj.set_name("lb_vip");

lbvserver.enable(ns_session, obj);
```

.NET - Sample code to enable a NetScaler resource

COPY

```
lbvserver obj = new lbvserver();

obj.name = "lb_vip";

lbvserver.enable(ns_session, obj);
```

Python - Sample code to enable a NetScaler resource

COPY

```
obj = lbvserver()

obj.name = "lb_vip"

lbvserver.enable(ns_session, obj)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Retrieving Properties of NetScaler Resources

Oct 16, 2015

To retrieve the properties of a resource, you retrieve the resource object from the NetScaler appliance. Once the object is retrieved, you can extract the required properties of the resource locally, without further network traffic.

The following sample code retrieves the details of a load balancing virtual server.

Java - Sample code to get details of resource COPY

```
//Retrieve the resource object from the NetScaler

new_lbvserver_obj = lbvserver.get(ns_session,"MyFirstLbVServer");



//Extract the properties of the resource from the object locally

System.out.println(new_lbvserver_obj.get_name());

System.out.println(new_lbvserver_obj.get_servicetype());
```

.NET - Sample code to get details of resource COPY

```
//Retrieve the resource object from the NetScaler

new_lbvserver_obj = lbvserver.get(ns_session,"MyFirstLbVServer");



//Extract the properties of the resource from the object locally

Console.WriteLine(new_lbvserver_obj.name);

Console.WriteLine(new_lbvserver_obj.servicetype);
```

Python - Sample code to get details of resource COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
#Retrieve the resource object from the NetScaler

new_lbvserver_obj = lbvserver.get(ns_session,"MyFirstLbVServer")



#Extract the properties of the resource from the object locally

print(new_lbvserver_obj.name)

print(new_lbvserver_obj.servicetype)
```

Filtering Results

You can also retrieve resources by specifying a filter on the value of their properties by using the *com.citrix.netscaler.nitro.util.filtervalue* class.

For example, you can retrieve all the load balancing virtual servers that have their port set to 80 and servicetype to HTTP.

Java - Sample code to get filtered results COPY

```
filtervalue[] filter = new filtervalue[2];

filter[0] = new filtervalue("port","80");

filter[1] = new filtervalue("servicetype","HTTP");

lbvserver[] result = lbvserver.get_filtered(ns_session,filter);
```

.NET - Sample code to get filtered results COPY

```
filtervalue[] filter = new filtervalue[2];

filter[0] = new filtervalue("port","80");

filter[1] = new filtervalue("servicetype","HTTP");

lbvserver[] result = lbvserver.get_filtered(ns_session,filter);
```

Python - Sample code to get filtered results COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
filter_params = []

    filter_params = [ filtervalue() for _ in range(2)]

    filter_params[0] = filtervalue("servicetype","HTTP")

    filter_params[1] = filtervalue("port","80")

    result = lbvserver.get_filtered(ns_session1, filter_params)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Retrieving Statistics of NetScaler Resources

Oct 14, 2015

The NetScaler appliance collects statistics about the usage of its features and the corresponding resources. You can retrieve these statistics by using NITRO API. The statistics APIs are available in different packages from the configuration APIs.

For example, the API to retrieve statistics of the load balancing virtual server are available in *com.citrix.netscaler.nitro.resource.stat.lb*.

Note:

- For the python SDK, the package path is of the form *nssrc.com.citrix.netscaler.....*
- Not all NetScaler features and resources have statistic objects associated with them.

The following sample code retrieves the statistics of a load balancing virtual server and displays some of the statistics returned.

Java - Sample code to get feature statistics COPY

```
lbvserver_stats stats = lbvserver_stats.get(ns_session,"MyFirstLbVServer");

System.out.println(stats.get_curclntconnections());

System.out.println(stats.get_deferredreqrate());
```

.NET - Sample code to get feature statistics COPY

```
lbvserver_stats stats = lbvserver_stats.get(ns_session,"MyFirstLbVServer");

Console.WriteLine(stats.curclntconnections);

Console.WriteLine(stats.deferredreqrate);
```

Python - Sample code to get feature statistics COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
stats = lbvserver_stats.get(ns_session,"MyFirstLbVServer")

print(stats.curclntconnections)

print(stats.deferredreqrate)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Resetting Properties of a NetScaler Resource

Oct 16, 2015

To unset the value that is set to a parameter, invoke the **unset()** method on the resource class, by passing the name of the resource and the parameters to be unset. If the parameter has a default value, the value is reset to that value.

The following sample code unsets the load balancing method and the comments of a load balancing virtual server named "lb_123".

Java - Sample code to reset properties

COPY

```
lbvserver lb1 = new lbvserver();

lb1.set_name("lb_123");

String args[] = {"comment", "lbmethod"};

lbvserver.unset(ns_session, lb1, args);
```

.NET - Sample code to reset properties

COPY

```
lbvserver obj = new lbvserver();

obj.name = "lb_123";

String[] args = { "lbmethod","comment" };

lbvserver.unset(ns_session, lb1, args);
```

Python - Sample code to reset properties

COPY

```
lb1 = lbvserver()

    lb1.name = "lb_123"

    args = ["comment", "lbmethod"]

    lbvserver.unset(nitroService, lb1, args)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Updating a NetScaler Resource

Oct 14, 2015

To update the properties of a resource, instantiate the resource class, specify the name of the resource to be updated, configure the resource by updating its properties locally, and then upload the updated resource instance to the NetScaler appliance.

Note: Some properties in some NetScaler resources are not allowed to be modified after creation. The port number or the service type (protocol) of a load balancing virtual server or a service, are examples of such properties. Even though the update method appears to succeed, these properties retain their original values on the appliance.

The following sample code updates the service type and load balancing method of a load balancing virtual server.

Java - Sample code to update a NetScaler resource COPY

```
//Create an instance of the lbvserver class

lbvserver update_lb = new lbvserver();



//Specify the name of the lbvserver to be updated

update_lb.set_name("MyFirstLbVServer");



//Specify the updated service type and lb method

update_lb.set_servicetype("https");

update_lb.set_lbmethod("LEASTRESPONSETIME");



//Upload the resource to NetScaler

lbvserver.update(ns_session,update_lb);
```

.NET - Sample code to update a NetScaler resource COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
//Create an instance of the lbvserver class

lbvserver update_lb = new lbvserver();



//Specify the name of the lbvserver to be updated

update_lb.name = "MyFirstLbVServer";



//Specify the updated service type and lb method

update_lb.servicetype = "https";

update_lb.lbmethod = "LEASTRESPONSETIME";



//Upload the resource to NetScaler

lbvserver.update(ns_session, update_lb);
```

Python - Sample code to update a NetScaler resource

COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
#Create an instance of the lbvserver class

update_lb = lbvserver()


#Specify the name of the lbvserver to be updated

update_lb.name = "MyFirstLbVServer"


#Specify the updated service type and lb method

update_lb.servicetype = "https"

update_lb.lbmethod = "LEASTRESPONSETIME"


#Upload the resource to the NetScaler

lbvserver.update(ns_session, update_lb)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Binding NetScaler Resources

Oct 14, 2015

NetScaler resources form relationships with each other through the process of binding. This is how services are associated with a load balancing virtual server (by binding them to it), or how various policies are bound to a load balancing virtual server. Each binding relationship is represented in NITRO by its own class.

To bind one NetScaler resource to another, you must instantiate the appropriate binding class (for example, to bind a service to a load balancing virtual server, you must instantiate the **lbvserver_service_binding** class) and add it to the NetScaler configuration (by using the static **add()** method on this class).

Binding classes have a property representing the name of each resource in the binding relationship. They can also have other properties related to that relationship (for example, the weight of the binding between a load balancing virtual server and a service).

The following sample code binds a service to a load balancing virtual server, by specifying a certain weight for the binding.

Java - Sample code for binding COPY

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();

bindObj.set_name("MyFirstLbVServer");

bindObj.set_servicename("svc_prod");

bindObj.set_weight(20);

lbvserver_service_binding.add(ns_session,bindObj);
```

.NET - Sample code for binding COPY

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();

bindObj.name = "MyFirstLbVServer";

bindObj.servicename = "svc_prod";

bindObj.weight = 20;

lbvserver_service_binding.add(ns_session,bindObj);
```

Python - Sample code for binding COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
bindObj = lbvserver_service_binding()

bindObj.name = "MyFirstLbVServer"

bindObj.servicename = "svc_prod"

bindObj.weight = 20

lbvserver_service_binding.add(ns_session, bindObj)
```

Unbinding Resources

To unbind a resource from another, invoke the **delete()** method from the resource binding class, by passing the name of the two resources.

The following code sample unbinds a service from a server.

Java - Sample Code for unbinding COPY

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();

bindObj.set_name("MyFirstLbVServer");

bindObj.set_servicename("svc_prod");

lbvserver_service_binding.delete(ns_session,bindObj);
```

.NET - Sample code for unbinding COPY

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();

bindObj.name("MyFirstLbVServer");

bindObj.servicename("svc_prod");

lbvserver_service_binding.delete(ns_session,bindObj);
```

Python - Sample code for unbinding COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
bindObj = lbvserver_service_binding()

bindObj.name = "MyFirstLbVServer"

bindObj.servicename = "svc_prod"

lbvserver_service_binding.delete(ns_session, bindObj)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Globally Binding NetScaler Resources

Oct 14, 2015

Some NetScaler resources can be bound globally to affect the whole system. For example, a compression policy can be bound to an load balancing virtual server, in which case the policy affects only the traffic on that load balancing virtual server. However, if bound globally, it can affect any traffic on the appliance, regardless of which virtual servers handle the traffic.

Some NITRO classes can be used to bind resources globally. These classes have names that follow the following pattern: **<featurename>global_<resourcetype>_binding**.

For example, the class **aaaglobal_preauthenticationpolicy_binding** is used to bind preauthentication policies globally.

The following sample code creates a preauthentication action and a preauthentication policy that uses that action, and then binds the policy globally at priority 200.

Java - Sample code COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
aaapreauthenticationaction preauth_act1;

aaapreauthenticationpolicy preauth_pol1;

aaaglobal_aaapreauthenticationpolicy_binding glob_binding;

preauth_act1 = new aaapreauthenticationaction();

preauth_act1.set_name("preauth_act1");

preauth_act1.set_preauthenticationaction("ALLOW");

aaapreauthenticationaction.add(ns_session,preauth_act1);


preauth_pol1 = new aaapreauthenticationpolicy();

preauth_pol1.set_name("preauth_pol1");

preauth_pol1.set_rule("CLIENT.APPLICATION.PROCESS(antivirus.exe) EXISTS");

preauth_pol1.set_reqaction("preauth_act1");

aaapreauthenticationpolicy.add(ns_session,preauth_pol1);


glob_binding = new aaaglobal_aaapreauthenticationpolicy_binding();

glob_binding.set_policy("preauth_pol1");

glob_binding.set_priority(200);

aaaglobal_aaapreauthenticationpolicy_binding.add(ns_session,glob_binding);
```

.NET - Sample code COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
aaapreauthenticationaction preauth_act1;

aaapreauthenticationpolicy preauth_pol1;

aaaglobal_aaapreauthenticationpolicy_binding glob_binding;

preauth_act1 = new aaapreauthenticationaction();

preauth_act1.name = "preauth_act1";

preauth_act1.preauthenticationaction = "ALLOW";

aaapreauthenticationaction.add(ns_session, preauth_act1);


preauth_pol1 = new aaapreauthenticationpolicy();

preauth_pol1.name = "preauth_pol1";

preauth_pol1.rule = "CLIENT.APPLICATION.PROCESS(antivirus.exe) EXISTS";

preauth_pol1.reqaction = "preauth_act1";

aaapreauthenticationpolicy.add(ns_session, preauth_pol1);


glob_binding = new aaaglobal_aaapreauthenticationpolicy_binding();

glob_binding.policy = "preauth_pol1";

glob_binding.priority = 200;

aaaglobal_aaapreauthenticationpolicy_binding.add(ns_session,glob_binding);
```

Python - Sample code COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
preauth_act1 = aaapreauthenticationaction()

preauth_act1.name = "preauth_act1"

preauth_act1.preauthenticationaction = "ALLOW"

aaapreauthenticationaction.add(ns_session, preauth_act1)


preauth_pol1 = aaapreauthenticationpolicy()

preauth_pol1.name = "preauth_pol1"

preauth_pol1.rule = "CLIENT.APPLICATION.PROCESS(antivirus.exe) EXISTS"

preauth_pol1.reqaction = "preauth_act1"

aaapreauthenticationpolicy.add(ns_session, preauth_pol1)


glob_binding = aaaglobal_aaapreauthenticationpolicy_binding()

glob_binding.policy = "preauth_pol1"

glob_binding.priority = 200

aaaglobal_aaapreauthenticationpolicy_binding.add(ns_session, glob_binding)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Deleting a NetScaler Resource

Oct 14, 2015

To delete an existing resource, invoke the static method **delete()** on the resource class, by passing the name of the resource.

The following sample code deletes a load balancing virtual server with name "MyFirstLbVServer".

Java - Sample code to delete a NetScaler resource COPY

```
lbvserver remove_lb = new lbvserver();

remove_lb.set_name("MyFirstLbVServer");

lbvserver.delete(ns_session, remove_lb);
```

.NET - Sample code to delete a NetScaler resource COPY

```
lbvserver remove_lb = new lbvserver();

remove_lb.name("MyFirstLbVServer");

lbvserver.delete(ns_session, remove_lb);
```

Python - Sample code to delete a NetScaler resource COPY

```
remove_lb = lbvserver()

remove_lb.name = "MyFirstLbVServer"

lbvserver.delete(ns_session, remove_lb)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Performing Bulk Operations

Oct 14, 2015

You can create, retrieve, update, and delete multiple resources simultaneously and thus minimize network traffic. For example, you can add multiple load balancing virtual servers in the same operation. To perform a bulk operation, you instantiate an array of the resource class, configure the properties of all the instances locally, and then upload all the instances to the NetScaler with one command.

Specifying the Bulk Operation Behavior on the NetScaler

To account for the failure of some operations within the bulk operation, NITRO allows you to configure one of the following behaviors while establishing a connection with the appliance.

- **Exit.** When the first error is encountered, the execution stops. The commands that were executed before the error are committed.
- **Rollback.** When the first error is encountered, the execution stops. The commands that were executed before the error are rolled back. Rollback is only supported for add and bind commands.
- **Continue.** All the commands in the list are executed even if some commands fail.

Java - Sample code to specify bulk operation behavior COPY

```
nitro_service ns_session = new nitro_service("10.102.29.60","http");

ns_session.set_onerror(OnerrorEnum.CONTINUE);

ns_session.login("admin","verysecret");
```

.NET - Sample code to specify bulk operation behavior COPY

```
nitro_service ns_session = new nitro_service("10.102.29.60","http");

ns_session.onerror = OnerrorEnum.CONTINUE;

ns_session.login("admin","verysecret");
```

Python - Sample code to specify bulk operation behavior COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
ns_session = nitro_service("10.102.29.60","http")

ns_session.onerror = OnerrorEnum.CONTINUE

ns_session.login("admin","verysecret")
```

Bulk Operations

The following sample code creates two load balancing virtual servers.

Java - Sample code for bulk creation COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
//Create an array of lbvserver instances

lbvserver[] lbs = new lbvserver[2];


//Specify properties of the first lbvserver

lbs[0] = new lbvserver();

lbs[0].set_name("lbvserv1");

lbs[0].set_servicetype("http");

lbs[0].set_ipv46("10.70.136.5");

lbs[0].set_port(80);


//Specify properties of the second lbvserver

lbs[1] = new lbvserver();

lbs[1].set_name("lbvserv2");

lbs[1].set_servicetype("https");

lbs[1].set_ipv46("10.70.136.5");

lbs[1].set_port(443);


//Upload the properties of the two lbvservers to the NetScaler

lbvserver.add(ns_session,lbs);
```

.NET - Sample code for bulk creation

COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
//Create an array of lbvserver instances

lbvserver[] lbs = new lbvserver[2];


//Specify details of first lbvserver

lbs[0] = new lbvserver();

lbs[0].name = "lbvserv1";

lbs[0].servicetype = "http";

lbs[0].ipv46 = "10.70.136.5";

lbs[0].port = 80;


//Specify details of second lbvserver

lbs[1] = new lbvserver();

lbs[1].name = "lbvserv2";

lbs[1].servicetype = "https";

lbs[1].ipv46 = "10.70.136.5";

lbs[1].port = 443;


//upload the details of the lbvservers to the NITRO server

lbvserver.add(ns_session,lbs);
```

Python - Sample code for bulk creation COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
#Create an array of lbvserver instances

lbs = lbvserver[2]


#Specify properties of the first lbvserver

lbs[0] = lbvserver()

lbs[0].name = "lbvserv1"

lbs[0].servicetype = "http"

lbs[0].ipv46 = "10.70.136.5"

lbs[0].port = 80


#Specify properties of the second lbvserver

lbs[1] = lbvserver()

lbs[1].name = "lbvserv2"

lbs[1].servicetype = "https"

lbs[1].ipv46 = "10.70.136.5"

lbs[1].port = 443


#Upload the properties of the two lbvservers to the NetScaler

lbvserver.add(ns_session, lbs)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Usage Scenarios

Sep 04, 2015

In this section, we provide NITRO API specific to certain resources and scenarios. We will be adding more scenarios in future updates to this documentation.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring a NetScaler Cluster

Oct 14, 2015

For managing clusters, you can add or remove a cluster instance or an individual node and perform a few other instance or node operations such as viewing instance or node properties. You can also configure the cluster IP address. Other cluster-management tasks include joining a NetScaler appliance to the cluster and configuring a linkset. For detailed information and best practices, see Clustering.

Note: For the python SDK, the package path is of the form *nssrc.com.citrix.netscaler.....*

Cluster Instance Operations

The *com.citrix.netscaler.nitro.resource.config.cluster.clusterinstance* class provides APIs to manage a cluster instance.

The following sample code creates a cluster instance with ID 1.

Java - Sample code to create a cluster instance COPY

```
clusterinstance new_cl_inst_obj = new clusterinstance();



//Set the properties of the cluster instance locally

new_cl_inst_obj.set_clid(1);

new_cl_inst_obj.set_preemption("ENABLED");



//Upload the cluster instance

clusterinstance.add(ns_session,new_cl_inst_obj);
```

.NET - Sample code to create a cluster instance COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
clusterinstance new_cl_inst_obj = new clusterinstance();

//Set the properties of the cluster instance locally

new_cl_inst_obj.clid = 1;

new_cl_inst_obj.preemption = "ENABLED";


//Upload the cluster instance

clusterinstance.add(ns_session,new_cl_inst_obj);
```

Python - Sample code to create a cluster instance COPY

```
new_cl_inst_obj = clusterinstance()


#Set the properties of the cluster instance locally

new_cl_inst_obj.clid = 1


#Upload the cluster instance

clusterinstance.add(ns_session, new_cl_inst_obj)
```

Cluster Node Operations

The *com.citrix.netscaler.nitro.resource.config.cluster.clusternode* class provides APIs to manage cluster nodes.

The following sample code adds a cluster node with NSIP address 10.102.29.60.

Java - Sample code to add a cluster node COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
clusternode new_cl_node_obj = new clusternode();

//Set the properties of the cluster node locally

new_cl_node_obj.set_nodeid(0);

new_cl_node_obj.set_ipaddress("10.102.29.60");

new_cl_node_obj.set_state("ACTIVE");

new_cl_node_obj.set_backplane("0/1/1");


//Upload the cluster node

clusternode.add(ns_session,new_cl_node_obj);
```

.NET - Sample code to add a cluster node COPY

```
clusternode new_cl_node_obj = new clusternode();

//Set the properties of the cluster node locally

new_cl_node_obj.nodeid = 0;

new_cl_node_obj.ipaddress = "10.102.29.60";

new_cl_node_obj.state = "ACTIVE";

new_cl_node_obj.backplane = "0/1/1";


//Upload the cluster node

clusternode.add(ns_session,new_cl_node_obj);
```

Python - Sample code to add a cluster node COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
new_cl_node_obj = clusternode()

#Set the properties of the cluster node locally

new_cl_node_obj.nodeid = 0

new_cl_node_obj.ipaddress = "10.102.29.60"

new_cl_node_obj.state = "ACTIVE"

new_cl_node_obj.backplane = "0/1/1"

#Upload the cluster node

clusternode.add(ns_session, new_cl_node_obj)
```

Add a Cluster IP Address

The *com.citrix.netscaler.nitro.resource.config.ns.nsip* class provides the **add()** API to configure an IP address. To configure the IP address as a cluster IP address, you must specify the type as CLIP.

The following sample code configures a cluster IP address on NetScaler appliance with IP address 10.102.29.60.

Java - Sample code to add a cluster IP address COPY

```
nsip new_nsip_obj = new nsip();

//Set the properties locally

new_nsip_obj.set_ipaddress("10.102.29.61");

new_nsip_obj.set_netmask("255.255.255.255");

new_nsip_obj.set_type("CLIP");


//Upload the cluster node

nsip.add(ns_session,new_nsip_obj);
```

.NET - Sample code to add a cluster IP address COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
nsip new_nsip_obj = new nsip();

//Set the properties locally

new_nsip_obj.ipaddress = "10.102.29.61";

new_nsip_obj.netmask = "255.255.255.255";

new_nsip_obj.type = "CLIP";


//Upload the cluster node

nsip.add(ns_session,new_nsip_obj);
```

Python - Sample code to add a cluster IP address COPY

```
new_nsip_obj = nsip()

#Set the properties locally

new_nsip_obj.ipaddress = "10.102.29.61"

new_nsip_obj.netmask = "255.255.255.255"

new_nsip_obj.type = "CLIP"

#Upload the cluster node

nsip.add(ns_session, new_nsip_obj)
```

Add a Spotted IP Address

The *com.citrix.netscaler.nitro.resource.config.ns.nsip* class provides the **add()** API to configure an IP address. To configure the IP address as spotted, you must specify the ID of the node that must own the IP address. This configuration must be done on the cluster IP address.

The following sample code configures a spotted SNIP address on a node with ID 1.

Java - Sample code to configure a spotted IP address COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
nsip new_nsip_obj = new nsip();

//Set the properties locally

new_nsip_obj.set_ipaddress("10.102.29.77");

new_nsip_obj.set_netmask("255.255.255.0");

new_nsip_obj.set_type("SNIP");

new_nsip_obj.set_ownernode(1);


//Upload the cluster node

nsip.add(ns_session,new_nsip_obj);
```

.NET - Sample code to configure a spotted IP address COPY

```
nsip new_nsip_obj = new nsip();

//Set the properties locally

new_nsip_obj.ipaddress = "10.102.29.77";

new_nsip_obj.netmask = "255.255.255.0";

new_nsip_obj.type = "SNIP";

new_nsip_obj.ownernode = 1;


//Upload the cluster node

nsip.add(ns_session,new_nsip_obj);
```

Python - Sample code to configure a spotted IP address COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
#Add a spotted IP address

new_nsip_obj = nsip()


#Set the properties locally

new_nsip_obj.ipaddress = "10.102.29.77"

new_nsip_obj.netmask = "255.255.255.0"

new_nsip_obj.type = "SNIP"

new_nsip_obj.ownernode = 1


#Upload the cluster node

nsip.add(ns_session, new_nsip_obj)
```

Join NetScaler Appliance to Cluster

The *com.citrix.netscaler.nitro.resource.config.cluster.cluster* class provides the **join()** API to join a NetScaler appliance to the cluster. You must specify the cluster IP address and the nsroot password of the configuration coordinator.

The following sample code joins a NetScaler appliance to a cluster.

Java - Sample code to join an appliance to a cluster COPY

```
cluster new_cl_obj = new cluster();

//Set the properties of the cluster  locally

new_cl_obj.set_clip("10.102.29.61");

new_cl_obj.set_password("verysecret");



//Upload the cluster

cluster.add(ns_session,new_cl_obj);
```

.NET - Sample code to join an appliance to a cluster COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
cluster new_cl_obj = new cluster();

//Set the properties of the cluster locally

new_cl_obj.clip = "10.102.29.61";

new_cl_obj.password = "verysecret";


//Upload the cluster node

cluster.add(ns_session,new_cl_node_obj);
```

Python - Sample code to join an appliance to a cluster COPY

```
new_cl_obj = cluster()


#Set the properties of the cluster locally

new_cl_obj.clip = "10.102.29.61"

new_cl_obj.password = "verysecret"


#Upload the cluster

cluster.add(ns_session, new_cl_obj)
```

Linkset Operations

The *com.citrix.netscaler.nitro.resource.config.network.linkset* class provides the APIs to manage linksets.

To configure a linkset, do the following:

1. Add a linkset by invoking the **add()** method of the *linkset* class.
2. Bind the interfaces to the linkset using the **add()** method of the *linkset_interface_binding* class.

The following sample code creates a linkset LS/1 and bind interfaces 1/1/2 and 2/1/2 to it.

Java - Sample code to configure linksets COPY

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
linkset new_linkset_obj = new linkset();

new_linkset_obj.set_id("LS/1");

linkset.add(ns_session,new_linkset_obj);


//Bind the interfaces to the linkset

linkset_interface_binding new_linkif_obj = new linkset_interface_binding();

new_linkif_obj.set_id("LS/1");

new_linkif_obj.set_ifnum("1/1/2 2/1/2");

linkset_interface_binding.add(ns_session,new_linkif_obj);
```

.NET - Sample code to configure linksets

COPY

```
linkset new_linkset_obj = new linkset();

new_linkset_obj.id = "LS/1";

linkset.add(ns_session,new_linkset_obj);


//Bind the interfaces to the linkset

linkset_interface_binding new_linkif_obj = new linkset_interface_binding();

new_linkif_obj.id = "LS/1";

new_linkif_obj.ifnum = "1/1/2 2/1/2";

linkset_interface_binding.add(ns_session,new_linkif_obj);
```

Python - Sample code to configure linksets

COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
#Create a new linkset

new_linkset_obj = linkset()

new_linkset_obj.id = "LS/1"

linkset.add(ns_session, new_linkset_obj)



#Bind the interfaces to the linkset

new_linkif_obj = linkset_interface_binding()

new_linkif_obj.id = "LS/1"

new_linkif_obj.ifnum = "1/1/2 2/1/2"

linkset_interface_binding.add(ns_session, new_linkif_obj)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Configuring Admin Partitions

Oct 16, 2015

To create an admin partition, you must perform a set of operations on the default partition. To understand this procedure, let us consider a company that has two departments each of which has an application that requires the NetScaler functionality. The NetScaler admin wants to have a different partition for each department so that there is isolation of users and configurations. The NetScaler admin must do the following (the sample shows configurations only for a single admin partition):

Note: For detailed information and best practices, see Admin Partitions.

Creating an Admin Partition

While creating an admin partition, you must also specify the system resources that must be allocated to that partition.

The following sample code creates an admin partition named "partition-dept1".

Java - Sample code to create an admin partition COPY

```
nspartition nspartitionObject = new nspartition();

nspartitionObject.set_partitionname("partition-dept1");

nspartitionObject.set_maxbandwidth(10240);

nspartitionObject.set_maxconn(1024);

nspartitionObject.set_maxmemlimit(10);

nspartitionObject.set_minbandwidth(1240);

base_response result = nspartition.add(nitroService, nspartitionObject);
```

.NET - Sample code to create an admin partition COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
nspartition nspartitionObject = new nspartition();

nspartitionObject.partitionname = "partition-dept1";

nspartitionObject.maxbandwidth = 10240;

nspartitionObject.maxconn = 1024;

nspartitionObject.maxmemlimit = 10;

nspartitionObject.minbandwidth = 1240;

base_response result = nspartition.add(nitroService, nspartitionObject);
```

Python - Sample code to create an admin partition COPY

```
nspartitionObject = nspartition()

nspartitionObject.partitionname = "partition-dept1"

nspartitionObject.maxbandwidth = 10240

nspartitionObject.maxconn = 1024

nspartitionObject.maxmemlimit = 10

nspartitionObject.minbandwidth = 1240

result = nspartition.add(nitroService, nspartitionObject)
```

Associating Users with Partitions

Associate the appropriate users with the partition.

The following sample code associates "user1" to a partition named "partition-dept1".

Java - Sample code for associating user with partition COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
systemuser_nspartition_binding systemuser_nspartition_binding_object = new systemuser_nspartition_binding();

systemuser_nspartition_binding_object.set_partitionname("partition-dept1");

systemuser_nspartition_binding_object.set_username("user1");

base_response result = systemuser_nspartition_binding.add(nitroService, systemuser_nspartition_binding_object);
```

.NET - Sample code for associating user with partition COPY

```
systemuser_nspartition_binding systemuser_nspartition_binding_object = new systemuser_nspartition_binding();

systemuser_nspartition_binding_object.partitionname = "partition-dept1";

systemuser_nspartition_binding_object.username = "user1";

base_response result = systemuser_nspartition_binding.add(nitroService, systemuser_nspartition_binding_object);
```

Python - Sample code for associating user with partition COPY

```
systemuser_nspartition_binding_object =  systemuser_nspartition_binding()

systemuser_nspartition_binding_object.partitionname = "partition-dept1"

systemuser_nspartition_binding_object.username = "user1"

result = systemuser_nspartition_binding.add(nitroService, systemuser_nspartition_binding_object)
```

Specifying Command Policy for Partition Users

Associate an appropriate command policy to the admin partition user.

The following sample code associates the command policy "partition-admin" to "user1".

Java - Sample code to specify command policy for partition user COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
systemuser_systemcmdpolicy_binding binding_object = new systemuser_systemcmdpolicy_binding();

binding_object.set_username("user1");

binding_object.set_policyname("partition-admin");

binding_object.set_priority(1);

base_response result = systemuser_systemcmdpolicy_binding.add(nitroService,binding_object);
```

.NET - Sample code to specify command policy for partition user COPY

```
systemuser_systemcmdpolicy_binding binding_object = new systemuser_systemcmdpolicy_binding();

binding_object.username = "user1";

binding_object.policyname = "partition-admin";

binding_object.priority = 1;

base_response result = systemuser_systemcmdpolicy_binding.add(nitroService,binding_object);
```

Python - Sample code to specify command policy for partition user COPY

```
binding_object = systemuser_systemcmdpolicy_binding()

binding_object.username = "user1"

binding_object.policyname = "partition-admin"

binding_object.priority = 1

result = systemuser_systemcmdpolicy_binding.add(nitroService,binding_object)
```

Specifying the Admin Partition VLAN or Bridgegroup

Specify the VLANs or bridgegroups to be associated with the partition. This step ensures network isolation of the traffic. Traffic received on the interfaces of the VLAN or bridgegroup is isolated from the traffic of other partitions.

The following sample code specifies a VLAN for an admin partition.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Java - Sample Code to specify the VLAN

COPY

```
nspartition_vlan_binding nspartition_vlan_binding_object = new nspartition_vlan_binding();

nspartition_vlan_binding_object.set_vlan(2);

nspartition_vlan_binding_object.set_partitionname("partition-dept1");

base_response result = nspartition_vlan_binding.add(nitroService, nspartition_vlan_binding_object);
```

.NET - Sample code to specify the VLAN

COPY

```
nspartition_vlan_binding nspartition_vlan_binding_object = new nspartition_vlan_binding();

nspartition_vlan_binding_object.vlan = 2;

nspartition_vlan_binding_object.partitionname = "partition-dept1";

base_response result = nspartition_vlan_binding.add(nitroService, nspartition_vlan_binding_object);
```

Python - Sample code to specify the VLAN

COPY

```
nspartition_vlan_binding_object = nspartition_vlan_binding()

nspartition_vlan_binding_object.vlan = 2

nspartition_vlan_binding_object.partitionname = "partition-dept1"

result = nspartition_vlan_binding.add(nitroService, nspartition_vlan_binding_object)
```

Switching Partitions

If you are associated with multiple admin partitions, you can switch to the required partition.

The following sample code switches from current partition to a partition named "partition-dept2".

Java - Sample code to switch partitions

COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
nspartition nspartitionObject = new nspartition();

vnspartitionObject.set_partitionname("partition-dept2");

base_response result = nspartition.Switch(nitroService, nspartitionObject);
```

.NET - Sample code to switch partitions

COPY

```
nspartition nspartitionObject = new nspartition();

nspartitionObject.partitionname = "partition-dept2";

base_response result = nspartition.Switch(nitroService, nspartitionObject);
```

Python - Sample code to switch partitions

COPY

```
nspartitionObject = nspartition()

nspartitionObject.partitionname = "partition-dept2"

result = nspartition.Switch(nitroService, nspartitionObject)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Managing AppExpert Applications

Oct 14, 2015

Exporting an AppExpert Application

To export an AppExpert application, you must do the following:

1. Instantiate the *com.citrix.netscaler.nitro.resource.config.app.application* class.

   Note: For the python SDK, the package path is of the form *nssrc.com.citrix.netscaler.....*

2. Configure the properties of the AppExpert locally.
3. Export the AppExpert application.

The following samples export an AppExpert application named "MyApp1".

JAVA - SAMPLE CODE TO EXPORT AN APPEXPERT APPLICATION COPY

```
application myapp = new application();

myapp.set_appname("MyApp1");

myapp.set_apptemplatefilename("myapp_template");

application.export(ns_session,myapp);
```

.NET - SAMPLE CODE TO EXPORT AN APPEXPERT APPLICATION COPY

```
application myapp = new application();

myapp.appname = "MyApp1";

myapp.apptemplatefilename = "myapp_template";

application.export(ns_session,myapp);
```

python - SAMPLE CODE TO EXPORT AN APPEXPERT APPLICATION COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
myapp = application()

myapp.appname = "MyApp1"

myapp.apptemplatefilename = "myapp_template"

application.export(ns_session, myapp)
```

Importing an AppExpert Application

To import an AppExpert application, you must do the following:

1. Instantiate the *com.citrix.netscaler.nitro.resource.config.app.application* class.

   Note: For the python SDK, the package path is of the form *nssrc.com.citrix.netscaler.....*

2. Configure the properties of the AppExpert locally.
3. Import the AppExpert application.

The following samples import an AppExpert application named "MyApp1".

Java - Sample code to import an AppExpert application COPY

```
application myapp = new application();

myapp.set_appname("MyApp1");

myapp.set_apptemplatefilename("myapp_template");

application.Import(ns_session,myapp);
```

.NET - SAMPLE CODE TO IMPORT AN APPEXPERT APPLICATION COPY

© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
application myapp = new application();

myapp.appname = "MyApp1";

myapp.apptemplatefilename = "myapp_template";

application.Import(ns_session,myapp);
```

Python - SAMPLE CODE TO IMPORT AN APPEXPERT APPLICATION

COPY

```
myapp = application()

myapp.appname = "MyApp1"

myapp.apptemplatefilename = "myapp_template"

application.Import(ns_session, myapp)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exception Handling

Jun 03, 2014

The status of a NITRO request is captured in the com.citrix.netscaler.nitro.exception.nitro_exception class. This class provides the following details of the exception:

- **Session ID.** The session in which the exception occurred.
- **Severity.** The severity of the exception: error or warning. By default, only errors are captured. To capture warnings, you must set the warning flag to true, while connecting to the appliance.
- **Error code.** The status of the NITRO request. An error code of 0 indicates that the NITRO request is successful. A non-zero error code indicates an error in processing the NITRO request.
- **Error message.** Provides a brief description of the exception.

For a list of error codes, see the errorlisting.html file available in the <NITRO_SDK_HOME>/doc/api_reference folder.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Java API

Jun 05, 2014

NetScaler NITRO APIs are categorized depending on the scope and purpose of the APIs into system APIs, feature configuration APIs, and feature statistics APIs. Additionally, you can import and export AppExpert applications. You can also troubleshoot NITRO operations.

Note: All NITRO operations are logged in the /var/log/nitro.log file on the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tutorials

Nov 05, 2013

These tutorials demonstrate the end-to-end usage of NITRO to achieve the following:

- Create Your First NITRO Application
- Create a NetScaler Cluster

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Create Your First NITRO Application

Sep 12, 2012

After completing this tutorial, you will understand and be able to perform the following tasks:

- Integrate NITRO with the IDE
- Log in to the appliance
- Create a load balancing virtual server (lbvserver)
- Retrieve details of an lbvserver
- Delete an lbvserver
- Save the configurations on the appliance
- Log out of the appliance
- Debug the NITRO application

Before you begin, make sure that you have the latest NITRO SDK and that the client application satisfies the prerequisites for using the NITRO SDK.

Sample Code

For the executable code, see the <NITRO_SDK_HOME>/sample/MyFirstNitroApplication.java sample file.

To create your first NITRO application:

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.
2. Create a new class and name it **MyFirstNitroApplication**.
3. Create an instance of com.citrix.netscaler.nitro.service.nitro_service class. This instance is used to perform all operations on the appliance:
   nitro_service ns_session = new nitro_service("10.102.29.170","HTTP");
   This code establishes a connection with an appliance that has IP address 10.102.29.170 and uses the HTTP protocol. Replace 10.102.29.170 with the IP address of the NetScaler appliance that you have access to.
4. Use the nitro_service instance to log in to the appliance using your credentials:
   ns_session.login("admin","verysecret");
   This code logs into the appliance, with user name as admin and password as verysecret. Replace the credentials with your login credentials.
5. Enable the load balancing feature:
   String[] features_to_be_enabled = {"lb"};
   ns_session.enable_features(features_to_be_enabled);
   This code first sets the features to be enabled in an array and then enables the LB feature.
6. Create an instance of the com.citrix.netscaler.nitro.resource.config.lb.lbvserver class. You will use this instance to perform operations on the lbvserver.
   lbvserver new_lbvserver_obj = new lbvserver();
7. Use the lbvserver instance to create a new lbvserver:
   new_lbvserver_obj.set_name("MyFirstLbVServer");
   new_lbvserver_obj.set_ipv46("10.102.29.88");
   new_lbvserver_obj.set_servicetype("HTTP");
   new_lbvserver_obj.set_port(88);
   new_lbvserver_obj.set_lbmethod("ROUNDROBIN");
   lbvserver.add(ns_session,new_lbvserver_obj);

© 1999-2017 Citrix Systems, Inc. All rights reserved.

This code first sets the attributes (name, IP address, service type, port, and load balancing method) of the lbvserver locally and then adds it to the appliance by using the corresponding add() method.

8. Retrieve the details of the lbvserver you have created:
   ```
   new_lbvserver_obj = lbvserver.get(ns_session,new_lbvserver_obj.get_name());
   System.out.println("Name : " +new_lbvserver_obj.get_name() +"\n" +"Protocol : " +new_lbvserver_obj.get_servicetype());
   ```
   This code first retrieves the details of the lbvserver as an object from the NetScaler, extracts the required attributes (name and service type) from the object, and displays the results.
9. Delete the lbvserver you created in the above steps:
   ```
   lbvserver.delete(ns_session, new_lbvserver_obj.get_name());
   ```
10. Save the configurations:
    ```
    ns_session.save_config();
    ```
11. Log out of the appliance:
    ```
    ns_session.logout();
    ```

Debug the NITRO application

All NITRO exceptions are captured by the com.citrix.netscaler.nitro.exception.nitro_exception class. For a more detailed description, see [Exception Handling].

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Create a NetScaler Cluster

Nov 05, 2013

After completing this tutorial you will be able to create a two-node NetScaler cluster. To add more appliances to the cluster you must repeat the procedure that adds and joins the node to the cluster.

Sample Code

For the executable code, see the <NITRO_SDK_HOME>/sample/CreateCluster.java sample file.

To create a cluster

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.
2. Create a new class and name it CreateCluster.
3. Log on to one of the appliances that you want to add to the cluster and create a cluster:

```
//Connect to the first appliance that you want to add to the cluster
nitro_service nonClipSession0 = new nitro_service(nsipAddress0,protocol);
nonClipSession0.login(uName,password);

//Create a cluster instance
clusterinstance newClusterInstance = new clusterinstance();
newClusterInstance.set_clid(1);
clusterinstance.add(nonClipSession0,newClusterInstance);

//Add the appliance to the cluster
clusternode ClusterNode0 = new clusternode();
ClusterNode0.set_nodeid(0);
ClusterNode0.set_ipaddress(nsipAddress0);
ClusterNode0.set_state("ACTIVE");
ClusterNode0.set_backplane("0/1/1");
clusternode.add(nonClipSession0,ClusterNode0);

//Add the cluster IP address
nsip newNSIPAddress = new nsip();
newNSIPAddress.set_ipaddress(clipAddress);
newNSIPAddress.set_netmask("255.255.255.255");
newNSIPAddress.set_type("CLIP");
nsip.add(nonClipSession0,newNSIPAddress);

//Enable the cluster instance
clusterinstance.enable(nonClipSession0, newClusterInstance);

//Save the configurations
nonClipSession0.save_config();

//Warm reboot the appliance
nonClipSession0.reboot(true);
```

   The cluster is created and the first node is added to the cluster. This node becomes the initial configuration coordinator of the cluster.

4. Log on to the cluster IP address to add other appliances to the cluster:

```
//Connect to the cluster IP address
nitro_service clipSession = new nitro_service(clipAddress,protocol);
clipSession.login(uName,password);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
//Add the node to the cluster
clusternode ClusterNode1 = new clusternode();
ClusterNode1.set_nodeid(1);
ClusterNode1.set_ipaddress(nsipAddress1);
ClusterNode1.set_state("ACTIVE");
ClusterNode1.set_backplane("1/1/1");
clusternode.add(clipSession,ClusterNode1);

//Save the configurations
clipSession.save_config();
```

5. Log on to the appliance that you added in the previous step and join it to the cluster:

```
//Connect to the node that you have just added to the cluster
nitro_service nonClipSession1 = new nitro_service(nsipAddress1,protocol);
nonClipSession1.login(uName,password);

//Join the node to the cluster
cluster newCluster = new cluster();
newCluster.set_clip(clipAddress);
newCluster.set_password(password);
cluster.join(nonClipSession1,newCluster);

//Save the configurations
nonClipSession1.save_config();

//Warm reboot the appliance
nonClipSession1.reboot(true);
```

The second node is now a part of the cluster.

6. Verify the details of the cluster by logging on to the cluster IP address

```
//Retrieving the cluster node details
Long id = new Long(1);
clusternode node= clusternode.get(clipSession, id);
System.out.println("Node ID: "+ node.get_nodeid() + " | Admin state: " + node.get_state() + " | Backplane interface: "+ node.get_backplane());

//Retrieving the cluster instance details
Long id1 = new Long(1);
clusterinstance instance= clusterinstance.get(clipSession, id1);
System.out.println("Cluster instance ID: "+ instance.get_clid() + " | Operational state: " +instance.get_operationalstate());
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

System APIs

Mar 16, 2015

The first step towards using NITRO is to establish a session with the NetScaler appliance and then authenticate the session by using the NetScaler administrator's credentials.

You must create an object of the com.citrix.netscaler.nitro.service.nitro_service class by specifying the NetScaler IP (NSIP) address and the protocol to connect to the appliance (HTTP or HTTPS). You then use this object and log on to the appliance by specifying the user name and the password of the NetScaler administrator.

Note: You must have a user account on that appliance. The configuration operations that you perform are limited by the administrative roles assigned to your account.
The following sample code establishes a session with a NetScaler appliance with IP address 10.102.29.60 by using the HTTPS protocol:

```
//Specify the NetScaler appliance IP address and protocol
nitro_service ns_session = new nitro_service("10.102.29.60","https");

//Specify the login credentials
ns_session.login("admin","verysecret");
```

Note: When using HTTPS, you must make sure that the root CA is added to the truststore. By default, NITRO validates the SSL certificate and verifies the hostname. To disable this validation, use the following:

```
ns_session.set_certvalidation(false);
ns_session.set_hostnameverification(false);
```

Note: You can modify the timeout period by specifying a new timeout period (in seconds) in the login method. For example, to modify the timeout period to 60 minutes:

```
ns_session.login("admin","verysecret",3600);
```

Some points to note with regards to session timeout for NetScaler 10.5 and later versions:

- When restricted timeout param is enabled, NITRO, by default, uses the timeout value that is configured for the logged in user. You can customize this value but it must be limited to the value specified for the user. If no value is specified for the user, the default timeout value of 15 minutes is used.
- When restricted timeout param is not enabled, NITRO uses the default value of 30 minutes as session timeout.

You must use the nitro_service object in all further NITRO operations on the appliance. For example to save the configurations on the appliance, you must use the nitro_service object as follows:

```
ns_session.save_config();
```

The nitro_service class also provides APIs to perform other system-level operations such as enabling and disabling NetScaler features and modes, saving and clearing NetScaler configurations, setting the session timeout, setting the severity of the exceptions to be handled, setting the behavior of bulk operations, and disconnecting from the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature Configuration APIs

Jun 03, 2014

NetScaler resources are organized into a set of packages or namespaces. Each package or namespace corresponds to a NetScaler feature. For example, all load-balancing related resources, such as load balancing virtual server, load balancing group, and load balancing monitor are available in com.citrix.netscaler.nitro.resource.config.lb.

Similarly, all application firewall related resources, such as application firewall policy and application firewall archive are available in com.citrix.netscaler.nitro.resource.config.appfw.

Each NetScaler resource is represented by a class. For example, the class that represents a load balancing virtual server is called lbvserver (in com.citrix.netscaler.nitro.resource.config.lb). The state of a resource is represented by properties of a class. You can set the value for these properties by using the set_<propertyname>() methods provided by the resource class. For example to set the IP address of a load balancing virtual server, the lbvserver class provides the set_ipv46() method. Similarly, you can get the value of these properties by using the get_<propertyname>() methods of the resource class.

Note: The setter and getter properties are always executed locally on the client. They do not involve any network interaction with the NITRO web service. All properties have basic simple types: integer, long, boolean, and string.
A resource class provides APIs to perform the following operations:

Create | Retrieve | Update | Delete | Enable/Disable | Unset | Bind/Unbind | Global bind | Bulk operations

Create

To create a new resource, instantiate the resource class, configure the resource by setting its properties locally, and then upload the new resource instance to the NetScaler appliance.

The following sample code creates a load balancing virtual server:

```
//Create an instance of the lbvserver class
lbvserver new_lbvserver_obj = new lbvserver();

//Set the properties of the resource locally
new_lbvserver_obj.set_name("MyFirstLbVServer");
new_lbvserver_obj.set_ipv46("10.102.29.88");
new_lbvserver_obj.set_port(88);
new_lbvserver_obj.set_servicetype("HTTP");
new_lbvserver_obj.set_lbmethod("ROUNDROBIN");

//Upload the resource to NetScaler
lbvserver.add(ns_session,new_lbvserver_obj);
```

Retrieve

To retrieve the properties of a resource, you retrieve the resource object from the NetScaler appliance. Once the object is retrieved, you can extract the required properties of the resource locally, without further network traffic.

The following sample code retrieves the details of a load balancing virtual server:

```
//Retrieve the resource object from the NetScaler
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
new_lbvserver_obj = lbvserver.get(ns_session,"MyFirstLbVServer");

//Extract the properties of the resource from the object locally
System.out.println(new_lbvserver_obj.get_name());
System.out.println(new_lbvserver_obj.get_servicetype());
```

You can also retrieve resources by specifying a filter on the value of their properties by using the com.citrix.netscaler.nitro.util.filtervalue class.

For example, you can retrieve all the load balancing virtual servers that have their port set to 80 and servicetype to HTTP:

```
filtervalue[] filter = new filtervalue[2];
filter[0] = new filtervalue("port","80");
filter[1] = new filtervalue("servicetype","HTTP");
lbvserver[] result = lbvserver.get_filtered(ns_session,filter);
```

You can also retrieve all NetScaler resources of a certain type, such as all services in the NetScaler appliance, by calling the static get() method on the service class, without providing a second parameter, as follows:

```
service[] resources = service.get(ns_session);
```

Update

To update the properties of a resource, instantiate the resource class, specify the name of the resource to be updated, configure the resource by updating its properties locally, and then upload the updated resource instance to the NetScaler appliance.

The following sample code updates the service type and load balancing method of a load balancing virtual server:

```
//Create an instance of the lbvserver class
lbvserver update_lb = new lbvserver();

//Specify the name of the lbvserver to be updated
update_lb.set_name("MyFirstLbVServer");

//Specify the updated service type and lb method
update_lb.set_servicetype("https");
update_lb.set_lbmethod("LEASTRESPONSETIME");

//Upload the resource to NetScaler
lbvserver.update(ns_session,update_lb);
```

Note: Some properties in some NetScaler resources are not allowed to be modified after creation. The port number or the service type (protocol) of a load balancing virtual server or a service, are examples of such properties. Even though the update method appears to succeed, these properties retain their original values on the appliance.

Delete

To delete an existing resource, invoke the delete() method on the resource class, by passing the name of the resource.

The following sample code deletes a load balancing virtual server with name "MyFirstLbVServer":

```
lbvserver remove_lb = new lbvserver();
remove_lb.set_name("MyFirstLbVServer");
lbvserver.delete(ns_session, remove_lb);
```

Enable/Disable

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

To enable a resource, invoke the enable() method.

The following sample code enables a load balancing virtual server named "lb_vip":

```
lbvserver obj = new lbvserver();
obj.set_name = "lb_vip";
lbvserver.enable(ns_session, obj);
```

Note: To disable a resource, invoke the disable() method.

```
lbvserver.disable(ns_session,obj);
```

Unset

To unset the value that is set to a parameter, invoke the unset() method on the resource class, by passing the name of the resource and the parameters to be unset. If the parameter has a default value, the value is reset to that value.

The following sample code unsets the load balancing method and the comments of a load balancing virtual server named "lb_123":

```
lbvserver lb1 = new lbvserver();
lb1.set_name("lb_123");
String args[] = {"comment", "lbmethod"};
lbvserver.unset(ns_session, lb1, args);
```

Bind/Unbind

NetScaler resources form relationships with each other through the process of binding. This is how services are associated with a load balancing virtual server (by binding them to it), or how various policies are bound to a load balancing virtual server. Each binding relationship is represented in NITRO by its own class.

To bind one NetScaler resource to another, you must instantiate the appropriate binding class (for example, to bind a service to a load balancing virtual server, you must instantiate the lbvserver_service_binding class) and add it to the NetScaler configuration (by using the static add() method on this class).

Binding classes have a property representing the name of each resource in the binding relationship. They can also have other properties related to that relationship (for example, the weight of the binding between a load balancing virtual server and a service).

The following sample code binds a service to a load balancing virtual server, by specifying a certain weight for the binding:

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();
bindObj.set_name("MyFirstLbVServer");
bindObj.set_servicename("svc_prod");
bindObj.set_weight(20);
lbvserver_service_binding.add(ns_session,bindObj);
```

Note: To unbind a resource from another, invoke the delete() method from the resource binding class, by passing the name of the two resources.
The following code sample unbinds a service from a server:

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();
bindObj.set_name("MyFirstLbVServer");
bindObj.set_servicename("svc_prod");
lbvserver_service_binding.delete(ns_session,bindObj);
```

Global bind

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Some NetScaler resources can be bound globally to affect the whole system. For example, a compression policy can be bound to an load balancing virtual server, in which case the policy affects only the traffic on that load balancing virtual server. However, if bound globally, it can affect any traffic on the appliance, regardless of which virtual servers handle the traffic.

Some NITRO classes can be used to bind resources globally. These classes have names that follow the following pattern: <featurename>global_<resourcetype>_binding.

For example, the class aaaglobal_preauthenticationpolicy_binding is used to bind preauthentication policies globally.

The following sample code creates a preauthentication action and a preauthentication policy that uses that action, and then binds the policy globally at priority 200:

```
aaapreauthenticationaction preauth_act1;
aaapreauthenticationpolicy preauth_pol1;
aaaglobal_aaapreauthenticationpolicy_binding glob_binding;
preauth_act1 = new aaapreauthenticationaction();
preauth_act1.set_name("preauth_act1");
preauth_act1.set_preauthenticationaction("ALLOW");
aaapreauthenticationaction.add(ns_session,preauth_act1);

preauth_pol1 = new aaapreauthenticationpolicy();
preauth_pol1.set_name("preauth_pol1");
preauth_pol1.set_rule("CLIENT.APPLICATION.PROCESS(antivirus.exe) EXISTS");
preauth_pol1.set_reqaction("preauth_act1");
aaapreauthenticationpolicy.add(ns_session,preauth_pol1);

glob_binding = new aaaglobal_aaapreauthenticationpolicy_binding();
glob_binding.set_policy("preauth_pol1");
glob_binding.set_priority(200);
aaaglobal_aaapreauthenticationpolicy_binding.add(ns_session,glob_binding);
```

Bulk operations

You can create, retrieve, update, and delete multiple resources simultaneously and thus minimize network traffic. For example, you can add multiple load balancing virtual servers in the same operation. To perform a bulk operation, you instantiate an array of the resource class, configure the properties of all the instances locally, and then upload all the instances to the NetScaler with one command.

To account for the failure of some operations within the bulk operation, NITRO allows you to configure one of the following behaviors:

- **Exit.** When the first error is encountered, the execution stops. The commands that were executed before the error are committed.
- **Rollback.** When the first error is encountered, the execution stops. The commands that were executed before the error are rolled back. Rollback is only supported for add and bind commands.
- **Continue.** All the commands in the list are executed even if some commands fail.

Note: You must configure the required behavior while establishing a connection with the appliance.

```
nitro_service ns_session = new nitro_service("10.102.29.60","http");
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
ns_session.set_onerror(OnerrorEnum.CONTINUE);
ns_session.login("admin","verysecret");
```

The following sample code creates two load balancing virtual servers:

```
//Create an array of lbvserver instances
lbvserver[] lbs = new lbvserver[2];

//Specify properties of the first lbvserver
lbs[0] = new lbvserver();
lbs[0].set_name("lbvserv1");
lbs[0].set_servicetype("http");
lbs[0].set_ipv46("10.70.136.5");
lbs[0].set_port(80);

//Specify properties of the second lbvserver
lbs[1] = new lbvserver();
lbs[1].set_name("lbvserv2");
lbs[1].set_servicetype("https");
lbs[1].set_ipv46("10.70.136.5");
lbs[1].set_port(443);

//Upload the properties of the two lbvservers to the NetScaler
lbvserver.add(ns_session,lbs);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cluster APIs

Jun 03, 2014

For managing clusters, you can add or remove a cluster instance or an individual node and perform a few other instance or node operations such as viewing instance or node properties. You can also configure the cluster IP address. Other cluster-management tasks include joining a NetScaler appliance to the cluster and configuring a linkset.

Cluster Instance Operations

The com.citrix.netscaler.nitro.resource.config.cluster.clusterinstance class provides APIs to manage a cluster instance.

The following sample code creates a cluster instance with ID 1:

```
clusterinstance new_cl_inst_obj = new clusterinstance();
//Set the properties of the cluster instance locally
new_cl_inst_obj.set_clid(1);
new_cl_inst_obj.set_preemption("ENABLED");

//Upload the cluster instance
clusterinstance.add(ns_session,new_cl_inst_obj);
```

Cluster Node Operations

The com.citrix.netscaler.nitro.resource.config.cluster.clusternode class provides APIs to manage cluster nodes.

The following sample code adds a cluster node with NSIP address 10.102.29.60:

```
clusternode new_cl_node_obj = new clusternode();
//Set the properties of the cluster node locally
new_cl_node_obj.set_nodeid(0);
new_cl_node_obj.set_ipaddress("10.102.29.60");
new_cl_node_obj.set_state("ACTIVE");
new_cl_node_obj.set_backplane("0/1/1");

//Upload the cluster node
clusternode.add(ns_session,new_cl_node_obj);
```

Add a Cluster IP Address

The com.citrix.netscaler.nitro.resource.config.ns.nsip class provides the add() API to configure an IP address. To configure the IP address as a cluster IP address, you must specify the type as CLIP.

The following sample code configures a cluster IP address on NetScaler appliance with IP address 10.102.29.60:

```
nsip new_nsip_obj = new nsip();
//Set the properties locally
new_nsip_obj.set_ipaddress("10.102.29.61");
new_nsip_obj.set_netmask("255.255.255.255");
new_nsip_obj.set_type("CLIP");

//Upload the cluster node
nsip.add(ns_session,new_nsip_obj);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Add a Spotted IP Address

The com.citrix.netscaler.nitro.resource.config.ns.nsip class provides the add() API to configure an IP address. To configure the IP address as spotted, you must specify the ID of the node that must own the IP address. This configuration must be done on the cluster IP address.

The following sample code configures a spotted SNIP address on a node with ID 1:

```
nsip new_nsip_obj = new nsip();
//Set the properties locally
new_nsip_obj.set_ipaddress("10.102.29.77");
new_nsip_obj.set_netmask("255.255.255.0");
new_nsip_obj.set_type("SNIP");
new_nsip_obj.set_ownernode(1);

//Upload the cluster node
nsip.add(ns_session,new_nsip_obj);
```

Join NetScaler Appliance to Cluster

The com.citrix.netscaler.nitro.resource.config.cluster.cluster class provides the join() API to join a NetScaler appliance to the cluster. You must specify the cluster IP address and the nsroot password of the configuration coordinator.

The following sample joins a NetScaler appliance to a cluster:

```
cluster new_cl_obj = new cluster();
//Set the properties of the cluster  locally
new_cl_obj.set_clip("10.102.29.61");
new_cl_obj.set_password("verysecret");

//Upload the cluster
cluster.add(ns_session,new_cl_obj);
```

Linkset Operations

The com.citrix.netscaler.nitro.resource.config.network.linkset class provides the APIs to manage linksets.

To configure a linkset, do the following:

1. Add a linkset by invoking the add() method of the linkset class.
2. Bind the interfaces to the linkset using the add() method of the linkset_interface_binding class.

The following sample code creates a linkset LS/1 and bind interfaces 1/1/2 and 2/1/2 to it:

```
//Create the linkset
linkset new_linkset_obj = new linkset();
new_linkset_obj.set_id("LS/1");
linkset.add(ns_session,new_linkset_obj);

//Bind the interfaces to the linkset
linkset_interface_binding new_linkif_obj = new linkset_interface_binding();
new_linkif_obj.set_id("LS/1");
new_linkif_obj.set_ifnum("1/1/2 2/1/2");
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
linkset_interface_binding.add(ns_session,new_linkif_obj);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Admin Partitions API

Nov 20, 2014

To create an admin partition, you must perform a set of operations on the default partition. To understand this procedure, let us consider a company that has two departments each of which has an application that requires the NetScaler functionality. The NetScaler admin wants to have a different partition for each department so that there is isolation of users and configurations. The NetScaler admin must do the following (the sample shows configurations only for a single admin partition):

Note: For detailed information and best practices, see Admin Partitions.

1. Create a partition and allocate the required resources to that partition.
   ```
   //Creating a new partition...
   nspartition nspartitionObject = new nspartition();
   nspartitionObject.set_partitionname("partition-dept1");
   nspartitionObject.set_maxbandwidth(10240);
   nspartitionObject.set_maxconn(1024);
   nspartitionObject.set_maxmemlimit(10);
   nspartitionObject.set_minbandwidth(1240);
   base_response result = nspartition.add(nitroService, nspartitionObject);
   ```
2. Associate the appropriate users with the partition.
   ```
   //Associating user1 to partition-dept1...
   systemuser_nspartition_binding systemuser_nspartition_binding_object = new systemuser_nspartition_binding();
   systemuser_nspartition_binding_object.set_partitionname("partition-dept1");
   systemuser_nspartition_binding_object.set_username("user1");
   base_response result = systemuser_nspartition_binding.add(nitroService, systemuser_nspartition_binding_object);
   ```
3. Associate an appropriate command policy to the admin partition user.
   ```
   //Associating partition-admin to user1...
   systemuser_systemcmdpolicy_binding systemuser_systemcmdpolicy_binding_object = new systemuser_systemcmdpolicy_binding();
   systemuser_systemcmdpolicy_binding_object.set_username("user1");
   systemuser_systemcmdpolicy_binding_object.set_policyname("partition-admin");
   systemuser_systemcmdpolicy_binding_object.set_priority(1);
   base_response result = systemuser_systemcmdpolicy_binding.add(nitroService,systemuser_systemcmdpolicy_binding_object);
   ```
4. Specify the VLANs or bridgegroups to be associated with the partition. This step ensures network isolation of the traffic. Traffic received on the interfaces of the VLAN or bridgegroup is isolated from the traffic of other partitions.
   ```
   //Specifying the VLAN to be associated with the partition...
   nspartition_vlan_binding nspartition_vlan_binding_object = new nspartition_vlan_binding();
   nspartition_vlan_binding_object.set_vlan(2);
   nspartition_vlan_binding_object.set_partitionname("partition-dept1");
   base_response result = nspartition_vlan_binding.add(nitroService, nspartition_vlan_binding_object);
   ```
5. Save the configurations
   ```
   //Saving the configurations...
   nitroService.save_config();
   ```
   The admin partition is created.

6. Now, to configure this admin partition, you must logout of the default partition and logon again. You are automatically taken to the admin partition to which you were first bound and once there you can configure the NetScaler.
   Note: If you want to configure another admin partition, perform the switch operation given in the next step before performing this step.
7. [Optional] If you are associated with multiple admin partitions, you can switch to the required partition.
   ```
   //Switching to other partition named partition-dept2...
   nspartition nspartitionObject = new nspartition();
   ```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
nspartitionObject.set_partitionname("partition-dept2");
base_response result = nspartition.Switch(nitroService, nspartitionObject);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature Statistics APIs

Jun 03, 2014

The NetScaler appliance collects statistics about the usage of its features and the corresponding resources. You can retrieve these statistics by using NITRO API. The statistics APIs are available in different packages from the configuration APIs.

The APIs to retrieve statistics of NetScaler features are available in packages that have the following pattern: com.citrix.netscaler.nitro.resource.stat.<feature>.

For example, APIs to retrieve statistics of the load balancing virtual server are available in the com.citrix.netscaler.nitro.resource.stat.lb package.

The following sample code retrieves the statistics of a load balancing virtual server and displays some of the statistics returned:

```
lbvserver_stats stats = lbvserver_stats.get(ns_session,"MyFirstLbVServer");
System.out.println(stats.get_curclntconnections());
System.out.println(stats.get_deferredreqrate());
```

Note: Not all NetScaler features and resources have statistic objects associated with them.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppExpert Application APIs

Jun 03, 2014

To export an AppExpert application, you must instantiate the com.citrix.netscaler.nitro.resource.config.app.application class, configure the properties of the AppExpert locally, and then export the AppExpert application.

The following sample code exports an AppExpert application named "MyApp1":

```
application myapp = new application();
myapp.set_appname("MyApp1");
myapp.set_apptemplatefilename("myapp_template");
application.export(ns_session,myapp);
```

You can also import an AppExpert application. You must instantiate the com.citrix.netscaler.nitro.resource.config.app.application class, configure the properties of the AppExpert locally, and then import the AppExpert application.

The following sample code imports an AppExpert application named "MyApp1":

```
application myapp = new application();
myapp.set_appname("MyApp1");
myapp.set_apptemplatefilename("myapp_template");
application.Import(ns_session,myapp);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exception Handling

Jun 03, 2014

The status of a NITRO request is captured in the com.citrix.netscaler.nitro.exception.nitro_exception class. This class provides the following details of the exception:

- **Session ID.** The session in which the exception occurred.
- **Severity.** The severity of the exception: error or warning. By default, only errors are captured. To capture warnings, you must set the warning flag to true, while connecting to the appliance.
- **Error code.** The status of the NITRO request. An error code of 0 indicates that the NITRO request is successful. A non-zero error code indicates an error in processing the NITRO request.
- **Error message.** Provides a brief description of the exception.

For a list of error codes, see the errorlisting.html file available in the <NITRO_SDK_HOME>/doc/api_reference folder.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

.NET API

Jun 05, 2014

NetScaler NITRO APIs are categorized depending on the scope and purpose of the APIs into system APIs, feature configuration APIs, and feature statistics APIs. Additionally, you can import and export AppExpert applications. You can also troubleshoot NITRO operations.

Note: All NITRO operations are logged in the /var/log/nitro.log file on the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Tutorials

Nov 05, 2013

These tutorials demonstrate the end-to-end usage of NITRO to achieve the following:

- Create Your First NITRO Application
- Create a NetScaler Cluster

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Create Your First NITRO Application

Sep 12, 2012

After completing this tutorial, you will understand and be able to perform the following tasks:

- Integrate NITRO with the IDE
- Log in to the appliance
- Create a load balancing virtual server (lbvserver)
- Retrieve details of an lbvserver
- Delete an lbvserver
- Save the configurations on the appliance
- Log out of the appliance
- Debug the NITRO application

Before you begin, make sure that you have the latest NITRO SDK and that the client application satisfies the prerequisites for using the NITRO SDK.

Sample Code

For the executable code, see the <NITRO_SDK_HOME>/sample/MyFirstNitroApplication.cs sample file.

To create your first NITRO application:

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.
2. Create a new class and name it **MyFirstNitroApplication**.
3. Create an instance of com.citrix.netscaler.nitro.service.nitro_service class. This instance is used to perform all operations on the appliance:
   nitro_service ns_session = new nitro_service("10.102.29.170", "http");
   This code establishes a connection with an appliance that has IP address 10.102.29.170 and uses the HTTP protocol. Replace 10.102.29.170 with the IP address of the NetScaler appliance that you have access to.
4. Use the nitro_service instance to log in to the appliance using your credentials:
   ns_session.login("admin","verysecret");
   This code logs into the appliance, with user name as admin and password as verysecret. Replace the credentials with your login credentials.
5. Enable the load balancing feature:
   String[] features_to_be_enabled = {"lb"};
   ns_session.enable_features(features_to_be_enabled);
   This code enables load balancing on the appliance.
6. Create an instance of the com.citrix.netscaler.nitro.resource.config.lb.lbvserver class. You will use this instance to perform operations on the lbvserver.
   lbvserver new_lbvserver_obj = new lbvserver();
7. Use the lbvserver instance to create a new lbvserver:
   new_lbvserver_obj.name = "MyFirstLbVServer";
   new_lbvserver_obj.ipv46 = "10.102.29.88";
   new_lbvserver_obj.servicetype = "HTTP";
   new_lbvserver_obj.port = 80;
   new_lbvserver_obj.lbmethod = "ROUNDROBIN";
   lbvserver.add(ns_session,new_lbvserver_obj);
   This code first sets the attributes (name, IP address, service type, port, and load balancing method) of the lbvserver locally

© 1999-2017 Citrix Systems, Inc. All rights reserved.

and then adds it to the appliance by using the corresponding add() method.

8. Retrieve the details of the lbvserver you have created:
   lbvserver new_lbvserver_obj1 = lbvserver.get(ns_session,new_lbvserver_obj.name);
   System.Console.Out.WriteLine("Name : " +new_lbvserver_obj1.name +"\n" +"Protocol : " +new_lbvserver_obj1.servicetype);
   This code first retrieves the details of the lbvserver as an object from the NetScaler, extracts the required attributes (name and service type) from the object, and displays the results.
9. Delete the lbvserver you created in the above steps:
   lbvserver.delete(ns_session, new_lbvserver_obj.name);
10. Save the configurations:
    ns_session.save_config();
11. Log out of the appliance:
    ns_session.logout();

Debug the NITRO application

All NITRO exceptions are captured by the com.citrix.netscaler.nitro.exception.nitro_exception class. For a more detailed description, see [Exception Handling].

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Create a NetScaler Cluster

Nov 05, 2013

After completing this tutorial you will be able to create a two-node NetScaler cluster. To add more appliances to the cluster you must repeat the procedure that adds and joins the node to the cluster.

Sample Code

For the executable code, see the <NITRO_SDK_HOME>/sample/CreateCluster.cs sample file.

To create a cluster

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.
2. Create a new class and name it CreateCluster.
3. Log on to one of the appliances that you want to add to the cluster and create a cluster:

```
//Connect to the first appliance that you want to add to the cluster
nitro_service nonClipSession0 = new nitro_service(nsipAddress0,protocol);
nonClipSession0.login(uName,password);

//Create a cluster instance
clusterinstance newClusterInstance = new clusterinstance();
newClusterInstance.clid = 1;
clusterinstance.add(nonClipSession0,newClusterInstance);

//Add the appliance to the cluster
clusternode ClusterNode0 = new clusternode();
ClusterNode0.nodeid = 0;
ClusterNode0.ipaddress = nsipAddress0;
ClusterNode0.state = "ACTIVE";
ClusterNode0.backplane = "0/1/1";
clusternode.add(nonClipSession0,ClusterNode0);

//Add the cluster IP address
nsip newNSIPAddress = new nsip();
newNSIPAddress.ipaddress = clipAddress;
newNSIPAddress.netmask = "255.255.255.255";
newNSIPAddress.type = "CLIP";
nsip.add(nonClipSession0,newNSIPAddress);

//Enable the cluster instance
clusterinstance.enable(nonClipSession0, newClusterInstance);

//Save the configurations
nonClipSession0.save_config();

//Warm reboot the appliance
nonClipSession0.reboot(true);
```

   The cluster is created and the first node is added to the cluster. This node becomes the initial configuration coordinator of the cluster.

4. Log on to the cluster IP address to add other appliances to the cluster:

```
//Connect to the cluster IP address
nitro_service clipSession = new nitro_service(clipAddress,protocol);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
clipSession.login(uName,password);

//Add the node to the cluster
clusternode ClusterNode1 = new clusternode();
ClusterNode1.nodeid = 1;
ClusterNode1.ipaddress = nsipAddress1;
ClusterNode1.state = "ACTIVE";
ClusterNode1.backplane = "1/1/1";
clusternode.add(clipSession,ClusterNode1);

//Save the configurations
clipSession.save_config();
```

5. Log on to the appliance that you added in the previous step and join it to the cluster:

```
//Connect to the node that you have just added to the cluster
nitro_service nonClipSession1 = new nitro_service(nsipAddress1,protocol);
nonClipSession1.login(uName,password);

//Join the node to the cluster
cluster newCluster = new cluster();
newCluster.clip = clipAddress;
newCluster.password = password;
cluster.join(nonClipSession1,newCluster);

//Save the configurations
nonClipSession1.save_config();

//Warm reboot the appliance
nonClipSession1.reboot(true);
```

The second node is now a part of the cluster.

6. Verify the details of the cluster by logging on to the cluster IP address

```
//Retrieving the cluster node details
uint id = 1;
clusternode node= clusternode.get(clipSession, id);
System.Console.Out.WriteLine("Node ID: " + node.nodeid + " | Admin state: " + node.state + " | Backplane interface: " + node.backplane);

//Retrieving the cluster instance details
uint id1 = 1;
clusterinstance instance= clusterinstance.get(clipSession, id1);
System.Console.Out.WriteLine("Cluster instance ID: "+ instance.clid + " | Operational state: " +instance.operationalstate);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

System APIs

Mar 16, 2015

The first step towards using NITRO is to establish a session with the NetScaler appliance and then authenticate the session by using the NetScaler administrator's credentials.

You must create an object of the com.citrix.netscaler.nitro.service.nitro_service class by specifying the NetScaler IP (NSIP) address and the protocol to connect to the appliance (HTTP or HTTPS). You then use this object and log on to the appliance by specifying the user name and the password of the NetScaler administrator.

Note: You must have a user account on that appliance. The configuration operations that you perform are limited by the administrative roles assigned to your account.
The following sample code establishes a session with a NetScaler appliance with IP address 10.102.29.60 by using the HTTPS protocol:

```
//Specify the NetScaler appliance IP address and protocol
nitro_service ns_session = new nitro_service("10.102.29.60","https");

//Specify the login credentials
ns_session.login("admin","verysecret");
```

Note: You can modify the timeout period by specifying a new timeout period (in seconds) in the login method. For example, to modify the timeout period to 60 minutes:

```
ns_session.login("admin","verysecret",3600);
```

Some points to note with regards to session timeout for NetScaler 10.5 and later versions:

- When restricted timeout param is enabled, NITRO, by default, uses the timeout value that is configured for the logged in user. You can customize this value but it must be limited to the value specified for the user. If no value is specified for the user, the default timeout value of 15 minutes is used.
- When restricted timeout param is not enabled, NITRO uses the default value of 30 minutes as session timeout.

You must use the nitro_service object in all further NITRO operations on the appliance. For example to save the configurations on the appliance, you must use the nitro_service object as follows:

```
ns_session.save_config();
```

The nitro_service class also provides APIs to perform other system-level operations such as enabling and disabling NetScaler features and modes, saving and clearing NetScaler configurations, setting the session timeout, setting the severity of the exceptions to be handled, setting the behavior of bulk operations, and disconnecting from the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature Configuration APIs

Jun 03, 2014

NetScaler resources are organized into a set of packages or namespaces. Each package or namespace corresponds to a NetScaler feature. For example, all load-balancing related resources, such as load balancing virtual server, load balancing group, and load balancing monitor are available in com.citrix.netscaler.nitro.resource.config.lb.

Similarly, all application firewall related resources, such as application firewall policy and application firewall archive are available in com.citrix.netscaler.nitro.resource.config.appfw.

Each NetScaler resource is represented by a class. For example, the class that represents a load balancing virtual server is called lbvserver (in com.citrix.netscaler.nitro.resource.config.lb). The state of a resource is represented by properties of a class. You can get and set the properties of the class.

Note: The setter and getter properties are always executed locally on the client. They do not involve any network interaction with the NITRO web service. All properties have basic simple types: integer, long, boolean, and string.
A resource class provides APIs to perform the following operations:

[Create] | [Retrieve] | [Update] | [Delete] | [Enable/Disable] | [Unset] | [Bind/Unbind] | [Global bind] | [Bulk operations]

Create

To create a new resource, instantiate the resource class, configure the resource by setting its properties locally, and then upload the new resource instance to the NetScaler appliance.

The following sample code creates a load balancing virtual server:

```
//Create an instance of the lbvserver class
lbvserver new_lbvserver_obj = new lbvserver();

//Set the properties of the resource locally
new_lbvserver_obj.name = "MyFirstLbVServer";
new_lbvserver_obj.ipv46 = "10.102.29.88";
new_lbvserver_obj.port = 88;
new_lbvserver_obj.servicetype = "HTTP";
new_lbvserver_obj.lbmethod = "ROUNDROBIN";

//Upload the resource to NetScaler
lbvserver.add(ns_session,new_lbvserver_obj);
```

Retrieve

To retrieve the properties of a resource, retrieve the resource object from the NetScaler appliance. Once the object is retrieved, you can extract the required properties of the resource locally, without incurring further network traffic.

The following sample code retrieves the details of a load balancing virtual server:

```
//Retrieve the resource object from the NetScaler
new_lbvserver_obj = lbvserver.get(ns_session,"MyFirstLbVServer");
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
//Extract the properties of the resource from the object locally
Console.WriteLine(new_lbvserver_obj.name);
Console.WriteLine(new_lbvserver_obj.servicetype);
```

You can also retrieve resources by specifying a filter on the value of their properties by using the com.citrix.netscaler.nitro.util.filtervalue class.

For example, you can retrieve all the load balancing virtual servers that have their port set to 80 and servicetype to HTTP:

```
filtervalue[] filter = new filtervalue[2];
filter[0] = new filtervalue("port","80");
filter[1] = new filtervalue("servicetype","HTTP");
lbvserver[] result = lbvserver.get_filtered(ns_session,filter);
```

You can also retrieve all NetScaler resources of a certain type, such as all services in the NetScaler appliance, by calling the static get() method on the service class, without providing a second parameter, as follows:

```
service[] resources = service.get(ns_session);
```

Update

To update the properties of a resource, instantiate the resource class, specify the name of the resource to be updated, configure the resource by updating its properties locally, and then upload the updated resource instance to the NetScaler appliance.

The following sample code updates the service type and load balancing method of a load balancing virtual server:

```
//Create an instance of the lbvserver class
lbvserver update_lb = new lbvserver();

//Specify the name of the lbvserver to be updated
update_lb.name = "MyFirstLbVServer";

//Specify the updated service type and lb method
update_lb.servicetype = "https";
update_lb.lbmethod = "LEASTRESPONSETIME";

//Upload the resource to NetScaler
lbvserver.update(ns_session, update_lb);
```

Note: Some properties in some NetScaler resources are not allowed to be modified after creation. The port number or the service type (protocol) of a load balancing virtual server or a service, are examples of such properties. Even though the update method appears to succeed, these properties retain their original values on the appliance.

Delete

To delete an existing resource, invoke the static method delete() on the resource class, by passing the name of the resource.

The following sample code deletes a load balancing virtual server with name "MyFirstLbVServer":

```
lbvserver remove_lb = new lbvserver();
remove_lb.name("MyFirstLbVServer");
lbvserver.delete(ns_session, remove_lb);
```

Enable/Disable

© 1999-2017 Citrix Systems, Inc. All rights reserved.

To enable a resource, invoke the enable() method.

The following sample code enables a load balancing virtual server named "lb_vip":

```
lbvserver obj = new lbvserver();
obj.name = "lb_vip";
lbvserver.enable(ns_session, obj);
```

Note: To disable a resource, invoke the disable() method.

```
lbvserver.disable(ns_session, obj);
```

Unset

To unset the value that is set to a parameter, invoke the unset() method on the resource class, by passing the name of the resource and the parameters to be unset. If the parameter has a default value, the value is reset to that value.

The following sample code unsets the load balancing method and the comments of a load balancing virtual server named "lb_123":

```
lbvserver obj = new lbvserver();
obj.name = "lb_123";
String[] args = { "lbmethod","comment" };
lbvserver.unset(ns_session, lb1, args);
```

Bind/Unbind

NetScaler resources form relationships with each other through the process of binding. This is how services are associated with a load balancing virtual server (by binding them to it), or how various policies are bound to a load balancing virtual server. Each binding relationship is represented in NITRO by its own class.

To bind one NetScaler resource to another, you must instantiate the appropriate binding class (for example, to bind a service to a load balancing virtual server, you must instantiate the lbvserver_service_binding class) and add it to the NetScaler configuration (by using the static add() method on this class).

Binding classes have a property representing the name of each resource in the binding relationship. They can also have other properties related to that relationship (for example, the weight of the binding between a load balancing virtual server and a service).

The following sample code binds a service to a load balancing virtual server, by specifying a certain weight for the binding:

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();
bindObj.name = "MyFirstLbVServer";
bindObj.servicename = "svc_prod";
bindObj.weight = 20;
lbvserver_service_binding.add(ns_session,bindObj);
```

Note: To unbind a resource from another, invoke the delete() method from the resource binding class, by passing the name of the two resources.
The following code sample unbinds a service from a server:

```
lbvserver_service_binding bindObj = new lbvserver_service_binding();
bindObj.name("MyFirstLbVServer");
bindObj.servicename("svc_prod");
lbvserver_service_binding.delete(ns_session,bindObj);
```

Global bind

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

Some NetScaler resources can be bound globally to affect the whole system. For example, a compression policy can be bound to an load balancing virtual server, in which case the policy affects only the traffic on that load balancing virtual server. However, if bound globally, it can affect any traffic on the appliance, regardless of which virtual servers handle the traffic.

Some NITRO classes can be used to bind resources globally. These classes have names that follow the following pattern: <featurename>global_<resourcetype>_binding.

For example, the class aaaglobal_preauthenticationpolicy_binding is used to bind preauthentication policies globally.

The following sample code creates a preauthentication action and a preauthentication policy that uses that action, and then binds the policy globally at priority 200:

```
aaapreauthenticationaction preauth_act1;
aaapreauthenticationpolicy preauth_pol1;
aaaglobal_aaapreauthenticationpolicy_binding glob_binding;
preauth_act1 = new aaapreauthenticationaction();
preauth_act1.name = "preauth_act1";
preauth_act1.preauthenticationaction = "ALLOW";
aaapreauthenticationaction.add(ns_session, preauth_act1);

preauth_pol1 = new aaapreauthenticationpolicy();
preauth_pol1.name = "preauth_pol1";
preauth_pol1.rule = "CLIENT.APPLICATION.PROCESS(antivirus.exe) EXISTS";
preauth_pol1.reqaction = "preauth_act1";
aaapreauthenticationpolicy.add(ns_session, preauth_pol1);

glob_binding = new aaaglobal_aaapreauthenticationpolicy_binding();
glob_binding.policy = "preauth_pol1";
glob_binding.priority = 200;
aaaglobal_aaapreauthenticationpolicy_binding.add(ns_session,glob_binding);
```

Bulk operations

You can create, retrieve, update, and delete multiple resources simultaneously and thus minimize network traffic. For example, you can add multiple load balancing virtual servers in the same operation. To perform a bulk operation, you instantiate an array of the resource class, configure the properties of all the instances locally, and then upload all the instances to the NetScaler with one command.

To account for the failure of some operations within the bulk operation, NITRO allows you to configure one of the following behaviors:

- **Exit.** When the first error is encountered, the execution stops. The commands that were executed before the error are committed.
- **Rollback.** When the first error is encountered, the execution stops. The commands that were executed before the error are rolled back. Rollback is only supported for add and bind commands.
- **Continue.** All the commands in the list are executed even if some commands fail.

Note: You must configure the required behavior while establishing a connection with the appliance.

```
nitro_service ns_session = new nitro_service("10.102.29.60","http");
ns_session.onerror = OnerrorEnum.CONTINUE;
ns_session.login("admin","verysecret");
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following sample code creates two load balancing virtual servers:

```
//Create an array of lbvserver instances
lbvserver[] lbs = new lbvserver[2];

//Specify details of first lbvserver
lbs[0] = new lbvserver();
lbs[0].name = "lbvserv1";
lbs[0].servicetype = "http";
lbs[0].ipv46 = "10.70.136.5";
lbs[0].port = 80;

//Specify details of second lbvserver
lbs[1] = new lbvserver();
lbs[1].name = "lbvserv2";
lbs[1].servicetype = "https";
lbs[1].ipv46 = "10.70.136.5";
lbs[1].port = 443;

//upload the details of the lbvservers to the NITRO server
lbvserver.add(ns_session,lbs);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cluster APIs

Nov 05, 2013

For managing clusters, you can add or remove a cluster instance or an individual node and perform a few other instance or node operations such as viewing instance or node properties. You can also configure the cluster IP address. Other cluster-management tasks include joining a NetScaler appliance to the cluster and configuring a linkset.

Cluster Instance Operations

The com.citrix.netscaler.nitro.resource.config.cluster.clusterinstance class provides APIs to manage a cluster instance.

The following sample code creates a cluster instance with ID 1:

```
clusterinstance new_cl_inst_obj = new clusterinstance();
//Set the properties of the cluster instance locally
new_cl_inst_obj.clid = 1;
new_cl_inst_obj.preemption = "ENABLED";

//Upload the cluster instance
clusterinstance.add(ns_session,new_cl_inst_obj);
```

Cluster Node Operations

The com.citrix.netscaler.nitro.resource.config.cluster.clusternode class provides APIs to manage cluster nodes.

The following sample code adds a cluster node with NSIP address 10.102.29.60:

```
clusternode new_cl_node_obj = new clusternode();
//Set the properties of the cluster node locally
new_cl_node_obj.nodeid = 0;
new_cl_node_obj.ipaddress = "10.102.29.60";
new_cl_node_obj.state = "ACTIVE";
new_cl_node_obj.backplane = "0/1/1";

//Upload the cluster node
clusternode.add(ns_session,new_cl_node_obj);
```

Add a Cluster IP Address

The com.citrix.netscaler.nitro.resource.config.ns.nsip class provides the add() API to configure an IP address. To configure the IP address as a cluster IP address, you must specify the type as CLIP.

The following sample code configures a cluster IP address on NetScaler appliance with IP address 10.102.29.60:

```
nsip new_nsip_obj = new nsip();
//Set the properties locally
new_nsip_obj.ipaddress = "10.102.29.61";
new_nsip_obj.netmask = "255.255.255.255";
new_nsip_obj.type = "CLIP";

//Upload the cluster node
nsip.add(ns_session,new_nsip_obj);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Add a Spotted IP Address

The com.citrix.netscaler.nitro.resource.config.ns.nsip class provides the add() API to configure an IP address. To configure the IP address as spotted, you must specify the ID of the node that must own the IP address. This configuration must be done on the cluster IP address.

The following sample code configures a spotted SNIP address on a node with ID 1:

```
nsip new_nsip_obj = new nsip();
//Set the properties locally
new_nsip_obj.ipaddress = "10.102.29.77";
new_nsip_obj.netmask = "255.255.255.0";
new_nsip_obj.type = "SNIP";
new_nsip_obj.ownernode = 1;

//Upload the cluster node
nsip.add(ns_session,new_nsip_obj);
```

Join NetScaler Appliance to Cluster

The com.citrix.netscaler.nitro.resource.config.cluster.cluster class provides the join() API to join a NetScaler appliance to the cluster. You must specify the cluster IP address and the nsroot password of the configuration coordinator.

The following sample code joins a NetScaler appliance to a cluster:

```
cluster new_cl_obj = new cluster();
//Set the properties of the cluster locally
new_cl_obj.clip = "10.102.29.61";
new_cl_obj.password = "verysecret";

//Upload the cluster node
cluster.add(ns_session,new_cl_node_obj);
```

Linkset Operations

The com.citrix.netscaler.nitro.resource.config.network.linkset class provides the APIs to manage linksets.

To configure a linkset, do the following:

1. Add a linkset by invoking the add() method of the linkset class.
2. Bind the interfaces to the linkset using the add() method of the linkset_interface_binding class.

The following sample code creates a linkset LS/1 and bind interfaces 1/1/2 and 2/1/2 to it:

```
//Create the linkset
linkset new_linkset_obj = new linkset();
new_linkset_obj.id = "LS/1";
linkset.add(ns_session,new_linkset_obj);

//Bind the interfaces to the linkset
linkset_interface_binding new_linkif_obj = new linkset_interface_binding();
new_linkif_obj.id = "LS/1";
new_linkif_obj.ifnum = "1/1/2 2/1/2";
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
linkset_interface_binding.add(ns_session,new_linkif_obj);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Admin Partitions API

Nov 20, 2014

To create an admin partition, you must perform a set of operations on the default partition. To understand this procedure, let us consider a company that has two departments each of which has an application that requires the NetScaler functionality. The NetScaler admin wants to have a different partition for each department so that there is isolation of users and configurations. The NetScaler admin must do the following (the sample shows configurations only for a single admin partition):

Note: For detailed information and best practices, see Admin Partitions.

1. Create a partition and allocate the required resources to that partition.

```
//Creating a new partition...
nspartition nspartitionObject = new nspartition();
nspartitionObject.partitionname = "partition-dept1";
nspartitionObject.maxbandwidth = 10240;
nspartitionObject.maxconn = 1024;
nspartitionObject.maxmemlimit = 10;
nspartitionObject.minbandwidth = 1240;
base_response result = nspartition.add(nitroService, nspartitionObject);
```

2. Associate the appropriate users with the partition.

```
//Associating user1 to partition-dept1...
systemuser_nspartition_binding systemuser_nspartition_binding_object = new systemuser_nspartition_binding();
systemuser_nspartition_binding_object.partitionname = "partition-dept1";
systemuser_nspartition_binding_object.username = "user1";
base_response result = systemuser_nspartition_binding.add(nitroService, systemuser_nspartition_binding_object);
```

3. Associate an appropriate command policy to the admin partition user.

```
//Associating partition-admin to user1...
systemuser_systemcmdpolicy_binding systemuser_systemcmdpolicy_binding_object = new systemuser_systemcmdpolicy_binding();
systemuser_systemcmdpolicy_binding_object.username = "user1";
systemuser_systemcmdpolicy_binding_object.policyname = "partition-admin";
systemuser_systemcmdpolicy_binding_object.priority = 1;
base_response result = systemuser_systemcmdpolicy_binding.add(nitroService,systemuser_systemcmdpolicy_binding_object);
```

4. Specify the VLANs or bridgegroups to be associated with the partition. This step ensures network isolation of the traffic. Traffic received on the interfaces of the VLAN or bridgegroup is isolated from the traffic of other partitions.

```
//Specifying the VLAN to be associated with the partition...
nspartition_vlan_binding nspartition_vlan_binding_object = new nspartition_vlan_binding();
nspartition_vlan_binding_object.vlan = 2;
nspartition_vlan_binding_object.partitionname = "partition-dept1";
base_response result = nspartition_vlan_binding.add(nitroService, nspartition_vlan_binding_object);
```

5. Save the configurations

```
//Saving the configurations...
nitroService.save_config();
```

The admin partition is created.

6. Now, to configure this admin partition, you must logout of the default partition and logon again. You are automatically taken to the admin partition to which you were first bound and once there you can configure the NetScaler.

   Note: If you want to configure another admin partition, perform the switch operation given in the next step before performing this step.

7. [Optional] If you are associated with multiple admin partitions, you can switch to the required partition.

```
//Switching to other partition named partition-dept2...
nspartition nspartitionObject = new nspartition();
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
nspartitionObject.partitionname = "partition-dept2";
base_response result = nspartition.Switch(nitroService, nspartitionObject);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature Statistics APIs

Mar 14, 2012

The NetScaler appliance collects statistics about the usage of its features and the corresponding resources. You can retrieve these statistics by using NITRO API. The statistics APIs are available in different namespaces from the configuration APIs.

The APIs to retrieve statistics of NetScaler features are available in namespaces that have the following pattern: com.citrix.netscaler.nitro.resource.stat.<feature>.

For example, APIs to retrieve statistics of the load balancing virtual server are available in the com.citrix.netscaler.nitro.resource.stat.lb namespace.

The following sample code retrieves the statistics of a load balancing virtual server and displays some of the statistics returned:

```
lbvserver_stats stats = lbvserver_stats.get(ns_session,"MyFirstLbVServer");
Console.WriteLine(stats.curclntconnections);
Console.WriteLine(stats.deferredreqrate);
```

Note: Not all NetScaler features and resources have statistic objects associated with them.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppExpert Application APIs

Mar 14, 2012

To export an AppExpert application, you must instantiate the com.citrix.netscaler.nitro.resource.config.app.application class, configure the properties of the AppExpert locally, and then export the AppExpert application.

The following sample code exports an AppExpert application named "MyApp1":

```
application myapp = new application();
myapp.appname = "MyApp1";
myapp.apptemplatefilename = "myapp_template";
application.export(ns_session,myapp);
```

You can also import an AppExpert application. You must instantiate the com.citrix.netscaler.nitro.resource.config.app.application class, configure the properties of the AppExpert locally, and then import the AppExpert application.

The following sample code imports an AppExpert application named "MyApp1":

```
application myapp = new application();
myapp.appname = "MyApp1";
myapp.apptemplatefilename = "myapp_template";
application.Import(ns_session,myapp);
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exception Handling

Mar 14, 2012

The status of a NITRO request is captured in the com.citrix.netscaler.nitro.exception.nitro_exception class. This class provides the following details of the exception:

- **Session ID.** The session in which the exception occurred.
- **Severity.** The severity of the exception: error or warning. By default, only errors are captured. To capture warnings, you must set the warning flag to true, while connecting to the appliance.
- **Error code.** The status of the NITRO request. An error code of 0 indicates that the NITRO request is successful. A non-zero error code indicates an error in processing the NITRO request.
- **Error message.** Provides a brief description of the exception.

For a list of error codes, see the errorlisting.html file available in the <NITRO_SDK_HOME>/doc/api_reference folder.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Python API

Jun 05, 2014

NetScaler NITRO APIs are categorized depending on the scope and purpose of the APIs into system APIs, feature configuration APIs, feature statistics APIs, and cluster APIs. Additionally, you can import and export AppExpert applications. You can also troubleshoot NITRO operations.

Note: All NITRO operations are logged in the /var/log/nitro.log file on the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

System APIs

Mar 16, 2015

The first step towards using NITRO is to establish a session with the NetScaler appliance and then authenticate the session by using the NetScaler administrator's credentials.

You must create an object of the nssrc.com.citrix.netscaler.nitro.service.nitro_service class by specifying the NetScaler IP (NSIP) address and the protocol to connect to the appliance (HTTP or HTTPS). You then use this object and log on to the appliance by specifying the user name and the password of the NetScaler administrator.

Note: You must have a user account on that appliance. The configuration operations that you perform are limited by the administrative roles assigned to your account.
The following sample code establishes a session with a NetScaler appliance with IP address 10.102.29.60 by using the HTTPS protocol:

```
//Specify the NetScaler appliance IP address and protocol
nitro_service ns_session = new nitro_service("10.102.29.60","https")

//Specify the login credentials
ns_session.login("admin","verysecret")
```

Note: When using HTTPS, you must make sure that the root CA is added to the truststore. By default, NITRO validates the SSL certificate and verifies the hostname. To disable this validation, use the following:

```
ns_session.certvalidation = false
ns_session.hostnameverification = false
```

Note: You can modify the timeout period by specifying a new timeout period (in seconds) in the login method. For example, to modify the timeout period to 60 minutes:

```
ns_session.login("admin","verysecret",3600)
```

Some points to note with regards to session timeout for NetScaler 10.5 and later versions:

- When restricted timeout param is enabled, NITRO, by default, uses the timeout value that is configured for the logged in user. You can customize this value but it must be limited to the value specified for the user. If no value is specified for the user, the default timeout value of 15 minutes is used.
- When restricted timeout param is not enabled, NITRO uses the default value of 30 minutes as session timeout.

You must use the nitro_service object in all further NITRO operations on the appliance. For example to save the configurations on the appliance, you must use the nitro_service object as follows:

```
ns_session.save_config()
```

The nitro_service class also provides APIs to perform other system-level operations such as enabling and disabling NetScaler features and modes, saving and clearing NetScaler configurations, setting the session timeout, setting the severity of the exceptions to be handled, setting the behavior of bulk operations, and disconnecting from the appliance.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature Configuration APIs

Jun 16, 2014

NetScaler resources are organized into a set of packages or namespaces. Each package or namespace corresponds to a NetScaler feature. For example, all load-balancing related resources, such as load balancing virtual server, load balancing group, and load balancing monitor are available in nssrc.com.citrix.netscaler.nitro.resource.config.lb.

Similarly, all application firewall related resources, such as application firewall policy and application firewall archive are available in nssrc.com.citrix.netscaler.nitro.resource.config.appfw.

Each NetScaler resource is represented by a class. For example, the class that represents a load balancing virtual server is called lbvserver (in nssrc.com.citrix.netscaler.nitro.resource.config.lb). The state of a resource is represented by properties of a class. You can get and set the properties of the class.

Note: The setter and getter properties are always executed locally on the client. They do not involve any network interaction with the NITRO web service. All properties have basic simple types: integer, long, boolean, and string.
A resource class provides APIs to perform the following operations:

[Create] | [Retrieve] | [Update] | [Delete] | [Enable/Disable] | [Unset] | [Bind/Unbind] | [Global bind] | [Bulk operations]

Create

To create a new resource, instantiate the resource class, configure the resource by setting its properties locally, and then upload the new resource instance to the NetScaler appliance.

The following sample code creates a load balancing virtual server:

```
//Create an instance of the lbvserver class
lbvserver new_lbvserver_obj = new lbvserver()

//Set the properties of the resource locally
new_lbvserver_obj.name = "MyFirstLbVServer"
new_lbvserver_obj.ipv46 = "10.102.29.88"
new_lbvserver_obj.port = 88
new_lbvserver_obj.servicetype = "HTTP"
new_lbvserver_obj.lbmethod = "ROUNDROBIN"

//Upload the resource to NetScaler
lbvserver.add(ns_session, new_lbvserver_obj)
```

Retrieve

To retrieve the properties of a resource, you retrieve the resource object from the NetScaler appliance. Once the object is retrieved, you can extract the required properties of the resource locally, without further network traffic.

The following sample code retrieves the details of a load balancing virtual server:

```
//Retrieve the resource object from the NetScaler
new_lbvserver_obj = lbvserver.get(ns_session,"MyFirstLbVServer")
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
//Extract the properties of the resource from the object locally
print(new_lbvserver_obj.name)
print(new_lbvserver_obj.servicetype)
```

You can also retrieve resources by specifying a filter on the value of their properties by using the nssrc.com.citrix.netscaler.nitro.util.filtervalue class.

For example, you can retrieve all the load balancing virtual servers that have their port set to 80 and servicetype to HTTP:

```
filtervalue[] filter = new filtervalue[2]
filter[0] = new filtervalue("port","80")
filter[1] = new filtervalue("servicetype","HTTP")
lbvserver[] result = lbvserver.filtered(ns_session, filter)
```

You can also retrieve all NetScaler resources of a certain type, such as all services in the NetScaler appliance, by calling the static get() method on the service class, without providing a second parameter, as follows:

```
service[] resources = service.get(ns_session)
```

Update

To update the properties of a resource, instantiate the resource class, specify the name of the resource to be updated, configure the resource by updating its properties locally, and then upload the updated resource instance to the NetScaler appliance.

The following sample code updates the service type and load balancing method of a load balancing virtual server:

```
//Create an instance of the lbvserver class
lbvserver update_lb = new lbvserver()

//Specify the name of the lbvserver to be updated
update_lb.name = "MyFirstLbVServer"

//Specify the updated service type and lb method
update_lb.servicetype = "https"
update_lb.lbmethod = "LEASTRESPONSETIME"

//Upload the resource to NetScaler
lbvserver.update(ns_session, update_lb)
```

Note: Some properties in some NetScaler resources are not allowed to be modified after creation. The port number or the service type (protocol) of a load balancing virtual server or a service, are examples of such properties. Even though the update method appears to succeed, these properties retain their original values on the appliance.

Delete

To delete an existing resource, invoke the delete() method on the resource class, by passing the name of the resource.

The following sample code deletes a load balancing virtual server with name "MyFirstLbVServer":

```
lbvserver remove_lb = new lbvserver()
remove_lb.name = "MyFirstLbVServer"
lbvserver.delete(ns_session, remove_lb)
```

Enable/Disable

To enable a resource, invoke the enable() method.

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

The following sample code enables a load balancing virtual server named "lb_vip":

```
lbvserver obj = new lbvserver()
obj.name = "lb_vip"
lbvserver.enable(ns_session, obj)
```

Note: To disable a resource, invoke the disable() method.

```
lbvserver.disable(ns_session, obj)
```

Unset

To unset the value that is set to a parameter, invoke the unset() method on the resource class, by passing the name of the resource and the parameters to be unset. If the parameter has a default value, the value is reset to that value.

The following sample code unsets the load balancing method and the comments of a load balancing virtual server named "lb_123":

```
lbvserver lb1 = new lbvserver()
lb1.name = "lb_123"
String args[] = {"comment", "lbmethod"}
lbvserver.unset(ns_session, lb1, args)
```

Bind/Unbind

NetScaler resources form relationships with each other through the process of binding. This is how services are associated with a load balancing virtual server (by binding them to it), or how various policies are bound to a load balancing virtual server. Each binding relationship is represented in NITRO by its own class.

To bind one NetScaler resource to another, you must instantiate the appropriate binding class (for example, to bind a service to a load balancing virtual server, you must instantiate the lbvserver_service_binding class) and add it to the NetScaler configuration (by using the static add() method on this class).

Binding classes have a property representing the name of each resource in the binding relationship. They can also have other properties related to that relationship (for example, the weight of the binding between a load balancing virtual server and a service).

The following sample code binds a service to a load balancing virtual server, by specifying a certain weight for the binding:

```
lbvserver_service_binding bindObj = new lbvserver_service_binding()
bindObj.name = "MyFirstLbVServer"
bindObj.servicename = "svc_prod"
bindObj.weight = 20
lbvserver_service_binding.add(ns_session, bindObj)
```

Note: To unbind a resource from another, invoke the delete() method from the resource binding class, by passing the name of the two resources.
The following code sample unbinds a service from a server:

```
lbvserver_service_binding bindObj = new lbvserver_service_binding()
bindObj.name = "MyFirstLbVServer"
bindObj.servicename = "svc_prod"
lbvserver_service_binding.delete(ns_session, bindObj)
```

Global bind

Some NetScaler resources can be bound globally to affect the whole system. For example, a compression policy can be

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

bound to an load balancing virtual server, in which case the policy affects only the traffic on that load balancing virtual server. However, if bound globally, it can affect any traffic on the appliance, regardless of which virtual servers handle the traffic.

Some NITRO classes can be used to bind resources globally. These classes have names that follow the following pattern: <featurename>global_<resourcetype>_binding.

For example, the class aaaglobal_preauthenticationpolicy_binding is used to bind preauthentication policies globally.

The following sample code creates a preauthentication action and a preauthentication policy that uses that action, and then binds the policy globally at priority 200:

```
aaapreauthenticationaction preauth_act1
aaapreauthenticationpolicy preauth_pol1
aaaglobal_aaapreauthenticationpolicy_binding glob_binding
preauth_act1 = new aaapreauthenticationaction()
preauth_act1.name = "preauth_act1"
preauth_act1.preauthenticationaction = "ALLOW"
aaapreauthenticationaction.add(ns_session, preauth_act1)

preauth_pol1 = new aaapreauthenticationpolicy()
preauth_pol1.name = "preauth_pol1"
preauth_pol1.rule = "CLIENT.APPLICATION.PROCESS(antivirus.exe) EXISTS"
preauth_pol1.reqaction = "preauth_act1"
aaapreauthenticationpolicy.add(ns_session, preauth_pol1)

glob_binding = new aaaglobal_aaapreauthenticationpolicy_binding()
glob_binding.policy = "preauth_pol1"
glob_binding.priority = 200
aaaglobal_aaapreauthenticationpolicy_binding.add(ns_session, glob_binding)
```

Bulk operations

You can create, retrieve, update, and delete multiple resources simultaneously and thus minimize network traffic. For example, you can add multiple load balancing virtual servers in the same operation. To perform a bulk operation, you instantiate an array of the resource class, configure the properties of all the instances locally, and then upload all the instances to the NetScaler with one command.

To account for the failure of some operations within the bulk operation, NITRO allows you to configure one of the following behaviors:

- **Exit.** When the first error is encountered, the execution stops. The commands that were executed before the error are committed.
- **Rollback.** When the first error is encountered, the execution stops. The commands that were executed before the error are rolled back. Rollback is only supported for add and bind commands.
- **Continue.** All the commands in the list are executed even if some commands fail.

Note: You must configure the required behavior while establishing a connection with the appliance.

```
nitro_service ns_session = new nitro_service("10.102.29.60","http")
ns_session.onerror = OnerrorEnum.CONTINUE
ns_session.login("admin","verysecret")
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

The following sample code creates two load balancing virtual servers:

```
//Create an array of lbvserver instances
lbvserver[] lbs = new lbvserver[2]

//Specify properties of the first lbvserver
lbs[0] = new lbvserver()
lbs[0].name = "lbvserv1"
lbs[0].servicetype = "http"
lbs[0].ipv46 = "10.70.136.5"
lbs[0].port = 80

//Specify properties of the second lbvserver
lbs[1] = new lbvserver()
lbs[1].name = "lbvserv2"
lbs[1].servicetype = "https"
lbs[1].ipv46 = "10.70.136.5"
lbs[1].port = 443

//Upload the properties of the two lbvservers to the NetScaler
lbvserver.add(ns_session, lbs)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Cluster APIs

Jun 16, 2014

For managing clusters, you can add or remove a cluster instance or an individual node and perform a few other instance or node operations such as viewing instance or node properties. You can also configure the cluster IP address. Other cluster-management tasks include joining a NetScaler appliance to the cluster and configuring a linkset.

Cluster Instance Operations

The nssrc.com.citrix.netscaler.nitro.resource.config.cluster.clusterinstance class provides APIs to manage a cluster instance.

The following sample code creates a cluster instance with ID 1:

```
clusterinstance new_cl_inst_obj = new clusterinstance()

//Set the properties of the cluster instance locally
new_cl_inst_obj.clid = 1
new_cl_inst_obj.preemption = "ENABLED"

//Upload the cluster instance
clusterinstance.add(ns_session, new_cl_inst_obj)
```

Cluster Node Operations

The nssrc.com.citrix.netscaler.nitro.resource.config.cluster.clusternode class provides APIs to manage cluster nodes.

The following sample code adds a cluster node with NSIP address 10.102.29.60:

```
clusternode new_cl_node_obj = new clusternode()

//Set the properties of the cluster node locally
new_cl_node_obj.nodeid = 0
new_cl_node_obj.ipaddress = "10.102.29.60"
new_cl_node_obj.state = "ACTIVE"
new_cl_node_obj.backplane = "0/1/1"

//Upload the cluster node
clusternode.add(ns_session, new_cl_node_obj)
```

Add a Cluster IP Address

The nssrc.com.citrix.netscaler.nitro.resource.config.ns.nsip class provides the add() API to configure an IP address. To configure the IP address as a cluster IP address, you must specify the type as CLIP.

The following sample code configures a cluster IP address on NetScaler appliance with IP address 10.102.29.60:

```
nsip new_nsip_obj = new nsip()

//Set the properties locally
new_nsip_obj.ipaddress = "10.102.29.61"
new_nsip_obj.netmask = "255.255.255.255"
new_nsip_obj.type = "CLIP"
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
//Upload the cluster node
nsip.add(ns_session, new_nsip_obj)
```

Add a Spotted IP Address

The nssrc.com.citrix.netscaler.nitro.resource.config.ns.nsip class provides the add() API to configure an IP address. To configure the IP address as spotted, you must specify the ID of the node that must own the IP address. This configuration must be done on the cluster IP address.

The following sample code configures a spotted SNIP address on a node with ID 1:

```
nsip new_nsip_obj = new nsip()

//Set the properties locally
new_nsip_obj.ipaddress = "10.102.29.77"
new_nsip_obj.netmask = "255.255.255.0"
new_nsip_obj.type = "SNIP"
new_nsip_obj.ownernode = 1

//Upload the cluster node
nsip.add(ns_session, new_nsip_obj)
```

Join NetScaler Appliance to Cluster

The nssrc.com.citrix.netscaler.nitro.resource.config.cluster.cluster class provides the join() API to join a NetScaler appliance to the cluster. You must specify the cluster IP address and the nsroot password of the configuration coordinator.

The following sample joins a NetScaler appliance to a cluster:

```
cluster new_cl_obj = new cluster()

//Set the properties of the cluster  locally
new_cl_obj.clip = "10.102.29.61"
new_cl_obj.password = "verysecret"

//Upload the cluster
cluster.add(ns_session, new_cl_obj)
```

Linkset Operations

The nssrc.com.citrix.netscaler.nitro.resource.config.network.linkset class provides the APIs to manage linksets.

To configure a linkset, do the following:

1. Add a linkset by invoking the add() method of the linkset class.
2. Bind the interfaces to the linkset using the add() method of the linkset_interface_binding class.

The following sample code creates a linkset LS/1 and bind interfaces 1/1/2 and 2/1/2 to it:

```
//Create the linkset
linkset new_linkset_obj = new linkset()
new_linkset_obj.id = "LS/1"
linkset.add(ns_session, new_linkset_obj)
```

 © 1999-2017 Citrix Systems, Inc. All rights reserved.

```
//Bind the interfaces to the linkset
linkset_interface_binding new_linkif_obj = new linkset_interface_binding()
new_linkif_obj.id = "LS/1"
new_linkif_obj.ifnum = "1/1/2 2/1/2"
linkset_interface_binding.add(ns_session, new_linkif_obj)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Admin Partitions API

Nov 20, 2014

To create an admin partition, you must perform a set of operations on the default partition. To understand this procedure, let us consider a company that has two departments each of which has an application that requires the NetScaler functionality. The NetScaler admin wants to have a different partition for each department so that there is isolation of users and configurations. The NetScaler admin must do the following (the sample shows configurations only for a single admin partition):

Note: For detailed information and best practices, see Admin Partitions.

1. Create a partition and allocate the required resources to that partition.

```
//Creating a new partition...
nspartition nspartitionObject = new nspartition()
nspartitionObject.partitionname = "partition-dept1"
nspartitionObject.maxbandwidth = 10240
nspartitionObject.maxconn = 1024
nspartitionObject.maxmemlimit = 10
nspartitionObject.minbandwidth = 1240
base_response result = nspartition.add(nitroService, nspartitionObject)
```

2. Associate the appropriate users with the partition.

```
//Associating user1 to partition-dept1...
systemuser_nspartition_binding systemuser_nspartition_binding_object = new systemuser_nspartition_binding()
systemuser_nspartition_binding_object.partitionname = "partition-dept1"
systemuser_nspartition_binding_object.username = "user1"
base_response result = systemuser_nspartition_binding.add(nitroService, systemuser_nspartition_binding_object)
```

3. Associate an appropriate command policy to the admin partition user.

```
//Associating partition-admin to user1...
systemuser_systemcmdpolicy_binding systemuser_systemcmdpolicy_binding_object = new systemuser_systemcmdpolicy_binding()
systemuser_systemcmdpolicy_binding_object.username = "user1"
systemuser_systemcmdpolicy_binding_object.policyname = "partition-admin"
systemuser_systemcmdpolicy_binding_object.priority = 1
base_response result = systemuser_systemcmdpolicy_binding.add(nitroService,systemuser_systemcmdpolicy_binding_object)
```

4. Specify the VLANs or bridgegroups to be associated with the partition. This step ensures network isolation of the traffic. Traffic received on the interfaces of the VLAN or bridgegroup is isolated from the traffic of other partitions.

```
//Specifying the VLAN to be associated with the partition...
nspartition_vlan_binding nspartition_vlan_binding_object = new nspartition_vlan_binding()
nspartition_vlan_binding_object.vlan = 2
nspartition_vlan_binding_object.partitionname = "partition-dept1"
base_response result = nspartition_vlan_binding.add(nitroService, nspartition_vlan_binding_object)
```

5. Save the configurations

```
//Saving the configurations...
nitroService.save_config()
```

The admin partition is created.

6. Now, to configure this admin partition, you must logout of the default partition and logon again. You are automatically taken to the admin partition to which you were first bound and once there you can configure the NetScaler.

   Note: If you want to configure another admin partition, perform the switch operation given in the next step before performing this step.

7. [Optional] If you are associated with multiple admin partitions, you can switch to the required partition.

```
//Switching to other partition named partition-dept2...
nspartition nspartitionObject = new nspartition()
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

```
nspartitionObject.partitionname = "partition-dept2"
base_response result = nspartition.Switch(nitroService, nspartitionObject)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Feature Statistics APIs

Jun 16, 2014

The NetScaler appliance collects statistics about the usage of its features and the corresponding resources. You can retrieve these statistics by using NITRO API. The statistics APIs are available in different packages from the configuration APIs.

The APIs to retrieve statistics of NetScaler features are available in packages that have the following pattern: nssrc.com.citrix.netscaler.nitro.resource.stat.<feature>.

For example, APIs to retrieve statistics of the load balancing virtual server are available in the nssrc.com.citrix.netscaler.nitro.resource.stat.lb package.

The following sample code retrieves the statistics of a load balancing virtual server and displays some of the statistics returned:

```
lbvserver_stats stats = lbvserver_stats.get(ns_session,"MyFirstLbVServer")
print(stats.curclntconnections)
print(stats.deferredreqrate)
```

Note: Not all NetScaler features and resources have statistic objects associated with them.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

AppExpert Application APIs

Jun 16, 2014

To export an AppExpert application, you must instantiate the nssrc.com.citrix.netscaler.nitro.resource.config.app.application class, configure the properties of the AppExpert locally, and then export the AppExpert application.

The following sample code exports an AppExpert application named "MyApp1":

```
application myapp = new application()
myapp.appname = "MyApp1"
myapp.apptemplatefilename = "myapp_template"
application.export(ns_session, myapp)
```

You can also import an AppExpert application. You must instantiate the nssrc.com.citrix.netscaler.nitro.resource.config.app.application class, configure the properties of the AppExpert locally, and then import the AppExpert application.

The following sample code imports an AppExpert application named "MyApp1":

```
application myapp = new application()
myapp.appname = "MyApp1"
myapp.apptemplatefilename = "myapp_template"
application.Import(ns_session, myapp)
```


© 1999-2017 Citrix Systems, Inc. All rights reserved.

Exception Handling

Jun 16, 2014

The status of a NITRO request is captured in the nssrc.com.citrix.netscaler.nitro.exception.nitro_exception class. This class provides the following details of the exception:

- **Session ID.** The session in which the exception occurred.
- **Severity.** The severity of the exception: error or warning. By default, only errors are captured. To capture warnings, you must set the warning flag to true, while connecting to the appliance.
- **Error code.** The status of the NITRO request. An error code of 0 indicates that the NITRO request is successful. A non-zero error code indicates an error in processing the NITRO request.
- **Error message.** Provides a brief description of the exception.

For a list of error codes, see the errorlisting.html file available in the <NITRO_SDK_HOME>/doc/api_reference folder.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

NITRO Changes Across Releases

Dec 08, 2015

Some NITRO API have changed across releases. This topic details information which can help you avoid compatibility issues in your application. The changes are categorized as:

- Changes made from 9.3 -> 10.1/10.5
  - Changes across NITRO flavors
  - Changes specific to NITRO SDKs
- Changes made from 10.5 57.x -> 11.0
  - Changes across NITRO flavors
  - Changes specific to NITRO SDKs

> **Note**
>
> No changes are introduced from NetScaler 10.1 to NetScaler 10.5. Therefore, you should not face any compatibility issues when migrating from NetScaler 10.1 to 10.5.

All NITRO Flavors - Changes from 9.3 to 10.1/10.5

The NITRO changes that were made in NetScaler 10.1/10.5 when compared with NetScaler 9.3.

Type of Change	Resource	Method	Attribute
Resource removed	lbmonitor_lbmetrictable_binding ---------- Replaced with the resource 'lbmonitor_metric_binding'.	-	-
	vserver	GET ---------- Perform the GET operation on specific virtual server types such as lb/cr/cs.	-
	filterpolicy	POST with 'action=unset' ----------	-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Method removed		This method is removed as unsetting the attributes('action') of a policy makes it invalid.	
	auditsyslogpolicy	POST with 'action=unset' ---------- This method is removed as unsetting the attributes('action') of a policy makes it invalid.	-
	auditnslogpolicy	POST with 'action=unset' ---------- This method is removed as unsetting the attributes('action') of a policy makes it invalid.	-
	authorizationpolicy	POST with 'action=unset' ---------- This method is removed as unsetting the attributes('action') of a policy makes it invalid.	-
Return-type	snmpengineid	GET ---------- Return type changed to an array.	-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

changed	nshostname	GET ---------- Return type changed to an array.	-
Attribute-type changed	appfwpolicy_lbvserver_binding	-	activepolicy ---------- Data type changed from Boolean to Integer.
	appfwpolicy_appfwglobal_binding	-	activepolicy ---------- Data type changed from Boolean to Integer.
	vlan	-	portbitmap ---------- Data type changed from uint to ulong.
	vlan	-	tagbitmap ---------- Data type changed from uint to ulong.
	policypatset_pattern_binding	-	indextype ---------- This attribute is moved to 'policypatset' resource as this attribute is applicable at patset level.
	system_stats	-	powersupply1failure ---------- Replaced with

© 1999-2017 Citrix Systems, Inc. All rights reserved.

			'powersupply1status'. Note: Change is applicable from NetScaler 9.3 Build 65.8.
	system_stats	-	powersupply2failure ---------- Replaced with 'powersupply2status'. Note: Change is applicable from NetScaler 9.3 Build 65.8.
	server_servicegroup_binding	-	servicetype ---------- Replaced with 'svctype'.
	server_service_binding	-	servicetype ---------- Replaced with 'svctype'.
	crvserver	-	hits ---------- Hits are calculated per policy binding hence moved this parameter to binding resources.
Attribute removed	crvserver	-	dstvsvr ---------- Replaced with 'destinationvserver'.
	crvserver	-	destvserver ---------- Replaced with 'domain'.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	crvserver	-	dnsvserver ---------- Replaced with 'dnsvservername'.
	appflowpolicylabel	-	type ---------- Replaced with 'policylabeltype'.
	sslcipher	-	ciphgrpals ---------- Replaced with 'ciphergroupname'.
	csvserver_cspolicy_binding	-	targetvserver ---------- Replaced with 'targetlbvserver'. Note: This change is applicable for the 'sslcipher_*_binding' resources also.
	csvserver_cspolicy_binding	-	targetvserver ---------- Replaced with 'targetlbvserver'.
	rewriteaction	-	allow_unsafe_pi1, allow_unsafe_pi ---------- Replaced with 'bypassSafetyCheck'.
	nsconfig	-	nwfwmode ---------- Marked as a hidden attribute.

© 1999-2017 Citrix Systems, Inc. All rights reserved.

The SDK-specific changes that were made in NetScaler 10.1/10.5 when compared with NetScaler 9.3.

Type of Change	Class	Method	Replace with...
Class removed	Routerbgp	-	This class is removed as all router configurations are deprecated in 9.2.
Method signature changed	dnsptrrec	get(dnsptrrec obj, nitro_service session)	get(nitro_service session, String reversedomain)
	dnsaddrec	get(dnsaddrec obj, nitro_service session)	get(nitro_service session, String hostname)
	dnsnsrec	get(dnsnsrec obj, nitro_service session)	get(nitro_service session, String domain)
	snmpengineid	unset(nitro_service session, String[] args)	unset(nitro_service session, snmpengineid resource, String[] args)
	arp	arp.get(nitro_service session, String ipaddress)	arp.get(nitro_service session, arp resource)
	nsip	get(nitro_service session, String ipaddress)	get(nitro_service client, nsip resource)
	nsip6	get(nitro_service session, String ipv6address)	get(nitro_service session, nsip6 resource)
	dnsmxrec	dnsmxrec.get(dnsmxrec obj, nitro_service session)	dnsmxrec[] get(nitro_service service, dnsmxrec_args args)
Method Missing	authenticationnegotiatepolicy	(base_response) unset(nitro_service session, String[] args, String name)' is missing in	-
	authenticationnegotiatepolicy	(base_response) unset(authenticationnegotiatepolicy obj, nitro_service session, String[] args)	-

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Attribute missing in method	nsconfig	(base_response) update(nsconfig obj, nitro_service session) ---------- 'nwfwmode' attribute is missing in this method.	-

All NITRO Flavors - Changes from 10.5 57.x to 11.0

The NITRO changes that were made in NetScaler 11.0 when compared with NetScaler 10.5 Build 57.x.

Type of Change	Resource	Attribute
Attribute removed	nstrace	doruntimemerge
	nstrace	tcpdump
	cacheobject	force

NITRO SDKs - Changes from 10.5 57.x to 11.0

The SDK-specific changes that were made in NetScaler 11.0 when compared with NetScaler 10.5 Build 57.x.

Type of Change	Class	Method	Replace with...
Attribute missing in method	cacheobject	(base_response) flush(cacheobject obj, nitro_service session) ---------- 'force' attribute missing in this method.	-
Method missing	clustersync	(base_response) Force(clustersync obj, nitro_service session)	(base_response) Force(nitro_service session)
	shutdown	(base_response) Shutdown(shutdown obj, nitro_service session)	(base_response) Shutdown(nitro_service session)
	systemfile	(systemfile) get(systemfile obj, nitro_service session)	-
		(base_response) reset(sslfips obj,	(base_response) reset(nitro_service

© 1999-2017 Citrix Systems, Inc. All rights reserved.

	sslfips	nitro_service session)	session)

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Unsupported NetScaler Operations

Oct 30, 2015

The topic lists the NetScaler operations that cannot be performed by using NITRO API.

> **Note**
>
> These operations can be performed on the NetScaler CLI or the GUI.

- install API (supported from NetScaler 11.1 onwards)
- diff API on nsconfig resource (supported from NetScaler 10.5 onwards)
- UI-internal APIs (update, unset, and get)
- show ns info
- shutdown
- Application firewall API
  - importwsdl
  - importcustom
  - importxmlschema
  - importxmlerrorpage
  - importhtmlerrorpage
  - rmcustom
  - rmxmlschema
  - rmxmlerrorpage
  - rmhtmlerrorpage
- CLI-specific API
  - ping/ping6
  - traceroute/traceroute6
  - start nstrace/stop nstrace/show nstrace
  - scp
  - configaudit
  - show defaults
  - show permission
  - batch
  - source

© 1999-2017 Citrix Systems, Inc. All rights reserved.

Reference Material

Jul 05, 2017

Use this reference information to get an in-depth understanding of the following NetScaler components:

- **NetScaler SNMP OIDs** - Details of the SNMP OIDs that can be used to obtain information from a NetScaler appliance.
- **NetScaler Syslog Messages** - Details of the Syslog messages given by the NetScaler appliance.
- **NetScaler CLI Commands** - Details of the commands that can be used to configure the NetScaler appliance through the CLI. You can also view the details of each command in the NetScaler CLI, by entering the "man <ns-command-name>" command.
- **NetScaler Advanced Policy Expressions** - Details of the expressions that can be used to define advanced policies.

© 1999-2017 Citrix Systems, Inc. All rights reserved.